

25th International Society for Music Information Retrieval Conference

ISMIR 2024



November 10-14, 2024
San Francisco, California, USA
and Online

Proceedings

ISMIR 2024 was organized by Blair Kaneshiro (Stanford University), Gautham Mysore (Adobe Research), Oriol Nieto (Adobe Research), Kennedy Knight (Conference Catalysts), and a diverse international committee of chairs and volunteers.

Website: <https://ismir2024.ismir.net>

ISMIR 2024 logo design: Justin Hampton (<https://justinhampton.com>)

Edited by:

Blair Kaneshiro (*Stanford University, USA*)

Gautham Mysore (*Adobe Research, USA*)

Oriol Nieto (*Adobe Research, USA*)

Chris Donahue (*Carnegie Mellon University, USA*)

Cheng-Zhi Anna Huang (*Massachusetts Institute of Technology, USA*)

Jin Ha Lee (*University of Washington, USA*)

Brian McFee (*New York University, USA*)

Matthew McCallum (*Pandora / SiriusXM, USA*)

ISBN: 978-1-7327299-4-0

Title: Proceedings of the 25th International Society for Music Information Retrieval Conference, San Francisco, California, USA and Online, Nov 10-14, 2024.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee, provided that copies are not made or distributed for profit or commercial advantage, and that copies bear this notice and the full citation on the first page.

© 2024 International Society for Music Information Retrieval



ISMIR Sponsors

Diamond Sponsors



MUSIC.AI

Platinum Sponsors



Gold Sponsors



Silver Sponsors



Bronze Sponsors



WiMIR Sponsors

Patrons



Contributors



Supporters



Organizing Team

Conference Chairs

General Chairs

Blair Kaneshiro (*Stanford University, USA*)

Gautham Mysore (*Adobe Research, USA*)

Oriol Nieto (*Adobe Research, USA*)

Scientific Program Chairs

Chris Donahue (*Carnegie Mellon University, USA*)

Cheng-Zhi Anna Huang (*Massachusetts Institute of Technology, USA*)

Jin Ha Lee (*University of Washington, USA*)

Brian McFee (*New York University, USA*)

Diversity, Equity, and Inclusion (DEI) Chair

Katherine M. Kinnaird (*U.S. Air Force Academy and Smith College, USA*)

Virtual Chair

Vinoo Alluri (*International Institute of Information Technology - Hyderabad, India*)

Sponsorships Chair

Jay LaBoeuf (*Adobe, USA*)

Accessibility Chair

So Yeon Park (*Waymo, USA*)

Advisory Chair

Ryan Groves (*Infinite Album, Switzerland*)

Creative Practice Chairs

Cynthia Liem (*Technische Universiteit Delft, Netherlands*)

Tomas Peire (*New York University, USA*)

Grants Chairs

Shuqi Dai (*Carnegie Mellon University, USA*)

Shanu Sushmita (*Northeastern University, USA*)

Hackathon Chair

Elena Georgieva (*New York University, USA*)

Industry Chairs

Brandi Frisbie (*Luminate, USA*)
Minz Won (*Suno, USA*)

Late-Breaking/Demo Chairs

Chih-Wei Wu (*Netflix, USA*)
Camille Noufi (*Stanford University, USA*)

MIREX Chair

Gus Xia (*Mohamed bin Zayed University of Artificial Intelligence, UAE*)

Newcomer Initiatives Chair

Nick Gang (*Apple, USA*)

New-to-ISMIR Chairs

Neha Rajagopalan (*Stanford University, USA*)
Ajay Srinivasamurthy (*Amazon AGI, India*)

Publications Chair

Matthew McCallum (*Pandora / SiriusXM, USA*)

Publicity Chair

Rythm Jain (*Pandora / SiriusXM, USA*)

Satellites Chair

Erick Siavichay (*Independent*)

Social Chairs

Andreas Ehmann (*Pandora / SiriusXM, USA*)
Ju-Chiang Wang (*ByteDance, USA*)

Technology Chair

Siddharth Gururani (*NVIDIA, USA*)

Tutorials Chairs

Rachel Bittner (*Spotify, France*)
Mohamed Sordo (*Pandora / SiriusXM, USA*)

Unconference Chair

Geoffroy Peeters (*Télécom Paris, France*)

Virtual Logistics Chairs

Jatin Agarwala (*International Institute of Information Technology - Hyderabad, India*)

Ray Gifford (*University of California Santa Barbara, USA*)

Lalit Mohan (*International Institute of Information Technology - Hyderabad, India*)

Web Chair

Ashvala Vinay (*Nonetype, USA*)

Event Management

Kennedy Knight (*Conference Catalysts, USA*)

Volunteers

Moiz Ali

Anna Aljanaki

Sanjay Anand Menon

Arjun Bahuguna

Stefan Balke

Nicholas Boyko

SeungHyun Cho

Karolayne Dessabato

Vani Dewan

Meiying Ding

Yiwei Ding

Natalia Farah

Michael Gancz

Tirna Ghosh

Devyani Hebbbar

Sonja Heinze

Hsiao-Tzu Hung

Hyon Kim

Tibor Kiss

Lele Liu

Shrinidhi Mahesh

Mayur Mankar

Aaquila Mariajohn

Hanyu Meng

Ivan Meresman Higgs

Anmol Mishra

Zixun Nicolas Guo

Yigitcan Ozer

Ziyue Piao

Laure Prétet

Neha Rajagopalan

Nancy Rico-Mineros

Jess Rucinski

Megha Sharma

Surabhi Shinde

Michael Taenzer

DJean Vasciannie

Joy Xu

Zeyu Yang

Sid Yu

Xueqi Zhang

Jingwei Zhao

Program Committee

Meta-Reviewers

Vinoo Alluri, IIIT - Hyderabad
Vipul Arora, IIT Kanpur
Claire Arthur, Georgia Institute of Technology
Andreas Arzt, Apple
Emmanouil Benetos, Queen Mary University of London
Dmitry Bogdanov, Universitat Pompeu Fabra
Juan J. Bosch, Spotify
Nicholas J. Bryan, Adobe Research
John Ashley Burgoyne, University of Amsterdam
Carlos Eduardo Cancino-Chacón, Johannes Kepler University Linz
Rafael Caro Repetto, Interdisciplinary Transformation University Austria
Michael Casey, Dartmouth College
Kahyun Choi, University of Illinois Urbana-Champaign
Keunwoo Choi, Gaudio Lab, Inc.
Tom Collins, University of Miami; MAIA, Inc.
Johanna Devaney, Brooklyn College
Simon Dixon, Queen Mary University of London
Hao-Wen Dong, University of Michigan
Stephen Downie, University of Illinois Urbana-Champaign
Zhiyao Duan, University of Rochester
Philippe Esling, IRCAM
Sebastian Ewert, Spotify
Ichiro Fujinaga, McGill University
Emilia Gomez, Universitat Pompeu Fabra
Masataka Goto, National Institute of Advanced Industrial Science and Technology (AIST)
Fabien Gouyon, Pandora / SiriusXM
Romain Hennequin, Deezer Research
Andre Holzapfel, KTH Royal Institute of Technology in Stockholm
Ozgur Izmirli, Connecticut College
Peter Knees, TU Wien
Katerina Kosta, ByteDance
Olivier Lartillot, RITMO, University of Oslo
Alexander Lerch, Georgia Institute of Technology
Florence Leve, Université de Picardie Jules Verne - Lab. MIS - Algomus
Cynthia C. S. Liem, Delft University of Technology
Ethan Manilow, Interactive Audio Lab, Northwestern University
Matthew C. McCallum, Pandora / SiriusXM
Cory McKay, Marianopolis College
Yuki Mitsufuji, Sony AI
Meinard Müller, International Audio Laboratories Erlangen
Juhan Nam, KAIST
Oriol Nieto, Adobe Research
Mitsunori Ogihara, University of Miami - Coral Gables, FL
Sergio Oramas, Pandora / SiriusXM
Philippe Pasquier, Simon Fraser University
Johan Pauwels, Queen Mary University of London
Geoffroy Peeters, LTCI - Télécom Paris, IP Paris
Preeti Rao, Indian Institute of Technology Bombay
Justin Salamon, Adobe Research
Joan Serra, Sony AI
Xavier Serra, Universitat Pompeu Fabra

Jordan B. L. Smith, TikTok
Mohamed Sordo, Pandora / SiriusXM
Ajay Srinivasamurthy, Amazon Alexa
Sebastian Stober, Otto von Guericke University
Bob L. T. Sturm, KTH Royal Institute of Technology
Li Su, Academia Sinica
John Thickstun, University of Washington
Timothy Tsai, Harvey Mudd College
Douglas Turnbull, Ithaca College
Cheng-i Wang, Smule, Inc.
Ye Wang, National University of Singapore
Christof Weiß, University of Würzburg
Gerhard Widmer, Johannes Kepler University
Guangyu Xia, NYU Shanghai
Yi-Hsuan Yang, National Taiwan University
Kazuyoshi Yoshii, Kyoto University
Eva Zangerle, University of Innsbruck

Reviewers

Jakob Abeßer	Guillem Cortès	Elena Georgieva
Sadie L. Allen	Louis Couturier	Riccardo Giampiccolo
Pablo Alonso-Jiménez	Laura Cros Vila	Jon Gillick
Lior Arbel	Frank Cwitkowitz	Matan Gover
Stefan Balke	Alexandre D'Hooge	Niccolo Granieri
Berker Banar	Shuqi Dai	Carlos Guedes
Julia Barnett	Roger B. Dannenberg	Siddharth Gururani
Mathieu Barthet	Reinier de Valk	Ranjani H G
Dogac Basaran	Alessio Degani	Gaëtan Hadjeres
Roser Batlle-Roca	Andrew M. Demetriou	Jan Hajič, jr.
Gilberto Bernardes	Ninon Devis	Ben Hayes
Louis Bigo	Bruno Di Giorgi	Florian Henkel
Geoffray Bonnin	Sivan Ding	Peyman Heydarian
Clara Borrelli	Christian Dittmar	Jiawen Huang
Charles Brazier	Seungheon Doh	Yu-Fen Huang
Paul Brossier	Guillaume Doras	Chris Hubbles
Dan Brown	Jonathan Driedger	Yun-Ning Hung
Bryony Buck	Xingjian Du	Karim M. Ibrahim
Morgan Buisson	Simon Durand	Charles Inskip
Marcelo Caetano	Morwared Farbood	Chang-Bin Jeon
Jorge Calvo-Zaragoza	Andres Ferraro	Dasaem Jeong
Pavel Camp	Flavio Figueiredo	Junyan Jiang
Julio Carabias	Hugo Flores García	Yaolong Ju
Mark Cartwright	Frederic Font	Maximos Kaliakatsos-Papakostas
Hugo T. Carvalho	Francesco Foscarin	Emmanouil Karystinaios
Bo-Yu Chen	Dominique Fourer	Haven Kim
Ke Chen	Klaus Frieler	Jaehun Kim
Yu-Hua Chen	Satoru Fukayama	Jong Wook Kim
Tian Cheng	Diego Furtado Silva	Phillip B. Kirlin
Ching-Yu Chiu	Giovanni Gabbolini	Anssi Klapuri
Shreyan Chowdhury	Nick Gang	Qiuqiang Kong
Ondřej Cífka	Kaustuv Kanti Ganguli	Hendrik Vincent Koops
Alice Cohen-Hadria	Chenyu Gao	Amanda E. Krause
Graham K. Coleman	Joshua Gardner	Michael Krause
Nathaniel Condit-Schultz	Roman B. Gebhardt	Kosmas Kritsis

Frank Kurth	Thomas Nuttall	Carl Thomé
Taegyun Kwon	Patricio Ovalle	Marko Tkalcic
Pierre Laffitte	Yigitcan Özer	Christopher J. Tralie
Mathieu Lagrange	Renato Panda	Kosetsu Tsukuda
Stefan Lattner	Piyush Papreja	Alexandra Uitdenbogerd
Jongpil Lee	Emilia Parada-Cabaleiro	Jose J. Valero-Mas
Mark Levy	Saebiyul Park	Jan Van Balen
David Lewis	So Yeon Park	Igor Vatolkin
Bochen Li	Marco Pasini	Gissel Velarde
Rongfeng Li	Ashis Pati	Prateek Verma
Yizhi Li	Miguel Perez Fernandez	Amruta Vidwans
Wei-Hsiang Liao	Antonio Pertusa	Ashvala Vinay
Elad Liebman	Matevž Pesek	Venkata S. Viraraghavan
Kin Wah Edward LIN	Pedro D. Pestana	Changhong Wang
Liwei Lin	Kaitlin Pet	Chung-Che Wang
Yuan-Pin Lin	Silvan Peter	Ju-Chiang Wang
Lele Liu	Christos Plachouras	Jun-You Wang
Wei-Tsung Lu	Genís Plaja-Roglans	Yu Wang
Hanna Lukashevich	Andrea Poltronieri	Ziyu Wang
Yin-Jyun Luo	Lorenzo Porcaro	Kento Watanabe
Yinghao Ma	Verena Praher	Roger Wattenhofer
Akira Maezawa	Zafar Rafii	Benno Weck
Lucas S. Maia	Antonio Ramires	I-Chieh Wei
Iliaria Manco	Pedro Ramoneda	Weixing Wei
Leandro Balby Marinho	Gaël Richard	David M. Weigl
Axel Marmoret	David Rizo	Gordon Wichern
Matija Marolt	Martín Rocamora	Julia Wilkins
Benjamin Martin	Axel Roebel	William F. Wilson
Marco A. Martinez Ramirez	Jean-Baptiste Rolland	Daniel Wolff
David Martins de Matos	Gerard Roma	Minz Won
Matthias Mauch	Iran R. Roman	Kyle J. Worrall
Rudolf Mayer	Joe Cheri Ross	Chih-Wei Wu
Gabriel Meseguer Brocal	Pedro Pereira Sarmento	Shangda Wu
Gianluca Micchi	Maximilian Schmitt	Shih-Lun Wu
Remi Mignot	Hendrik Schreiber	Yuxuan Wu
Marius Miron	Bjorn W. Schuller	Anna Xambó
Ronald Mo	Simon J. Schwär	Luwei Yang
Hyeonggi Moon	Prem Seetharaman	Furkan Yesiler
Fabio Morreale	Sertan Şentürk	Minjoon Yoo
Alia Morsi	Micael A. Silva	Ruibin Yuan
Manuel Moussallam	Anup Singh	Johannes Zeitler
Lucas N. Ferreira	George Sioros	Huan Zhang
Tomoyasu Nakano	Christian J. Steinmetz	Yixiao Zhang
Maria Navarro	Daniel Stoller	Yudong Zhao
Shahan Necessian	Fabian-Robert Stöter	Terry Yi Zhong
Michele Newman	Michael Taenzer	Ge Zhu
Javier Nistal	Nazif Can Tamer	Tiange Zhu
Camille Noufi	Jingjing Tang	Alon Ziv
Zachary Novack	Karan Thakkar	

Preface

Message from the General Chairs

We are delighted to present the proceedings of the 25th International Society for Music Information Retrieval Conference, which took place in hybrid format in the vibrant city of San Francisco and online from November 10–14, 2024. This event marked a significant culmination of the efforts of the organizing team, transforming our ambitious vision into a memorable reality. The conference brought together a diverse and dynamic group of researchers, practitioners, and enthusiasts from around the globe, all united by a passion for music information research.

The planning for ISMIR 2024 began with a conviction that the San Francisco Bay Area, with its rich legacy in both AI and music, would provide an ideal background for the conference. Our team of organizers worked to ensure that every detail was meticulously planned, from selecting ideal venues (including a legendary music hall for the jam!) to managing logistics and securing sponsorships. Despite the high costs associated with the location, our commitment to inclusivity and accessibility allowed us to welcome a broad spectrum of attendees at the conference venue and online.

The 25th ISMIR conference introduced several exciting new elements. We embraced a **Special Theme**, "Bridging Technology and Musical Creativity", which was reflected in **Creative Practice** sessions designed to connect creative and research communities. We also implemented a form of **Open Review**, making several reviews and meta-reviews publicly available alongside accepted papers. We proudly presented the first-ever **Test of Time Award** and included presentations of recent **Transactions of the International Society for Music Information Retrieval (TISMIR) publications** in the scientific program. Finally, we appointed an **Accessibility Chair** who made significant cross-functional contributions in areas that are often overlooked.

The success of ISMIR 2024 is a testament to the collaborative spirit and innovative thinking that define the MIR community. We are proud to share the insights, research, and discussions that emerged from this conference, and we look forward to the continued growth and evolution of the field. Thank you to everyone who contributed to this year's ISMIR. We had a wonderful time organizing this busy yet epic event, and we sincerely hope that attendees enjoyed it as much as we did.

Blair Kaneshiro, Gautham Mysore, and Oriol Nieto
Anchorage, AK and San Francisco, CA
December 2024

Hybrid Conference

Recent virtual and hybrid ISMIR conferences have shown that remote participation options make the conference accessible to a broader range of participants. Accordingly, this year's conference was organized in a hybrid format, with goals of enabling all attendees—whether attending in person or online—to engage with the conference program, share their work, and interact with other attendees.

The hybrid format of ISMIR 2024 included the following:

- The organizing team included a dedicated Virtual Chair and Virtual Logistics Chairs.
- Remote presentation was freely offered—that is, without needing to request special permission—for ISMIR paper, TISMIR, and LBD submissions. Presenting authors were assigned to poster sessions according to their preferred availability. Tutorial presenters could also participate remotely, as long as at least one presenter for each Tutorial was onsite.
- The virtual conference ran on a 24-hour schedule so that each participant could attend from their preferred time zone. The program was designed around a primarily synchronous online experience to foster community and engagement amongst attendees. Both the original and replay sessions included a live Slack backchannel for Oral sessions (see "Diversity and Inclusion" section, below) as well as live online sessions. The virtual conference offered asynchronous options as well, including session recordings shared amongst attendees; the ability for attendees to access all papers and LBD materials via MiniConf; and dedicated Slack channels for each presentation so that attendees could interact with authors and keynotes.

- For the first time, the ISMIR conference included a Virtual Pre-Conference. This two-day event included keynotes and other sessions across global time zones.
- The organizers aimed to support each attendee’s preferred mode of presentation as much as possible, maximizing financial support to those wishing to attend in person and offering lower registration fees (and additional need-based discounts) for virtual attendees.
- Virtual platforms of the conference were streamlined to Slack, Zoom, and the online MiniConf program hosted on the conference website.
- Finally, onsite attendees were granted full virtual access to the conference, enabling them to attend original and replay sessions online, interact via Slack, and access session recordings.

Scientific Program

The ISMIR 2024 scientific program comprised 123 papers. A total of 346 submissions (up from 229 in 2023) were reviewed out of 410 abstracts that were registered on the submission system (up from 272 in 2023). In keeping with the practices of the previous years, a two-tier double-blind review process was conducted involving a total of 256 reviewers and 70 meta-reviewers. Each paper was assigned to a single meta-reviewer and at least three reviewers, and replacement reviewers were found when the originally assigned reviewer was unable to complete their review. Meta-reviewers were also instructed to complete a full review of each of their assigned papers, in addition to the final meta-review summarizing the individual reviews. Each meta-reviewer and reviewer was responsible for no more than 5 papers, in order that the reviewing load would be manageable, thus promoting careful and thorough reviews. The initial reviewing phase was followed by a discussion period, in which reviewers and meta-reviewers could discuss and revise their assessments of each paper. Meta-reviewers were then instructed to summarize the discussion and reviews in the final report. The Scientific Program Chairs (SPC) made the final decisions on each paper, based on the recommendations of meta-reviewers and reviewers. 124 papers were accepted (one of which was later withdrawn by the authors), giving an acceptance rate of 35.84%. The SPC would like to express their thanks to the ISMIR community of reviewers and meta-reviewers for their wholehearted support of this critical aspect of a successful ISMIR technical program.

Table 1 summarizes the number of reviewed and accepted papers in each subject area (as selected by authors during the submission process) together with the corresponding proportion of papers in the program. Table 2 summarizes the publication statistics over the 24-year history of the conference. In this table, we add the “Unique Authors” column to illustrate how many authors appear on more than one paper (i.e., if the columns “Authors” and “Unique Authors” would have the same number, then all authors would only appear on exactly one paper).

Table 1: Papers submitted and accepted by subject area

Subject Area	Submitted	Accepted	Accepted %
MIR tasks	94	35	37.2%
MIR fundamentals and methodology	38	13	34.2%
Musical features and properties	36	12	33.3%
Evaluation, datasets, reproducibility	34	13	38.2%
Generative Tasks	32	15	46.9%
Knowledge-driven approaches to MIR	30	9	30.0%
Applications	28	8	28.6%
Computational musicology	16	7	43.8%
Human-centered MIR	16	2	12.5%
Creativity	12	6	50.0%
Philosophical and ethical discussions	6	3	50.0%
MIR and machine learning for musical acoustics	4	0	0.0%
Total	346	123	35.5%

Table 2: Summary of publication statistics over the 25-year-history of the ISMIR conference

Year	Location	Oral	Poster	Total	Authors	Unique Authors	<u>Authors Paper</u>	<u>Unique Authors Paper</u>
2000	Plymouth	19	16	35	68	63	1.9	1.8
2001	Indiana	25	16	41	100	86	2.4	2.1
2002	Paris	35	22	57	129	117	2.3	2.1
2003	Baltimore	26	24	50	132	111	2.6	2.2
2004	Barcelona	61	44	105	252	214	2.4	2.0
2005	London	57	57	114	316	233	2.8	2.0
2006	Victoria	59	36	95	246	198	2.6	2.1
2007	Vienna	62	65	127	361	267	2.8	2.1
2008	Philadelphia	24	105	105	296	253	2.8	2.4
2009	Kobe	38	85	123	375	292	3.0	2.4
2010	Utrecht	24	86	110	314	263	2.0	2.4
2011	Miami	36	97	133	395	322	3.0	2.4
2012	Porto	36	65	101	324	264	3.2	2.6
2013	Curitiba	31	67	98	395	236	3.0	2.4
2014	Taipei	33	73	106	343	271	3.2	2.6
2015	Málaga	24	90	114	370	296	3.2	2.6
2016	New York	25	88	113	341	270	3.0	2.4
2017	Suzhou	24	73	97	324	248	3.3	2.6
2018	Paris			104	337	265	3.2	2.5
2019	Delft			114	390	315	3.4	2.8
2020	Virtual			115	426	343	3.7	3.0
2021	Virtual			104	334	269	3.2	2.6
2022	Bengaluru			113	423	355	3.8	3.0
2023	Milan			103	374	311	3.6	3.0
2024	San Francisco			123	497	433	4.0	3.5

Open Review

In recent years, the ISMIR community has discussed whether implementing an Open Review process would be of benefit. This year, the ISMIR 2024 General Chairs and Scientific Program Chairs, in consultation with the ISMIR Board, launched a pilot of Open Review, in which reviews of accepted papers were published when all authors, reviewers, and meta-reviewers of a given paper opted in. To do so, we discussed a potential Open Review framework, described as follows:

- **Consent for Publication:** We sought consent from all reviewers, meta-reviewers, and authors to publish their reviews and meta-reviews during the peer review process.
- **Anonymity:** All content was published anonymously to protect the identities of the reviewers and meta-reviewers.
- **Soft Release:** We conducted a soft release of the reviews to gauge the community’s reception.
- **Positive Reception:** The response was overwhelmingly positive. A total of 32 papers had all authors, reviewers, and meta-reviewers consenting to share their reviews and meta-reviews. These are now published and available in the official program.
- **Historical Milestone:** This marks the first time ISMIR has published some of its reviews, setting a precedent for future transparency in the review process.

Best Paper Awards

The selection process for Best Paper Awards varies from year to year, depending on the organizers of the conference. One goal of the ISMIR 2024 selection process was to come up with a model that can be applied more consistently in the future. Given the growth in the number of papers, we wanted to give awards to a more reasonable number of papers to acknowledge people's contributions. We recommend aiming to award Best Paper Awards to the top 3% of accepted papers and Honorable Mentions to the remaining papers in the top 10%. Unlike some past ISMIR conferences, we do not distinguish between Best Paper and Best Student Paper.

This year, the SPC selected 33 candidate papers (approximately 10% of all submissions) based on reviewers' and meta-reviewers' nominations as well as the paper review scores and comments. This year, given 123 accepted papers, we aimed for 3–4 Best Papers and 7–8 honorable mentions. We awarded 3 papers for Best Paper Award and 4 papers for Honorable Mention. The final selections were made by the SPCs, all of whom were MIR researchers who had no conflict of interest with any of the award candidates.

The following papers received the Best Paper Awards:

- *Six Dragons Fly Again: Reviving 15th-Century Korean Court Music with Transformers and Novel Encoding*, Danbinaerin Han, Mark R. H. Gotham, DongMin Kim, Hannah Park, Sihun Lee, Dasaem Jeong
- *ST-ITO: Controlling Audio Effects for Style Transfer with Inference-Time Optimization*, Christian J. Steinmetz, Shubhr Singh, Marco Comunità, Ilias Ibyahya, Shanxin Yuan, Emmanouil Benetos, Joshua D. Reiss
- *MuChoMusic: Evaluating Music Understanding in Multimodal Audio-Language Models*, Benno Weck, Ilaria Manco, Emmanouil Benetos, Elio Quinton, George Fazekas, Dmitry Bogdanov

Honorable Mentions were given to the following papers:

- *Cluster and Separate: A GNN Approach to Voice and Staff Prediction for Score Engraving*, Francesco Foscarin, Emmanouil Karystinaios, Eita Nakamura, Gerhard Widmer
- *Green MIR? Investigating Computational Cost of Recent Music-AI Research in ISMIR*, Andre Holzapfel, Anna-Kaisa Kaila, Petra Jääskeläinen
- *Formal Modeling of Structural Repetition using Tree Compression*, Zeng Ren, Yannis Rammos, Martin A. Rohrmeier
- *Scoring Intervals Using Non-Hierarchical Transformer for Automatic Piano Transcription*, Yujia Yan and Zhiyao Duan

Best Paper Awards or Honorable Mention awardees will be invited to submit extended versions of their ISMIR papers to the TISMIR journal; accepted papers will be published at no cost to the authors.

Best Reviewer Awards

The Best Reviewer Awards were determined based on specific criteria to ensure fairness and recognition of excellence. Award recipients achieved an average rating of 3/3, indicating they exceeded expectations across at least two reviews. Alternatively, reviewers could qualify with an average rating greater than 2/3, meaning they consistently met or exceeded expectations, over a minimum of five reviews. This dual-criteria approach ensured that both consistently high-performing reviewers and those who excelled in fewer but impactful reviews were acknowledged.

Congratulations to this year's Best Reviewer Awardees (listed in alphabetical order of first name):

- Akira Maezawa
- Alon Ziv
- Christian Steinmetz
- Dasaem Jeong
- Emilia Parada-Cabaleiro
- Fabian-Robert Stöter
- Jaehun Kim
- Lorenzo Porcaro
- Maximilian Schmitt
- Pablo Alonso-Jiménez
- Simon Durand
- Simon Schwär
- William Wilson

Test of Time Award

To commemorate the 25th anniversary of ISMIR, we introduced the Test of Time Award for the first time. This award aims to recognize research that has had a significant long-term impact on MIR.

Eligible candidates for this award included all ISMIR papers published from 2000, the year of the first ISMIR conference, to 2004. A total of 261 papers were considered. This year's recipient was selected by the ISMIR 2024 General Chairs.

For ISMIR 2024, the inaugural Test of Time Award was given to the paper titled "Automatic Musical Genre Classification of Audio Signals" by George Tzanetakis, George Essl, and Perry Cook, published at ISMIR 2001. This paper has had a profound impact on MIR and beyond, defining classic MIR tasks, remaining highly relevant to this day.

During the awards ceremony, the first author of the winning paper delivered a 10-minute presentation, highlighting the paper's contributions and its enduring influence on the field.

Accessibility

One goal of ISMIR 2024 was to make accessibility an intentional facet of the conference planning: We wanted to ensure that all attendees experienced a sense of inclusivity and belonging, whether attending in person or online. To this end, ISMIR 2024 welcomed its first-ever Accessibility Chair, who worked cross-functionally to ensure a number of accessibility factors were implemented for the conference:

- **Publication** best practices included recommendations and instructions for alt-text, color choice, and fonts; a captioning requirement and instructions for video submissions; and captioning of any session recordings that will be posted online.
- Various **onsite accessibility features** were confirmed with the conference venue including wheelchair accessibility, assistive listening devices available upon request, service animal accommodation, food labels, gender-neutral bathrooms, reader boards on every floor, and nearby public transportation options.
- Attendees' **name tags** included pronouns and language(s) spoken in order to communicate identities and how attendees would like to be addressed, and help attendees find others who speak their language.
- The conference supported onsite attendees who wished to take safety measures with regard to **COVID-19** by providing written guidelines for all attendees as well as freely available COVID-19 tests, KN95 masks, and hand sanitizer at the conference venue.
- Finally, a **virtual registration discount program** was implemented separately from the Grants program in order to make virtual attendance more accessible.

Diversity & Inclusion

The mission of Diversity, Equity, and Inclusion (DEI) work at ISMIR is to highlight the research contributions of MIR researchers from marginalized and underrepresented intersectional identities. As part of that work, the DEI efforts at the ISMIR conference seek to support, encourage, and advocate for researchers in MIR who occupy marginalized identities. Initiatives include highlighting work by MIR researchers through keynotes and talks, supporting conference registrations and accommodation grants at the conference hotel, bringing DEI experts from adjacent fields to help our community think more expansively, and periodically re-evaluating our efforts towards creating a more inclusive and equitable ISMIR conference and society.

This year as we thought about the 25th ISMIR, we wanted to take stock of our efforts as a conference towards inclusion and assess if we were engaging in Equity or Equality. As a result of those conversations, we decided it was time to "grow the I", shifting to the acronym at the conference level from WiMIR (meaning Women in MIR) to WIMIR (meaning Widening Inclusion in MIR), and to include many axes of diversity including geographic location and racial diversity. In truth we had already been doing that work to some degree, but we felt it was time to be more explicit.

This year, the DEI initiatives included programming both before and during the main conference. The overarching theme of this programming was the "Growing the I": Kicking off the rebranding of WiMIR → WIMIR. We had a DEI Session

in the Virtual Pre-Conference (see “Virtual Pre-Conference” section, below) and welcomed Dr. Valerie Joseph as the DEI Keynote speaker (see “Main Conference Keynotes” section, below). We introduced the new Slack Backchannel initiative to engage with our virtual attendees in a new way. We continued providing financial support through accommodation and registration grants, welcoming new attendees through Newcomer Initiatives, and mentoring prospective new authors in the New-to-ISMIR Paper Mentoring Program.

Slack Backchannel

New to ISMIR this year was the Slack Backchannel. For each of the 7 paper sessions, plus the replays, there was a host who offered running commentary on each session over Slack in real time. Paper authors were asked to submit interesting behind-the-scenes factoids about their papers, fun facts about themselves, and/or relevant background information about their work to help those new to the field contextualize their work. While not identical to a “hallway” track at the in-person conference, these hosts helped virtual attendees learn more about the presenters themselves as well as connect with in-person participants. There were numerous fun conversations had on the backchannel. We thank our backchannel hosts:

- Amélie Anglade
- Stefan Balke
- Dan Ellis
- Arthur Flexer
- Youngmoo Kim
- Katherine M. Kinnaird
- Alia Morsi
- Lalit Mohan
- Zafar Rafii
- Doug Turnbull
- Christof Weiß

Grants

As part of ISMIR’s continued commitment to supporting new and diverse voices in the community, ISMIR 2024 offered the opportunity for presenters and attendees to apply for registration, accommodation, and childcare grants. Though anyone could apply, grants were awarded based on financial need, student and Diversity & Inclusion eligibility, and availability of funds. For registration waiver grants, the following funding categories were prioritized:

- Student: Applicants enrolled in a degree-granting academic program in the 2023-2024 and/or 2024-2025 academic year(s)
- Minorities in MIR: Applicants identifying as Black, African, African-American, or an ethnic/racial minority (of the applicant’s region)
- Applicant’s professional affiliation is in a low- or middle-income country
- Queer in MIR: Applicants identifying as LGBTQIA+
- Unaffiliated researcher: Applicants who currently have no professional affiliation that will cover the conference registration fee
- Women in MIR: Applicants identifying as a woman or other gender minority
- Caregiver: Applicants seek financial support to cover childcare costs so that they may attend the conference

This year, 91 people applied for financial support from the conference, for a total nearly double that of available funds. For ISMIR 2024, we were able to support 10 virtual registrations, 30 in-person registrations, and 12 accommodation grants.

Newcomer Initiatives

Multiple Newcomer Initiatives were carried out during ISMIR 2024, with the goal of helping less-experienced attendees get the most out of the conference and gain a foothold in the MIR community.

The first of these initiatives was a “Navigating the Conference” session that took place during the virtual pre-conference program. This session, led by the Newcomer Initiatives Chair, contained information about the ISMIR Society, the field of MIR, ISMIR 2024 online resources, as well as general advice on networking and attending an academic conference.

The “Newcomer Squads” initiative assigned groups of new attendees to experienced community members for the duration of the conference. The squads were connected via Slack and conducted meetups during the conference (either virtually or in-person). The 2024 Newcomer Squads featured 8 leaders and over 75 newcomers!

Finally, the “Anonymous Question Board” initiative provided a public space for attendees to ask questions anonymously. Question were submitted via a Google Form link, and could be viewed/answered in a Google Sheet.

We thank the following individuals (listed in alphabetical order of last name) for generously volunteering to lead Newcomer Squads at this year’s conference:

- Stefan Balke
- SeungHeon Doh
- Nick Gang
- Blair Kaneshiro
- Jin Ha Lee
- Brian McFee
- Ajay Srinivasamurthy
- Daniel Wolff

Late Breaking/Demo Session

As a forum for presenting prototype systems, initial concepts, and early research results, the Late Breaking/Demo (LBD) session has been growing steadily. This year, we accepted a record total of 81 submissions for both in-person and virtual presentations. We rejected 3 submissions that failed to adhere to the submission guidelines. It is worth noting that 50 submissions were self-identified as first-time attendees to ISMIR, showcasing the importance of LBD as the entry point for newcomers to engage with the ISMIR community. Based on the feedback from last year’s LBD chairs, we explicitly instructed the in-person presenters to interact with virtual attendees towards the end of the allocated presentation time. This change aimed to make LBD a more inclusive space for both in-person and virtual attendees. Despite not being part of the official proceedings, LBD offers a unique opportunity for both senior and junior members of the ISMIR community to socialize over exciting and interesting demos.

MIREX

The Music Information Retrieval Evaluation eXchange (MIREX) was established in 2005 to provide a platform for MIR researchers to compare and discuss their results. MIREX became an annual event at the ISMIR conference but paused after 2021 due to hosting challenges. Given its significance and community interest, MIREX is being revived in 2024 with plans to modernize it by introducing new platforms, tasks, and evaluation methods to keep pace with advancements in computer music research.

In this year’s MIREX, modern MIR tasks were added to reflect new directions in the MIR community. A Call for Challenges was also released to collect new tasks that the community was interested in. Two submissions were received that year, and one was selected for the MIREX 2024 task list (Singing Voice Deepfake Detection).

Since MIREX 2024 ran on limited scalability, the final MIREX 2024 task list included eight tasks as shown in Table 1, with three traditional MIR tasks and five modern MIR tasks.

Table 3: MIREX 2024 tasks. # Teams and # Subs denote the number of teams and submissions, respectively.

	Tasks	Submission Platform	# Teams	# Sub.
Traditional MIR Tasks	Audio Chord Estimation	Forum	0	0
	Lyrics-to-Audio Alignment	Forum	1	1
	Cover Song Identification	Email	3	3
Modern MIR Tasks	Symbolic Music Generation	Email	1	1
	Music Audio Generation	Email	1	1
	Music Description & Captioning	Codabench	4	15
	Polyphonic Transcription	Forum	3	3
	Singing Voice Deepfake Detection	Codabench	4	7

Instead of using the IMIRSEL submission system, new submission platforms were explored. A new submission forum was built for submissions and discussions, but task captains could freely choose their preferred method of receiving

submissions. In the end, three tasks adopted forum submission (i.e., submitting by posting on the forum), three tasks adopted email submission to the task captain, and two tasks adopted the Codabench platform.

Codabench is an online platform to organize AI benchmarks and host custom competitions. The task captain created an automatic evaluation system that could be used to update the leaderboard in real time. Participants did not have to upload their models to the platform. Instead, they received a public test set from the task captain and ran model inference on their own computational resources. The inference results were uploaded to Codabench for metric calculation and leaderboard updates.

This new submission process had many advantages. For example, it eliminated potential issues in sharing code with the task captain or configuring the model on the task captain's environment. It also showed popularity in practice. The two tasks using Codabench received 21 submissions from eight teams in total, surpassing all other submission platforms. It should be noted that Codabench is not suitable for some tasks. The fact that the test set is shared with participants may raise potential concerns, including difficulty in distributing proprietary test sets and test set contamination. Many tasks may have to adopt other methods of submission.

Since generative tasks were included that year, the idea of subjective evaluation was also explored. The task captain of symbolic music generation created a questionnaire containing samples of the submission and baselines generated under the same conditions. Listeners were asked to score the generated pieces according to their coherency, naturalness, creativity, and musicality in a blind listening test. Twenty-two responses were received and were used to calculate the results.

Out of eight tasks in MIREX 2024, seven tasks successfully received at least one submission. Among the tasks that received submissions, four tasks reported significantly better performance compared to baselines:

- In cover song detection, ByteDance's submission achieved the best performance using a ResNet architecture with a dimensional reduction module, while other systems also achieved impressive performance, like Discog-VI from MTG-Sony.
- In music audio generation, the team S1-CodecLM achieved better overall performance compared to baselines using a transformer decoder structure with two-stage semantic tokenization.
- In music description and captioning, the team ee895 achieved the highest ROUGE-L score among all entries using Llama and the Joint Music and Language Attention (JMLA) architecture. The architecture was also used in other entries like CUHKDSP.
- In singing voice deepfake detection, the submission UNIBS1 achieved the lowest error rate on both test sets with a ResNet model that received a log-spectrogram of the vocal sound as input. Other submissions also used WavLM, SingGraph, or ensemble methods.

The Future of MIREX

We want to gradually recover the scalability of MIREX and make it a yearly event for the ISMIR community. More tasks are planned for MIREX 2025 with a new call for challenge proposals. The submission platform will be more formal with better guidance. Besides that, we also want to gradually introduce other improvements over the years.

- **Open:** To make the evaluation process transparent and open-sourced, and also to encourage (but not force) submissions to be open-sourced.
- **Interactive:** To make the evaluation process more interactive, e.g., automatic evaluation for real-time leaderboards on more tasks.
- **Modernized:** To host more emergent MIR tasks while refining traditional tasks, making MIREX keep pace with the community's rapid advancements.

Finally, the time and efforts of the ISMIR organizers, all task captains, participants, and MIREX 2024 organizers are greatly appreciated.

Unconference

An Unconference was held both during the pre-conference and on the last day of the conference, immediately following the Society Meeting. Discussion topics were proposed using the Dotstorming platform, and participants voted live using the Wooclap platform. Two rounds of discussions took place, each lasting 45 minutes. Afterward, the secretary of each discussion group presented a summary of their group's discussion results to all participants, including those attending via Zoom. The following topics were discussed:

- Safety and human rights considerations for selecting ISMIR conference venues for members of the community
- What will be the next direction in Music Research?
- Change our name from Music Information Retrieval to Music Information Research?
- Moving away from Twitter (and other questionable organizations)
- GenAI Copyright/ GenAI Cultural Diversity
- Open Review

Social Program

The Social Program this year contained the following activities:

- Concert by Camilo y Los Cruzers at the Welcome Reception during the first night at ISMIR
- Concert by Wil Blades during the official ISMIR Banquet
- Jam Session during the Banquet, with over 40 sets of performers sign ups

Additionally, the Social Chairs assembled a list of restaurants/bars/music venues near the conference hotel, which was published on the ISMIR website.

Hackathon

HAMR: Hacking Audio and Music Research was rebranded as Highlighting Audio and Music Researchathon for 2024. HAMR took place after the conference on Saturday, November 16 in San Francisco. The event applied the hackathon model to the development of new techniques for analyzing, processing, and synthesizing audio and music signals. This was a free event open to researchers and hackers from any stage in their career.

The hackathon was an all-day event with around 20 participants. Individuals and teams worked on research projects, or took time to catch up on personal work, review conference materials, and socialize and brainstorm for future projects.

The organizers thank Replicate (<https://replicate.com/>) for offering up their hackage for the ISMIR 2024 Hackathon.



Satellite Events

In addition to the main conference, ISMIR 2024 included the following satellite events hosted around the dates of the conference:

- **3rd Workshop on NLP for Music and Audio (NLP4Musa 2024)**
November 15, 2024, Oakland, CA, USA
Website: <https://sites.google.com/view/nlp4musa-2024>
- **HAMR 2024: Music and Audio Hackathon**
November 16, 2024, San Francisco, CA, USA
Website: <https://partiful.com/e/pnNRBcvgLBf0Jej9QX36>
- **6th International Workshop on Reading Music Systems (WoRMS 2024)**
November 22, 2024, Online
Website: <https://sites.google.com/view/worms2024>
- **1st Latin American Music Information Retrieval Workshop (LAMIR)**
December 9–11, 2024, Rio de Janeiro, BR
Website: <https://lamir-workshop.github.io/>

Acknowledgements

We are happy to present to you the proceedings of ISMIR 2024. The conference program was made possible thanks to the hard work of many people, including the ISMIR 2024 conference chairs, ISMIR Board members, volunteers, and the many reviewers and meta-reviewers from the program committee.

We would also like to thank our sponsors, whose contributions made this conference possible:

Diamond sponsor

- Music.AI

Platinum sponsor

- Adobe

Gold sponsors

- ByteDance
- Google
- Riffusion
- Splice
- Suno

Silver sponsors

- Deezer
- Pro Sound Effects
- Steinberg
- Universal Music Group
- Yamaha

Bronze sponsors

- Audible Magic
- BMAT
- Cochlear
- Dolby
- Kits.ai
- Netflix
- SiriusXM
- Yousician

We would like to thank the sponsors that explicitly chose to sponsor WiMIR, its grants, and its initiatives:

Patrons

- Adobe
- Deezer

Contributor

- Music.AI

Supporters

- Kits.ai
- Pro Sound Effects
- Riffusion
- Steinberg
- Yousician

ISMIR 2024 would not have been possible without the exceptional contributions of our community in response to our call for participation. The biggest acknowledgment goes to you, the researchers, presenters and participants.

Chris Donahue

Cheng-Zhi Anna Huang

Jin Ha Lee

Brian McFee

Scientific Program Chairs

Blair Kaneshiro

Gautham Mysore

Oriol Nieto

General Chairs

Table of Contents

Virtual Pre-Conference	1
Keynote Talks	3
The Advent of Quantum Computer Music	
<i>Eduardo Miranda</i>	3
The MIR Field: From Knowledge to Data-Driven, from Features to Ethical and Regulatory Considerations	
<i>Emilia Gómez</i>	4
Invited Presentations and Sessions	5
Keynote Talks	7
Towards a Fairer Approach to Generative AI Training	
<i>Ed Newton-Rex</i>	9
Listening For Diversity: The Ways in Which Critical Attention to Words Helps Move Us Closer Towards Realizing Our Full Humanity	
<i>Valerie Joseph</i>	10
Navigating the Intersection of AI and Music: Innovation, Ethics, and the Future of the Industry	
<i>Elizabeth Moody</i>	12
Status Report: AI Music in Q1 of the 21st Century	
<i>Douglas Eck</i>	13
Tutorials	15
Connecting Music Audio and Natural Language	
<i>SeungHeon Doh, Ilaria Manco, Zachary Novack, Jongwook Kim, and Ke Chen</i>	17
Exploring 25 Years of Music Information Retrieval: Perspectives and Insights	
<i>Masataka Goto, Jin Ha Lee, and Meinard Müller</i>	19
From White Noise to Symphony: Diffusion Models for Music and Sound	
<i>Chieh-Hsin Lai, Koichi Saito, Bac Nguyen Cong, Yuki Mitsufuji, and Stefano Ermon</i>	21
Humans at the Center of MIR: Human-subjects Research Best Practices	
<i>Claire Arthur, Nat Condit-Schultz, David R. W. Sears, John Ashley Burgoyne, and Josuha Albrecht</i>	22
Deep Learning 101 for Audio-based MIR	
<i>Geoffroy Peeters, Gabriel Meseguer Brocal, Alain Riou, and Stefan Lattner</i>	24
Lyrics and Singing Voice Processing in Music Information Retrieval: Analysis, Alignment, Transcription and Applications	
<i>Daniel Stoller, Emir Demirel, Kento Watanabe, and Brendan O'Connor</i>	26

Special Session: Remembering Don Byrd	29
Creative Practice Sessions	33
Industry Sessions	37
Live Online Sessions	41
TISMIR Presentations	47
Papers – Session I	51
Formal Modeling of Structural Repetition Using Tree Compression <i>Zeng Ren, Yannis Rammos, Martin A. Rohrmeier</i>	53
Saraga Audiovisual: A Large Multimodal Open Data Collection for the Analysis of Carnatic Music <i>Adithi Shankar, Genís Plaja-Roglans, Thomas Nuttall, Martín Rocamora, Xavier Serra</i>	61
X-Cover: Better Music Version Identification System by Integrating Pretrained ASR Model <i>Xingjian Du, Mingyu Liu, Pei Zou, Xia Liang, Zijie Wang, Huidong Liang, Bilei Zhu</i>	70
Harmonic and Transposition Constraints Arising From the Use of the Roland TR-808 Bass Drum <i>Emmanuel Deruty</i>	78
FruitsMusic: A Real-World Corpus of Japanese Idol-Group Songs <i>Hitoshi Suda, Shunsuke Yoshida, Tomohiko Nakamura, Satoru Fukayama, Jun Ogata</i>	86
Classical Guitar Duet Separation Using GuitarDuets - A Dataset of Real and Synthesized Guitar Recordings <i>Marios Glytsos, Christos Garoufis, Athanasia Zlatintsi, Petros Maragos</i>	95
Can LLMs "Reason" in Music? an Evaluation of LLMs' Capability of Music Understanding and Generation <i>Ziya Zhou, Yuhang Wu, Zhiyue Wu, Xinyue Zhang, Ruibin Yuan, Yinghao Ma, Lu Wang, Emmanouil Benetos, Wei Xue, Yike Guo</i>	103
Music2Latent: Consistency Autoencoders for Latent Audio Compression <i>Marco Pasini, Stefan Lattner, George Fazekas</i>	111
Robust and Accurate Audio Synchronization Using Raw Features From Transcription Models <i>Johannes Zeitler, Ben Maman, Meinard Müller</i>	120
Harnessing the Power of Distributions: Probabilistic Representation Learning on Hypersphere for Multimodal Music Information Retrieval <i>Takayuki Nakatsuka, Masahiro Hamasaki, Masataka Goto</i>	128
Towards Automated Personal Value Estimation in Song Lyrics <i>Andrew M. Demetriou, Jaehun Kim, Sandy Manolios, Cynthia Liem</i>	137
Audio Conditioning for Music Generation via Discrete Bottleneck Features <i>Simon Rouard, Yossi Adi, Jade Copet, Axel Roebel, Alexandre Défossez</i>	146
Variation Transformer: New Datasets, Models, and Comparative Evaluation for Symbolic Music Variation Generation <i>Chenyu Gao, Federico Reuben, Tom Collins</i>	154
Automatic Detection of Moral Values in Music Lyrics <i>Vjosa Preniqi, Iacopo Ghinassi, Julia Ive, Kyriaki Kalimeri, Charalampos Saitis</i>	164

Semi-Supervised Piano Transcription Using Pseudo-Labeling Techniques <i>Sebastian Strahl, Meinard Müller</i>	173
Note-Level Transcription of Choral Music <i>Huiran Yu, Zhiyao Duan</i>	182
Learning Multifaceted Self-Similarity Over Time and Frequency for Music Structure Analysis <i>Tsung-Ping Chen, Kazuyoshi Yoshii</i>	189
A Contrastive Self-Supervised Learning Scheme for Beat Tracking Amenable to Few-Shot Learning <i>Antonin Gagneré, Slim Essid, Geoffroy Peeters</i>	198
Using Pairwise Link Prediction and Graph Attention Networks for Music Structure Analysis <i>Morgan Buisson, Brian McFee, Slim Essid</i>	207
Papers – Session II	215
Six Dragons Fly Again: Reviving 15th-Century Korean Court Music With Transformers and Novel Encoding <i>Danbinaerin Han, Mark R. H. Gotham, DongMin Kim, Hannah Park, Sihun Lee, Dasaem Jeong</i>	217
Lessons Learned From a Project to Encode Mensural Music on a Large Scale With Optical Music Recognition <i>David Rizo, Jorge Calvo-Zaragoza, Patricia García-Iasci, Teresa Delgado-Sánchez</i>	225
The Changing Sound of Music: An Exploratory Corpus Study of Vocal Trends Over Time <i>Elena Georgieva, Pablo Ripollés, Brian McFee</i>	232
Music Proofreading With RefinPaint: Where and How to Modify Compositions Given Context <i>Pedro Ramoneda, Martín Rocamora, Taketo Akama</i>	240
Notewise Evaluation for Music Source Separation: A Case Study for Separated Piano Tracks <i>Yigitcan Özer, Hans-Ulrich Berendes, Vlora Arifi-Müller, Fabian-Robert Stöter, Meinard Müller</i>	248
Automatic Estimation of Singing Voice Musical Dynamics <i>Jyoti Narang, Nazif Can Tamer, Viviana De La Vega, Xavier Serra</i>	256
Joint Audio and Symbolic Conditioning for Temporally Controlled Text-to-Music Generation <i>Or Tal, Alon Ziv, Itai Gat, Felix Kreuk, Yossi Adi</i>	264
Diff-a-Riff: Musical Accompaniment Co-Creation via Latent Diffusion Models <i>Javier Nistal, Marco Pasini, Cyran Aouameur, Maarten Grachten, Stefan Lattner</i>	272
Exploring Internet Radio Across the Globe With the MIRAGE Online Dashboard <i>Ngan V.T. Nguyen, Elizabeth Acosta, Tommy Dang, David Sears</i>	281
MIDI-to-Tab: Guitar Tablature Inference via Masked Language Modeling <i>Andrew C. Edwards, Xavier Riley, Pedro Pereira Sarmiento, Simon Dixon</i>	288
Transcription-Based Lyrics Embeddings: Simple Extraction of Effective Lyrics Embeddings From Audio <i>Jaehun Kim, Florian Henkel, Camilo Landau, Samuel E. Sandberg, Andreas F. Ehmann</i>	295
A Method for MIDI Velocity Estimation for Piano Performance by a U-Net With Attention and FiLM <i>Hyon Kim, Xavier Serra</i>	304
MusiConGen: Rhythm and Chord Control for Transformer-Based Text-to-Music Generation <i>Yun-Han Lan, Wen-Yi Hsiao, Hao-Chung Cheng, Yi-Hsuan Yang</i>	311
End-to-End Piano Performance-MIDI to Score Conversion With Transformers <i>Tim Beyer, Angela Dai</i>	319

From Real to Cloned Singer Identification <i>Dorian Desblancs, Gabriel Meseguer-Brocal, Romain Hennequin, Manuel Moussallam</i>	327
Emotion-Driven Piano Music Generation via Two-Stage Disentanglement and Functional Representation <i>Jingyue Huang, Ke Chen, Yi-Hsuan Yang</i>	335
Efficient Adapter Tuning for Joint Singing Voice Beat and Downbeat Tracking With Self-Supervised Learning Features <i>Jiajun Deng, Yaolong Ju, Jing Yang, Simon Lui, Xunying Liu</i>	343
Which Audio Features Can Predict the Dynamic Musical Emotions of Both Composers and Listeners? <i>Eun Ji Oh, Hyunjae Kim, Kyung Myun Lee</i>	352
Exploring Musical Roots: Applying Audio Embeddings to Empower Influence Attribution for a Generative Music Model <i>Julia Barnett, Hugo Flores García, Bryan Pardo</i>	360
Papers – Session III	369
Green MIR? Investigating Computational Cost of Recent Music-Ai Research in ISMIR <i>Andre Holzapfel, Anna-Kaisa Kaila, Petra Jääskeläinen</i>	371
Field Study on Children’s Home Piano Practice: Developing a Comprehensive System for Enhanced Student-Teacher Engagement <i>Seikoh Fukuda, Yuko Fukuda, Masamichi Hosoda, Ami Motomura, Eri Sasao, Masaki Matsubara, Masahiro Niitsuma</i>	381
Inner Metric Analysis as a Measure of Rhythmic Syncopation <i>Brian Bemman, Justin Christensen</i>	389
HAISP: A Dataset of Human-AI Songwriting Processes From the AI Song Contest <i>Lidia J. Morris, Rebecca Leger, Michele Newman, John Ashley Burgoyne, Ryan Groves, Natasha Mangal, Jin Ha Lee</i>	397
Cue Point Estimation Using Object Detection <i>Giulia Argüello, Luca A. Lanzendörfer, Roger Wattenhofer</i>	405
The ListenBrainz Listens Dataset <i>Kartik Ohri, Robert Kaye</i>	413
SpecMaskGIT: Masked Generative Modeling of Audio Spectrogram for Efficient Audio Synthesis and Beyond <i>Marco Comunità, Zhi Zhong, Akira Takahashi, Shiqi Yang, Mengjie Zhao, Koichi Saito, Yukara Ikemiya, Takashi Shibuya, Shusuke Takahashi, Yuki Mitsufuji</i>	420
Long-Form Music Generation With Latent Diffusion <i>Zach Evans, Julian D. Parker, CJ Carr, Zachary Zukowski, Josiah Taylor, Jordi Pons</i>	429
Composer’s Assistant 2: Interactive Multi-Track MIDI Infilling With Fine-Grained User Control <i>Martin E. Malandro</i>	438
Towards Zero-Shot Amplifier Modeling: One-to-Many Amplifier Modeling via Tone Embedding Control <i>Yu-Hua Chen, Yen-Tung Yeh, Yuan-Chiao Cheng, Jui-Te Wu, Yu-Hsiang Ho, Jyh-Shing Roger Jang, Yi-Hsuan Yang</i>	446
Mel-RoFormer for Vocal Separation and Vocal Melody Transcription <i>Ju-Chiang Wang, Wei-Tsung Lu, Jitong Chen</i>	454

Unsupervised Synthetic-to-Real Adaptation for Optical Music Recognition <i>Noelia N. Luna-Barahona, Adrián Roselló, María Alfaro-Contreras, David Rizo, Jorge Calvo-Zaragoza</i>	462
MMT-BERT: Chord-Aware Symbolic Music Generation Based on Multitrack Music Transformer and MusicBERT <i>Jinlong Zhu, Keigo Sakurai, Ren Togo, Takahiro Ogawa, Miki Haseyama</i>	470
Discogs-VI: A Musical Version Identification Dataset Based on Public Editorial Metadata <i>Recep Oguz Araz, Xavier Serra, Dmitry Bogdanov</i>	478
Who’S Afraid of the ‘Artyfyshall Byrd’? Historical Notions and Current Challenges of Musical Artificiality <i>Nicholas Cornia, Bruno Forment</i>	486
End-to-End Automatic Singing Skill Evaluation Using Cross-Attention and Data Augmentation for Solo Singing and Singing With Accompaniment <i>Yaolong Ju, Chun Yat Wu, Betty Cortiñas Lorenzo, Jing Yang, Jiajun Deng, Fan Fan, Simon Lui</i>	493
Papers – Session IV	501
Cluster and Separate: A GNN Approach to Voice and Staff Prediction for Score Engraving <i>Francesco Foscarin, Emmanouil Karystinaios, Eita Nakamura, Gerhard Widmer</i>	503
From Audio Encoders to Piano Judges: Benchmarking Performance Understanding for Solo Piano <i>Huan Zhang, Jinhua Liang, Simon Dixon</i>	511
Towards Explainable and Interpretable Musical Difficulty Estimation: A Parameter-Efficient Approach <i>Pedro Ramoneda, Vsevolod E. Eremenko, Alexandre D’Hooge, Emilia Parada-Cabaleiro, Xavier Serra</i>	520
Purposeful Play: Evaluation and Co-Design of Casual Music Creation Applications With Children <i>Michele Newman, Lidia J. Morris, Jun Kato, Masataka Goto, Jason Yip, Jin Ha Lee</i>	529
El Bongosero: A Crowd-Sourced Symbolic Dataset of Improvised Hand Percussion Rhythms Paired With Drum Patterns <i>Nicholas Evans, Behzad Haki, Daniel Gómez-Marín, Sergi Jordà</i>	540
Utilizing Listener-Provided Tags for Music Emotion Recognition: A Data-Driven Approach <i>Joanne Affolter, Martin A. Rohrmeier</i>	547
PiCoGen2: Piano Cover Generation With Transfer Learning Approach and Weakly Aligned Data <i>Chih-Pin Tan, Hsin Ai, Yi-Hsin Chang, Shuen-Huei Guan, Yi-Hsuan Yang</i>	555
Diff-MST: Differentiable Mixing Style Transfer <i>Soumya Sai Vanka, Christian J. Steinmetz, Jean-Baptiste Rolland, Joshua D. Reiss, George Fazekas</i>	563
Semi-Supervised Contrastive Learning of Musical Representations <i>Julien PM Guinot, Elio Quinton, George Fazekas</i>	571
Improved Symbolic Drum Style Classification With Grammar-Based Hierarchical Representations <i>Léo Géré, Nicolas Audebert, Philippe Rigaux</i>	580
Nested Music Transformer: Sequentially Decoding Compound Tokens in Symbolic Music and Audio Generation <i>Jiwoo Ryu, Hao-Wen Dong, Jongmin Jung, Dasaem Jeong</i>	588
Continual Learning for Music Classification <i>Pedro González-Barrachina, María Alfaro-Contreras, Jorge Calvo-Zaragoza</i>	596
TheGlueNote: Learned Representations for Robust and Flexible Note Alignment <i>Silvan Peter, Gerhard Widmer</i>	603

GAPS: A Large and Diverse Classical Guitar Dataset and Benchmark Transcription Model <i>Xavier Riley, Zixun Guo, Andrew C. Edwards, Simon Dixon</i>	611
A Kalman Filter Model for Synchronization in Musical Ensembles <i>Hugo T. Carvalho, Min Susan Li, Massimiliano Di Luca, Alan M. Wing</i>	618
Stem-JEPA: A Joint-Embedding Predictive Architecture for Musical Stem Compatibility Estimation <i>Alain Riou, Stefan Lattner, Gaëtan Hadjeres, Michael Anslow, Geoffroy Peeters</i>	625
Audio Prompt Adapter: Unleashing Music Editing Abilities for Text-to-Music With Lightweight Finetuning <i>Fang Duo Tsai, Shih-Lun Wu, Haven Kim, Bo-Yu Chen, Hao-Chung Cheng, Yi-Hsuan Yang</i>	634
MelodyT5: A Unified Score-to-Score Transformer for Symbolic Music Processing <i>Shangda Wu, Yashan Wang, Xiaobing Li, Feng Yu, Maosong Sun</i>	642
GraphMuse: A Library for Symbolic Music Graph Processing <i>Emmanouil Karystinaios, Gerhard Widmer</i>	651
Papers – Session V	659
ST-ITO: Controlling Audio Effects for Style Transfer With Inference-Time Optimization <i>Christian J. Steinmetz, Shubhr Singh, Marco Comunità, Ilias Ibyahya, Shanxin Yuan, Emmanouil Benetos, Joshua D. Reiss</i>	661
ComposerX: Multi-Agent Symbolic Music Composition With LLMs <i>Qixin Deng, Qikai Yang, Ruibin Yuan, Yipeng Huang, Yi Wang, Xubo Liu, Zeyue Tian, Jiahao Pan, Ge Zhang, Hanfeng Lin, Yizhi Li, Yinghao Ma, Jie Fu, Chenghua Lin, Emmanouil Benetos, Wenwu Wang, Guangyu Xia, Wei Xue, Yike Guo</i>	669
Do Music Generation Models Encode Music Theory? <i>Megan Wei, Michael Freeman, Chris Donahue, Chen Sun</i>	680
PolySinger: Singing-Voice to Singing-Voice Translation From English to Japanese <i>Silas Antonisen, Iván López-Espejo</i>	688
On the Validity of Employing ChatGPT for Distant Reading of Music Similarity <i>Arthur Flexer</i>	697
Sanidha: A Studio Quality Multi-Modal Dataset for Carnatic Music <i>Venkatakrishnan Vaidyanathapuram Krishnan, Noel Alben, Anish A. Nair, Nathaniel Condit-Schultz</i>	705
Between the AI and Me: Analysing Listeners’ Perspectives on AI- and Human-Composed Progressive Metal Music <i>Pedro Pereira Sarmiento, Jackson J. Loth, Mathieu Barthet</i>	713
Combining Audio Control and Style Transfer Using Latent Diffusion <i>Nils Demerlé, Philippe Esling, Guillaume Doras, David Genova</i>	721
Computational Analysis of Yaredawi YeZema Silt in Ethiopian Orthodox Tewahedo Church Chants <i>Mequanent Argaw Muluneh, Yan-Tsung Peng, Li Su</i>	729
Lyrics Transcription for Humans: A Readability-Aware Benchmark <i>Ondřej Čířka, Hendrik Schreiber, Luke Miner, Fabian-Robert Stöter</i>	737
A Critical Survey of Research in Music Genre Recognition <i>Owen Green, Bob L. T. Sturm, Georgina Born, Melanie Wald-Fuhrmann</i>	745
Content-Based Controls for Music Large Language Modeling <i>Liwei Lin, Gus Xia, Junyan Jiang, Yixiao Zhang</i>	783

Exploring the Inner Mechanisms of Large Generative Music Models <i>Marcel A. Vélez Vásquez, Charlotte Pouw, John Ashley Burgoyne, Willem Zuidema</i>	791
Quantitative Analysis of Melodic Similarity in Music Copyright Infringement Cases <i>Saebyul Park, Halla Kim, Jiye Jung, Juyong Park, Jeounghoon Kim, Juhan Nam</i>	799
Robust Lossy Audio Compression Identification <i>Hendrik Vincent Koops, Gianluca Micchi, Elio Quinton</i>	807
RNBert: Fine-Tuning a Masked Language Model for Roman Numeral Analysis <i>Malcolm Sailor</i>	814
Papers – Session VI	823
MuChoMusic: Evaluating Music Understanding in Multimodal Audio-Language Models <i>Benno Weck, Ilaria Manco, Emmanouil Benetos, Elio Quinton, George Fazekas, Dmitry Bogdanov</i>	825
Human Pose Estimation for Expressive Movement Descriptors in Vocal Musical Performances <i>Sujoy Roychowdhury, Preeti Rao, Sharat Chandran</i>	834
Enhancing Predictive Models of Music Familiarity With EEG: Insights From Fans and Non-Fans of K-Pop Group NCT127 <i>Seokbeom Park, Hyunjae Kim, Kyung Myun Lee</i>	842
Mosaikbox: Improving Fully Automatic DJ Mixing Through Rule-Based Stem Modification and Precise Beat-Grid Estimation <i>Robert Sowula, Peter Knees</i>	850
MidiCaps: A Large-Scale MIDI Dataset With Text Captions <i>Jan Melechovsky, Abhinaba Roy, Dorien Herremans</i>	858
A New Dataset, Notation Software, and Representation for Computational Schenkerian Analysis <i>Stephen Ni-Hahn, Weihan Xu, Zirui Yin, Rico Zhu, Simon Mak, Yue Jiang, Cynthia Rudin</i>	866
DITTO-2: Distilled Diffusion Inference-Time T-Optimization for Music Generation <i>Zachary Novack, Julian McAuley, Taylor Berg-Kirkpatrick, Nicholas J. Bryan</i>	874
The Concatenator: A Bayesian Approach to Real Time Concatenative Mosaicing <i>Christopher J. Tralie, Ben Cantil</i>	882
Deep Recombinant Transformer: Enhancing Loop Compatibility in Digital Music Production <i>Muhammad Taimoor Haseeb, Ahmad Hammoudeh, Gus Xia</i>	890
I Can Listen but Cannot Read: An Evaluation of Two-Tower Multimodal Systems for Instrument Recognition <i>Yannis Vasilakis, Rachel Bittner, Johan Pauwels</i>	897
Streaming Piano Transcription Based on Consistent Onset and Offset Decoding With Sustain Pedal Detection <i>Weixing Wei, Jiahao Zhao, Yulun Wu, Kazuyoshi Yoshii</i>	906
Towards Universal Optical Music Recognition: A Case Study on Notation Types <i>Juan Carlos Martinez-Sevilla, David Rizo, Jorge Calvo-Zaragoza</i>	914
Controlling Surprisal in Music Generation via Information Content Curve Matching <i>Mathias Rose Bjare, Stefan Lattner, Gerhard Widmer</i>	922
Toward a More Complete OMR Solution <i>Guang Yang, Muru Zhang, Lin Qiu, Yanming Wan, Noah A. Smith</i>	930

Augment, Drop & Swap: Improving Diversity in LLM Captions for Efficient Music-Text Representation Learning <i>Ilaria Manco, Justin Salamon, Oriol Nieto</i>	938
Music Discovery Dialogue Generation Using Human Intent Analysis and Large Language Models <i>Seunghoon Doh, Keunwoo Choi, Daeyong Kwon, Taesoo Kim, Juhan Nam</i>	946
STONE: Self-Supervised Tonality Estimator <i>Yuxuan Kong, Vincent Lostanlen, Gabriel Meseguer-Brocal, Stella Wong, Mathieu Lagrange, Romain Hennequin</i>	954
Beat This! Accurate Beat Tracking Without DBN Postprocessing <i>Francesco Foscarin, Jan Schlüter, Gerhard Widmer</i>	962
Papers – Session VII	971
Scoring Time Intervals Using Non-Hierarchical Transformer for Automatic Piano Transcription <i>Yujia Yan, Zhiyao Duan</i>	973
PerTok: Expressive Encoding and Modeling of Symbolic Musical Ideas and Variations <i>Julian Lenz, Anirudh Mani</i>	981
Looking for Tactus in All the Wrong Places: Statistical Inference of Metric Alignment in Rap Flow <i>Nathaniel Condit-Schultz</i>	989
Exploring GPT’s Ability as a Judge in Music Understanding <i>Kun Fang, Ziyu Wang, Gus Xia, Ichiro Fujinaga</i>	996
Towards Assessing Data Replication in Music Generation With Music Similarity Metrics on Raw Audio <i>Roser Batlle-Roca, Wei-Hsiang Liao, Xavier Serra, Yuki Mitsufuji, Emilia Gómez</i>	1004
Generating Sample-Based Musical Instruments Using Neural Audio Codec Language Models <i>Shahan Nercessian, Johannes Imort, Ninon Devis, Frederik Blang</i>	1012
Hierarchical Generative Modeling of Melodic Vocal Contours in Hindustani Classical Music <i>Nithya Nadig Shikarpur, Krishna Maneesha Dendukuri, Yusong Wu, Antoine Caillon, Cheng-Zhi Anna Huang</i>	1020
SymPAC: Scalable Symbolic Music Generation With Prompts and Constraints <i>Haonan Chen, Jordan B. L. Smith, Janne Spijkervet, Ju-Chiang Wang, Pei Zou, Bochen Li, Qiuqiang Kong, Xingjian Du</i>	1029
Unsupervised Composable Representations for Audio <i>Giovanni Bindi, Philippe Esling</i>	1037
Lyrical Speaking: Exploring the Link Between Lyrical Emotions, Themes and Depression Risk <i>Pavani B. Chowdary, Bhavyajeet Singh, Rajat Agarwal, Vinoo Alluri</i>	1046
A Stem-Agnostic Single-Decoder System for Music Source Separation Beyond Four Stems <i>Karn N. Watcharasupat, Alexander Lerch</i>	1051
In-Depth Performance Analysis of the ADTOF-Based Algorithm for Automatic Drum Transcription <i>Mickael Zehren, Marco Alunno, Paolo Bientinesi</i>	1060
Towards Musically Informed Evaluation of Piano Transcription Models <i>Patricia Hu, Lukáš Samuel Marták, Carlos Eduardo Cancino-Chacón, Gerhard Widmer</i>	1068
Using Item Response Theory to Aggregate Music Annotation Results of Multiple Annotators <i>Tomoyasu Nakano, Masataka Goto</i>	1076

Just Label the Repeats for In-the-Wild Audio-to-Score Alignment
Irmak Bukey, Michael Feffer, Chris Donahue 1085

Investigating Time-Line-Based Music Traditions With Field Recordings: A Case Study of Candomblé Bell
Patterns
Lucas S. Maia, Richa Namballa, Martín Rocamora, Magdalena Fuentes, Carlos Guedes 1093

Author Index **1101**

Virtual Pre-Conference

The Virtual Pre-Conference was a new initiative this year and took place completely online from October 28–30 across global time zones. The Virtual Pre-Conference was organized with the goals of giving all attendees an opportunity to begin networking—whether attending the main conference online or in person; and helping newcomers gain context on the ISMIR conference and broader community.

The ISMIR 2024 Virtual Pre-Conference was organized by the General Chairs and Virtual Chair, and featured two keynote talks, invited presentations from conference and community organizers, social events, and informal topic-based discussion sessions.

Keynote Talks

Keynote Talk – 1

The Advent of Quantum Computer Music

Eduardo Miranda

University of Plymouth, UK

Abstract

Quantum computing technology is developing at a fast pace. The impact of quantum computing on the music industry is inevitable. The emerging field of Quantum Computer Music investigates and develops applications and methods to process music using quantum computing technology. This talk will discuss examples of approaches to leverage quantum computing to learn, process and generate music. The methods discussed range from rendering music using data from physical quantum mechanical systems and quantum mechanical simulations to computational quantum algorithms to generate music, including quantum AI. The ambition to develop techniques to encode audio quantumly for making sound synthesisers and audio signal processing systems is also discussed.

Biography

Eduardo Reck Miranda is a classically trained composer and computer scientist. He has composed for renowned ensembles such as the BBC Concert Orchestra, Scottish Chamber Orchestra and London Sinfonietta. He is a Professor of Computer Music at the University of Plymouth, UK, and works with at Moth, a quantum technology company building the next era of music, gaming and the arts. Prof Miranda published over 100 research papers in learned journals and 16 books. He is world-renowned for his groundbreaking work in AI and music. He is a pioneer of quantum computing with a focus on creativity and music composition. His latest book, *Quantum Computer Music*, comprising a collection of chapters by leading practitioners in the field, was published in 2022 by Springer Nature.

Keynote Talk – 2

The MIR Field: From Knowledge to Data-Driven, from Features to Ethical and Regulatory Considerations

Emilia Gómez

European Commission's Joint Research Centre, ES

Abstract

This talk focuses on audio-based music information retrieval (MIR) and reflects on the origins of the field, the different MIR eras, and the recent developments. I will first focus on the paradigm shift from knowledge-driven to data-driven algorithmic design, thanks to recent developments in machine learning. After that, I will discuss the current challenges that the MIR field addresses and the current and future research challenges, notably on the social and ethical impact of MIR algorithmic systems.

Biography

Dr. Emilia Gómez (MSc. Telecommunication Engineering, PhD in Computer Science, Full professor accreditation) is a senior scientist at the European Commission's Joint Research Centre, where she leads the Human Behaviour and Machine Intelligence (HUMAIN) team that provides scientific support to EU AI policies as part of the European Centre for Algorithmic Transparency, notably the AI Act and the Digital Services Act. She is also a guest professor in Music Technology at Universitat Pompeu Fabra in Barcelona, Spain.

Dr. Gómez has a long academic experience in the field of Music Information Retrieval, where she has contributed to different approaches for music content description, notably in pitch-content description. Starting from the music domain, she now studies the impact of AI in human behaviour, notably how AI affects jobs, decisions, fundamental rights and children. She was the first female president of ISMIR, is currently a member of the OECD One AI expert group, an ELLIS (European Laboratory for Learning and Intelligent systems) fellow, and her work has been recognized by means of citations and honors, e.g. EUWomen4Future, Red Cross Award to Humanitarian Technologies or ICREA Academia.

Invited Presentations and Sessions

DEI and “Growing the I”

Presenter: Katherine M. Kinnaird (ISMIR 2024 DEI Chair)

Ideation Session

Session host: Vinoo Alluri (ISMIR 2024 Virtual Chair)

ISMIR Board

Presenters: Emmanouil Benetos, Carlos Cancino-Chacón, Ajay Srinivasamurthy (representing the ISMIR Board)

ISMIR Ethics Working Group

Presenters: Fabio Morreale, Pedro Sarmiento (representing the ISMIR Ethics Working Group Organizers)

Open Review at ISMIR

Presenters: Magdalena Fuentes, Blair Kaneshiro, Oriol Nieto, Geoffroy Peeters (representing the ISMIR Open Review Working Group)

Navigating the ISMIR Conference

Presenter: Nick Gang (ISMIR 2024 Newcomer Initiatives Chair)

New-to-ISMIR Paper Mentoring Program

Presenter: Ajay Srinivasamurthy (representing the ISMIR 2024 New-to-ISMIR Paper Mentoring Program Chairs)

TISMIR

Presenter: Meinard Müller (representing the TISMIR Editors-in-Chief)

Unconference

Session host: Geoffroy Peeters (ISMIR 2024 Unconference Chair)

WiMIR Mentoring Program

Presenters: Yun-Ning (Amy) Hung, Zafar Rafii (representing the WiMIR Mentoring Program Organizers)

Keynote Talks

Keynote Talk – 1

Towards a Fairer Approach to Generative AI Training

Ed Newton-Rex

Fairly Trained

Abstract

Ed will discuss the issues that arise when generative AI companies scrape training data without consent, and the alternative - licensing training data - that is being embraced by many AI music companies.

Biography

Ed Newton-Rex is the founder of Fairly Trained, a non-profit that certifies generative AI companies for fair training data practices. He is also a Visiting Scholar at Stanford University.

In 2010, Ed founded Jukedeck, one of the first AI music generation startups. Jukedeck let video creators generate music for their videos, and was used to create more than a million pieces of music. It was acquired by ByteDance in 2019. At ByteDance, Ed led the AI Music lab, then led Product for TikTok in Europe.

In 2022 Ed joined Stability AI, the company behind Stable Diffusion, to lead their Audio team. His team launched Stable Audio, Stability's music generation product, which was named one of TIME Magazine's best inventions of the year in 2023. He resigned from Stability in November 2023 due to the company's policy of training AI models on copyrighted work without consent, and in 2024 founded Fairly Trained. He is a published composer of choral music.

DEI Keynote

Listening For Diversity: The Ways in Which Critical Attention to Words Helps Move Us Closer Towards Realizing Our Full Humanity

Valerie Joseph

Smith College

Abstract

Using her experience as a dancer, therapist, mediator, diversity trainer, anthropologist, college educator, and originator of Grounded Knowledge Panels®, Valerie Joseph distills lessons learned about the power of intentional and principled listening. She offers ideas on how to harness the energy derived from listening differently to fuel the capacity to have uncomfortable, rich, dynamic and productive thinking. This forms the basis upon which we are challenged to make transformative choices about how we operate with those other humans with whom we share the planet.

Biography

Valerie Joseph earned a Ph.D. in Cultural Anthropology from the University of Massachusetts at Amherst. Her doctoral research investigated the enduring legacies of British colonialism and African heritage memory among the members of the African Diaspora in Carriacou, Grenada. Specifically, she mapped how the game songs and dance play of Carriacouan Black girls as well as their words, beliefs, and attitudes reflected both the detrimental internalization of colonial ideology and the restorative nature of African retentions.

Prior to her fieldwork in Carriacou, Dr. Joseph lived and worked in Botswana for seven years starting as a Peace Corps Volunteer science teacher in a junior secondary school, then as a training coordinator at the Cheshire Foundation's Mogoditshane Rehabilitation Center. She closed out her years in the country by working as co-director of the School for International Training's college semester abroad program. During her time in Botswana, Dr. Joseph sharpened her interest in cross-cultural conflicts, including those that seemed to be intractable, though traceable, in part, to cultural mores as well as historical and social patterns embedded in racial or ethnic bias and discrimination.

Dr. Joseph has a Masters in Movement Therapy with a concentration in counseling psychology and a Masters in Social Justice Education. Her supplemental training, work and experience in several fields includes gymnastics coaching, dance performance, diversity training, Authentic Movement (a contemplative dance form), mediation, teaching and management in higher education.

Dr. Joseph is an educator-interventionist working at Smith College as the Mentoring Administrative Director for AEMES (Achieving Excellence in Math Engineering and Science). In that role, she manages programs to support the most marginalized students who are pursuing STEM. She also teaches college success seminars within the AEMES Scholars Program.

Dr. Joseph is co-founder of the Smith Roundtable Group. Started in 2020, the SRG is a small contingent of staff, faculty, and students dedicated to creating opportunities for information sharing and conversation about important current events. Past Roundtable offerings included: "Daring to be Hopeful: A Critical Response to the White Supremacist Storming of Our Capitol" and "Why is the Power of Young People so Threatening to the Status Quo?" The most recent Roundtable event took place in September of this year: "'Calling In' for Democracy and Human Rights: A Consideration of Project 2025."

In and outside of Smith, Dr. Joseph convenes a unique form of public discourse that she originated. Grounded Knowledge Panels® are public conversations by small groups of people who have realistic, authentic and personal experience and understanding of a particular topic or question. Emerging from core Black culture, Grounded Knowledge Panels are a synthesis of Dr. Joseph's study and work in various fields including anthropology, Authentic Movement, education and mediation. As panelists converse among themselves, audience members are invited as "witnesses" to observe the

discussion. Both groups - panelists and witnesses – bring a distinctive power, depth and responsibility to the experience of speaking and listening.

Dr. Joseph is a five time recipient of the Smith College Spotlight Award, an honor presented to staff members, chosen by peers, in appreciation of exceptional service. She is a 2020 recipient of the Elizabeth B. Wyant Gavel Award awarded by students to staff members who have performed outstanding work in the Smith community.

Dr. Joseph's first children's book, *This is What Maisie Believes*, is published by 619 Wreath Publishing.

Keynote Talk – 2

Navigating the Intersection of AI and Music: Innovation, Ethics, and the Future of the Industry

Elizabeth Moody

Granderson Des Rochers, LLP's New Media Group

Abstract

This speech explores the complex relationship between artificial intelligence and the music industry, tracing the evolution from early digital disruptions like Napster to today's AI-driven landscape. It examines how streaming platforms revolutionized music consumption and distribution, while also introducing new challenges such as streaming fraud. The speech delves into AI's multifaceted impact on music creation, production, and personalization, highlighting both its transformative potential and ethical concerns. The presentation also addresses controversial uses of voice AI technology and the legal and ethical considerations surrounding AI training data, including a fair use arguments and budding internal laws. Finally, we address proposed solutions, including the use of transparent attribution systems modeled after YouTube's Content ID and policies for opt-in/out rights management. This keynote calls for a balanced approach, urging collaboration between artists, technologists, and policymakers to ensure that AI's integration into music creation and distribution respects artistic integrity and promotes innovation.

Biography

Elizabeth Moody, partner and chair of Granderson Des Rochers, LLP's New Media Group, is a pioneer in the digital media world. Moody has been spearheading digital music and video initiatives since the post-Napster era, both as outside counsel, and as a business executive in-house at companies like YouTube and Pandora. Today, Moody remains positioned at the intersection of technology and music rights and continues to advise her technology and rightsholder clients toward new and innovative business models and licensing deals.

Moody is at the forefront of the developing issues and opportunities that AI presents to the music and entertainment industries. She counsels several prominent generative voice and audio AI companies, advises the non-profit Fairly Trained, which certifies AI companies who are training the data sets with fairly acquired, licensed or owned data, and Audioshake, an AI-based stem separation tool in use by record labels, movie studios, and entertainment companies today to ease production and marketing.

She is also keyed into the gaming and the web 3.0 world. She is partnerships counsel for the gaming company Roblox and also works closely with Wave XR, a virtual reality concerts start-up that works with artists to create unique live performances as avatar versions of themselves in imaginative digital landscapes. She developed and continues to grow Styng'r's efforts to power music in video games and online gaming experiences.

Along with gaming and the metaverse, she is passionate about the opportunities web 3.0 will bring to the music community and creators. She represents Audius, the blockchain-based music streaming service, in its efforts to help creators and their fans connect more authentically by embracing the opportunities offered through a decentralized network and Revelator, an all-in-one music platform providing digital distribution, analytics, and web 3.0 services to artists, record labels and publishers. She advises Copyright Delta, providing data connections to rights holders and AI tech platforms.

Moody is excited to bring opportunities to the music industry by forging deals with those in industries outside of music, including at the intersection of music and fitness. She represents connected fitness, yoga, pilates, mindfulness, cycling, and dance services to help them integrate music into their services. She has worked closely with Hydrow, the successful Peloton-style live reality-connected rowing experience, since its launch in 2019. She believes that VR plays an important role in fitness and works with Litesport and FitXR to ensure they have access to top-notch music experiences. She has also been working in the medical and wellness space exploring licensing structures to use music in the treatment of pain, dementia, and mental illness concerns through her work with MediMusic and her advisory participation on the board of Music Health.

Keynote Talk – 3

Status Report: AI Music in Q1 of the 21st Century

Douglas Eck

Google DeepMind

Abstract

I finished my PhD in 2000; a lot has happened over the ensuing ~25 years in the field of music and computation. It seems like an appropriate moment to look back at where we were, how far we've come, and where we're going next. I will discuss early experiments in RNN-generated music, the open-source Magenta project, the rise of LLM and diffusion models for music generation, and more recent work we've done at Google DeepMind in text, image, video and music generation. I'll also address the question of how AI might help us better understand music and maybe even give rise to new forms of musical expression.

Biography

Doug is a Senior Research Director at Google, and leads research efforts at Google DeepMind in Generative Media, including image, video, 3D, music and audio generation. His own research lies at the intersection of machine learning and human-computer interaction (HCI). In 2015, Doug created Magenta, an ongoing research project exploring the role of AI in art and music creation. Before joining Google in 2010, Doug did research in music perception, aspects of music performance, machine learning for large audio datasets and music recommendation. He completed his PhD in Computer Science and Cognitive Science at Indiana University in 2000 and went on to a postdoctoral fellowship with Juergen Schmidhuber at IDSIA in Lugano Switzerland. From 2003-2010, Doug was faculty in Computer Science in the University of Montreal machine learning group (now MILA machine learning lab), where he became Associate Professor. For more information see <http://g.co/research/douglaseck>.

Tutorials

Tutorial 1

Connecting Music Audio and Natural Language

Seung Heon Doh, Ilaria Manco, Zachary Novack, Jong Wook Kim and Ke Chen

Abstract

Language serves as an efficient interface for communication between humans as well as between humans and machines. Through the integration of recent advancements in deep learning-based language models, the understanding, search, and creation of music is becoming capable of catering to user preferences with better diversity and control. This tutorial will start with an introduction to how machines understand natural language, alongside recent advancements in language models, and their application across various domains. We will then shift our focus to MIR tasks that incorporate these cutting-edge language models. The core of our discussion will be segmented into three pivotal themes: music understanding through audio annotation and beyond, text-to-music retrieval for music search, and text-to-music generation to craft novel sounds. In parallel, we aim to establish a solid foundation for the emergent field of music-language research, and encourage participation from new researchers by offering comprehensive access to 1) relevant datasets, 2) evaluation methods, and 3) coding best practices.

Biographies of the Presenters

SeungHeon Doh is a Ph.D. student at the Music and Audio Computing Lab, KAIST, under the guidance of Juhan Nam. His research focuses on conversational music annotation, retrieval, and generation. SeungHeon has published papers related to music & language models at ISMIR, ICASSP and IEEE TASLP. He aims to enable machines to comprehend diverse modalities during conversations, thus facilitating the understanding and discovery of music through dialogue. SeungHeon has interned at Adobe Research, Chartmetric, NaverCorp, and ByteDance, applying his expertise in real-world scenarios.

Ilaria Manco is a Ph.D. student at the Centre for Doctoral Training in Artificial Intelligence and Music (Queen Mary University of London), under the supervision of Emmanouil Benetos, George Fazekas, and Elio Quinton (UMG). Her research focuses on multimodal deep learning for music information retrieval, with an emphasis on audio-and-language. Her contributions to the field have been published at ISMIR and ICASSP and include the first captioning model for music, and representation learning approaches to connect music and language for a variety of music understanding tasks. Previously, she was a research intern at Google DeepMind, Adobe and Sony, and obtained an MSci in physics from Imperial College London.

Zachary Novack is a Ph.D. Student at the University of California – San Diego, where he is advised by Julian McAuley and Taylor Berg-Kirkpatrick. His research is primarily aimed at controllable music and audio generation. Zachary seeks to build generative music models that allow for arbitrary musically-salient control mechanisms and enable stable multi-round generative audio editing, publishing such work at ICML, ICLR, and NeurIPS. Zachary has interned at Adobe Research, contributing such works as DITTO to be deployed in end-user applications. Outside of academics, Zachary is passionate about music education and teaches percussion in the southern California area.

Jongwook Kim is a Member of Technical Staff at OpenAI where he has worked on multimodal deep learning models such as Jukebox, CLIP, Whisper, and GPT-4. He has published at ICML, CVPR, ICASSP, IEEE SPM, and ISMIR, and he co-presented a tutorial on self-supervised learning at the NeurIPS 2021 conference. He completed a Ph.D. in Music Technology at New York University with a thesis focusing on automatic music transcription, and he has an M.S. in Computer Science and Engineering from the University of Michigan, Ann Arbor. He interned at Pandora and Spotify during the Ph.D. study, and he worked as a software engineer at NCSOFT and Kakao.

Ke Chen is a Ph.D. Candidate in the department of computer science and engineering at University of California San Diego. His research interests span across the music and audio representation learning, with a particular focus on its downstream applications of music generative AI, audio source separation, multi-modal learning, and music information retrieval. He has interned at Apple, Mitsubishi, Tencent, Bytedance, and Adobe, to further explore his research directions.

During his PhD study, Ke Chen has published more than 20 papers in top-tier conferences in the fields of artificial intelligence, signal processing, and music, such as AAAI, ICASSP, and ISMIR. Outside of academics, he indulges in various music-related activities, including piano performance, singing, and music composition.

Tutorial 2

Exploring 25 Years of Music Information Retrieval: Perspectives and Insights

Masataka Goto, Jin Ha Lee, and Meinard Muller

Abstract

This tutorial reflects on the journey of Music Information Retrieval (MIR) over the last 25 years, offering insights from three distinct perspectives: research, community, and education. Drawing from the presenters' personal experiences and reflections, it provides a holistic view of MIR's evolution, covering historical milestones, community dynamics, and pedagogical insights. Through this approach, the tutorial aims to give attendees a nuanced understanding of MIR's past, present, and future directions, fostering a deeper appreciation for the field and its interdisciplinary and educational aspects.

The tutorial is structured into three parts, each based on one of the aforementioned perspectives. The first part delves into the research journey of MIR. It covers the inception of query-by-humming and the emergence of MP3s, discusses the establishment of standard tasks such as beat tracking and genre classification, and highlights significant advancements, applications, and future challenges in the field. The second part explores the community aspect of ISMIR. It traces the growth of the society from a small symposium to a well-recognized international community, emphasizing core values such as interdisciplinary collaboration and diversity, and invites the audience to imagine the future of the ISMIR community together. Lastly, the third part discusses the role of music as an educational domain. It examines the broad implications of MIR research, the value of pursuing a PhD in MIR, and the significant educational resources available.

Each part invites audience interaction, aiming to provide attendees with a deeper appreciation of MIR's past achievements and insights into its potential future directions. This tutorial is not just a historical overview but also a platform for fostering a deeper understanding of the interplay between technology and music.

Biographies of the Presenters

Masataka Goto received the Doctor of Engineering degree from Waseda University, Tokyo, Japan, in 1998. He is currently a Principal Researcher at the National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, Japan. In 1992 he was one of the first to start working on automatic music understanding and has since been at the forefront of research in music technologies and music interfaces based on those technologies. Over the past 32 years he has published more than 300 papers in refereed journals and international conferences and has received 68 awards, including several best paper awards, best presentation awards, the Tenth Japan Academy Medal, and Tenth JSPS PRIZE. He has served as a committee member of over 120 scientific societies and conferences, including the General Chair of ISMIR 2009 and 2014, the Program Chair of ISMIR 2022, and the Member-at-large of the ISMIR Board from 2009 to 2011. As the research director, he began the OngaACCEL project in 2016 and the RecMus project in 2021, which are five-year JST-funded research projects (ACCEL and CREST) related to music technologies. He gave tutorials at major conferences, including ISMIR 2015, ACM Multimedia 2013, ICML 2013, ICPR 2012, and ICMR 2012.

Jin Ha Lee is a Professor and the Founder and Director of the GAMER (GAME Research) Group at the University of Washington Information School. She holds an M.S. (2002) and a Ph.D. (2008) in Library and Information Science from the University of Illinois at Urbana-Champaign. Her research focuses on exploring new ideas and approaches for organizing and providing access to popular music, multimedia, and interactive media, understanding user behavior related to the creation and consumption of these media, and using these media for informal learning in venues such as libraries and museums. She has been actively engaging with the ISMIR community from the early days of ISMIR, and was at the forefront of user-centered MIR research at ISMIR, contributing a number of papers on user perception of music similarity and mood, music listening and sharing behavior, cross-cultural aspects of MIR, and human-AI collaboration. She served as the Secretary of the ISMIR Board from the inception to 2015, and also as the General Co-Chair of ISMIR 2021, and the Scientific Program Co-Chair of ISMIR 2014, 2020, and 2024. She also serves as an Editorial Board Member for the Transactions of the International Society for Music Information Retrieval.

Meinard Müller received the Diploma degree (1997) in mathematics and the Ph.D. degree (2001) in computer science from the University of Bonn, Germany. Since 2012, he has held a professorship for Semantic Audio Signal Processing at the International Audio Laboratories Erlangen, a joint institute of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) and the Fraunhofer Institute for Integrated Circuits IIS. His recent research interests include music processing, music information retrieval, audio signal processing, and motion processing. He was a member of the IEEE Audio and Acoustic Signal Processing Technical Committee (2010-2015), a member of the Senior Editorial Board of the IEEE Signal Processing Magazine (2018-2022), and a member of the Board of Directors, International Society for Music Information Retrieval (2009-2021, being its president in 2020/2021). In 2020, he was elevated to IEEE Fellow for contributions to music signal processing. Currently, he also serves as Editor-in-Chief for the Transactions of the International Society for Music Information Retrieval (TISMIR). Besides his scientific research, Meinard Müller has been very active in teaching music and audio processing. He gave numerous tutorials at major conferences, including ICASSP (2009, 2011, 2019) and ISMIR (2007, 2010, 2011, 2014, 2017, 2019, 2023). Furthermore, he wrote a monograph titled “Information Retrieval for Music and Motion” (Springer 2007) as well as a textbook titled “Fundamentals of Music Processing” (Springer-Verlag 2015).

Tutorial 3

From White Noise to Symphony: Diffusion Models for Music and Sound

Chieh-Hsin Lai, Koichi Saito, Bac Nguyen Cong, Yuki Mitsufuji, and Stefano Ermon

Abstract

This tutorial will cover the theory and practice of diffusion models for music and sound. We will explain the methodology, explore its history, and demonstrate music and sound-specific applications such as real-time generation and various other downstream tasks. By bridging the gap from computer vision techniques and models, we aim to spark further research interest and democratize access to diffusion models for the music and sound domains.

The tutorial comprises four sections. The first provides an overview of deep generative models and delves into the fundamentals of diffusion models. The second section explores applications such as sound and music generation, as well as utilizing pre-trained models for music/sound editing and restoration. In the third section, a hands-on demonstration will focus on training diffusion models and applying pre-trained models for music/sound restoration. The final section outlines future research directions.

We anticipate that this tutorial, emphasizing both the foundational principles and practical implementation of diffusion models, will stimulate interest among the music and sound signal processing community. It aims to illuminate insights and applications concerning diffusion models, drawn from methodologies in computer vision.

Biographies of the Presenters

Chieh-Hsin Lai earned his Ph.D. in Mathematics from University of Minnesota in 2021. Currently, he is a research scientist at Sony AI and a visiting assistant professor at the Department of Applied Mathematics of National Yang Ming Chiao Tung University, Taiwan. His expertise is in deep generative models, especially diffusion models and its application for media content restoration. He has organized an EXPO workshop at NeurIPS 2023 on “Media Content Restoration and Editing with Deep Generative Models and Beyond”. Please refer here for his detailed information <https://chiehhsinjesselai.github.io/>.

Koichi Saito is an AI engineer at Sony AI. He has been working on deep generative models for music and sound, especially, solving inverse problems for music signals based on diffusion models and diffusion-based text-to-sound generation. He has extensive experience in showcasing advanced diffusion model technologies to businesses and industries related to music.

Bac Nguyen Cong earned his M.Sc. degree (summa cum laude) in computer science from Universidad Central de Las Villas in 2015, followed by a Ph.D. from Ghent University in 2019. He joined Sony in 2019, focusing his research on representation learning, vision-language models, and generative modeling. With four years of hands-on professional industry experience in deep learning and machine learning, his work spans various application domains, such as text-to-speech and voice conversion, showing his important contributions to the field.

Yuki Mitsufuji holds dual roles at Sony, leading two departments, and is a specially appointed associate professor at TokyoTech, where he lectures on generative models. He’s achieved Senior Member status in IEEE and serves on the IEEE AASP Technical Committee 2023-2026. He chaired “Diffusion-based Generative Models for Audio and Speech” at ICASSP 2023 and “Generative Semantic Communication: How Generative Models Enhance Semantic Communications” at ICASSP 2024. Please refer here for his detailed information <https://www.yukimitsufuji.com/>.

Stefano Ermon is an associate professor at Stanford, specializing in probabilistic data modeling with a focus on computational sustainability. He has received Best Paper Awards from ICLR, AACL, UAI, CP, and an NSF Career Award. He also organized a course on Diffusion Models at SIGGRAPH 2023. Please refer here for his detailed information <https://cs.stanford.edu/~ermon/>.

Tutorial 4

Humans at the Center of MIR: Human-subjects Research Best Practices

Claire Arthur, Nat Condit-Schultz, David R. W. Sears, John Ashley Burgoyne, and Joshua Albrecht

Abstract

In one form or another, most MIR research depends on the judgment of humans. Humans provide our ground-truth data, whether through explicit annotation or through observable behavior (e.g., listening histories); Humans also evaluate our results, whether in academic research reports or in the commercial marketplace. Will users like it? Will customers buy it? Does it sound good? These are all critical questions for MIR researchers which can only be answered by asking people. Unfortunately, measuring and interpreting the judgments and experiences of humans in a rigorous manner is difficult. Human responses can be fickle, changeable, and inconsistent—they are, by definition, subjective. There are many factors that influence human responses, some of which can be controlled or accounted for in experimental design, and others which must be tolerated but ameliorated through statistical analysis. Fortunately, researchers in the field of behavioral psychology have amassed extensive expertise and institutional knowledge related to the practice and pedagogy of human-subject research, but MIR researchers receive little exposure to research methods involving human subjects. This tutorial, led by MIR researchers with training (and publications) in psychological research, aims to share these insights with the ISMIR community. The tutorial will introduce key concepts, terminology, and concerns in carrying out human-subject research, all in the context of MIR. Through the discussion of real and hypothetical human research, we will explore the nuances of experiment and survey design, stimuli creation, sampling, psychometric modeling, and statistical analysis. We will review common pitfalls and confounds in human research, and present guidelines for best practices in the field. We will also cover fundamental ethical and legal requirements of human research. Any and all ISMIR members are welcome and encouraged to attend: it is never too early, or too late, in one's research career to learn (or practice) these essential skills.

Biographies of the Presenters

Claire Arthur is an assistant professor in the School of Music and co-director of the Computational and Cognitive Musicology Lab at the Georgia Institute of Technology, and adjunct faculty in the School of Psychology. She received her PhD in music theory and cognition from Ohio State University under David Huron. Her research largely focuses on modeling musical structure from a statistical perspective, as well as examining the cognitive and behavioral correlates of those structures, especially as it relates to musical expectations and emotional responses. Her MIR-related research interests lie in the intersection of music perception, computational musicology, and emotion prediction, with an emphasis on melody, voice-leading, and harmony.

Nat Condit-Schultz is a Lecturer and the Director of the Graduate Program for the Georgia Tech School of Music. Nat is a musician, composer, and scientist, specializing in the statistical modeling of musical structure. Nat directs the Georgia Tech rock and pop bands, and teaches courses in research methodology, music psychology, and music production. Nat's research interests include rhythm and tonality in popular music, the perceptual and structural roles of language and lyrics in music, and the music theory of hip-hop. Nat is a performer and composer, specializing in electric and classical guitar: as a composer, he specializes in imitative counterpoint and complex rhythmic/metric ideas like polyrhythm, “tempo spirals,” and irama, realized through classical guitar, rock instrumentation, and Indonesian Gamelan.

David R. W. Sears is Associate Professor of Interdisciplinary Arts and Co-Director of the Performing Arts Research Lab at Texas Tech University, where he teaches courses in arts psychology, arts informatics, and music theory. His current research examines the structural parallels between music and language using both behavioral and computational methods, with a particular emphasis on the many topics associated with pitch structure, including scale theory, tonality, harmony, cadence, and musical form. He also has ancillary interests in music on the global radio, music and emotion, and cross-cultural research. Recent publications appear in his Google Scholar profile.

John Ashley Burgoyne is Assistant Professor in Computational Musicology at the University of Amsterdam, teaching in the Musicology and Artificial Intelligence and conducting research in the Language and Music Cognition unit at the Institute for Logic, Language, and Computation. His current research focuses on using psychometric approaches in combination with representations and embeddings from deep learning models to improve the interpretability of AI models and flexibility in the design of musical stimuli and experiments. As director of the Amsterdam Music Lab, he is also interested in citizen science and online experimentation, and leads a team developing the MUSCLE infrastructure for facilitating online experiments requiring fine control of audio and music.

Joshua Albrecht is an Assistant Professor of Music Theory at the University of Iowa, and directs the Iowa Cognitive and Empirical Musicology lab. His current research blends statistical and computational musical analysis with behavioral studies to model listeners' perception of musical affect, melodic and harmonic complexity, and intonation. Working in a traditional School of Music, his research also focused on applying computational methods to traditional historical and analytical problems, using compositional output as proxies for investigating the cognition of historical compositional practices.

Tutorial 5

Deep Learning 101 for Audio-based MIR

Geoffroy Peeters, Gabriel Meseguer Brocal, Alain Riou, and Stefan Lattner

Abstract

Audio-based MIR (MIR based on the processing of audio signals) covers a broad range of tasks, including analysis (pitch, chord, beats, tagging), similarity/cover identification, and processing/generation of samples or music fragments. A wide range of techniques can be employed for solving each of these tasks, spanning from conventional signal processing and machine learning algorithms to the whole zoo of deep learning techniques.

This tutorial aims to review the various elements of this deep learning zoo commonly applied in Audio-based MIR tasks. We review typical audio front-ends (such as waveform, Log-Mel-Spectrogram, HCQT, SincNet, LEAF, quantization using VQ-VAE, RVQ), as well as projections (including 1D-Conv, 2D-Conv, Dilated-Conv, TCN, WaveNet, RNN, Transformer, Conformer, U-Net, VAE), and examine the various training paradigms (such as supervised, self-supervised, metric-learning, adversarial, encoder-decoder, diffusion). Rather than providing an exhaustive list of all of these elements, we illustrate their use within a subset of (commonly studied) Audio-based MIR tasks such as multi-pitch/chord-estimation, cover-detection, auto-tagging, source separation, music-translation or music generation. This subset of Audio-based MIR tasks is designed to encompass a wide range of deep learning elements. For each task we address a) the goal of the tasks, b) how it is evaluated, c) provide some popular datasets to train a system, and d) explain (using slides and pytorch code) how we can solve it using deep learning.

The objective is to provide a 101 lecture (introductory lecture) on deep learning techniques for Audio-based MIR. It does not aim at being exhaustive in terms of Audio-based MIR tasks nor on deep learning techniques but to provide an overview for newcomers to Audio-Based MIR on how to solve the most common tasks using deep learning. It will provide a portfolio of codes (Colab notebooks and Jupyter book) to help newcomers achieve the various Audio-based MIR Tasks.

Biographies of the Presenters

Geoffroy Peeters is a full professor in the Image-Data-Signal (IDS) department of Télécom Paris. Before that (from 2001 to 2018), he was Senior Researcher at IRCAM, leading research related to Music Information Retrieval. He received his Ph.D. in signal processing for speech processing in 2001 and his Habilitation (HDR) in Music Information Retrieval in 2013 from the University Paris VI. His research topics concern signal processing and machine learning (including deep learning) for audio processing, with a strong focus on music. He has participated in many national or European projects, published numerous articles and several patents in these areas, and co-authored the ISO MPEG-7 audio standard. He has been co-general-chair of the DAFx-2011 and ISMIR-2018 conferences, member and president of the ISMIR society, and is the current AASP review chair for ICASSP. At Telecom-Paris, he created the 40-hour program "Audio and Music Information Retrieval" for the Master-2 level "Data Science" which deals mostly with deep learning applied to MIR that inspired this tutorial.

Gabriel Meseguer Brocal is a research scientist at Deezer with over two years of experience at the company. Before joining Deezer, he completed postdoctoral research at Centre National de la Recherche Scientifique (CNRS) in France. In 2020, he earned his Ph.D. in Computer Science, Telecommunications, and Electronics with a focus on the Sciences & Technologies of Music and Sound at IRCAM. His research interests include signal processing and deep learning techniques for music processing, with a focus on areas such as source separation, dataset creation, multi-tagging, self-supervised learning, and multimodal analysis.

Alain Riou is a PhD student working on self-supervised learning of musical representations at Télécom-Paris and Sony CSL Paris, under the supervision of Stefan Lattner, Gaëtan Hadjeres and Geoffroy Peeters. Before that, he obtained a master degree in mathematics for machine learning at Ecole Normale Supérieure de Cachan (2020) and another one in signal processing and computer science applied to music at IRCAM (2021). His main research interests are related to

deep representation learning, with a strong focus on self-supervised methods for music information retrieval and controllable music generation. His work "PESTO: Pitch Estimation with Self-supervised Transposition-equivariant Objective" received the Best Paper Award at ISMIR 2023.

Stefan Lattner serves as a researcher leader at the music team at Sony CSL Paris, where he focuses on generative AI for music production, music information retrieval, and computational music perception. He earned his PhD in 2019 from Johannes Kepler University (JKU) in Linz, Austria, following his research at the Austrian Research Institute for Artificial Intelligence in Vienna and the Institute of Computational Perception Linz. His studies centered on the modeling of musical structure, encompassing transformation learning and computational relative pitch perception. His current interests include human-computer interaction in music creation, live staging, and information theory in music. He specializes in generative sequence models, computational short-term memories, (self-supervised) representation learning and musical audio generation. In 2019, Lattner received the best paper award at ISMIR for his work, "Learning Complex Basis Functions for Invariant Representations of Audio."

Tutorial 6

Lyrics and Singing Voice Processing in Music Information Retrieval: Analysis, Alignment, Transcription and Applications

Daniel Stoller, Emir Demirel, Kento Watanabe, and Brendan O'Connor

Abstract

Singing, a universal human practice, intertwines with lyrics to form a core part of profound musical experiences, conveying emotions, narratives, and real-world connections. This tutorial explores the commonly used techniques and practices in lyrics and singing voice processing, which are vital in numerous music information retrieval tasks and applications.

Despite the importance of song lyrics in MIR and the industry, high-quality paired audio & transcript annotations are often scarce. In the first part of this tutorial, we'll delve into automatic lyrics transcription and alignment techniques, which significantly reduce the annotation cost and enable more performant solutions. Our tutorial provides insights into the current state-of-the-art methods for transcription and alignment, highlighting their capabilities and limitations while fostering further research into these systems.

Moreover, we present "lyrics information processing", which encompasses lyrics generation and leveraging lyrics to discern musically relevant aspects such as emotions, themes, and song structure. Understanding the rich information embedded in lyrics opens avenues for enhancing audio-based tasks by incorporating lyrics as supplementary input.

Finally, we discuss singing voice conversion as one such task, which involves the conversion of acoustic features embedded in a vocal signal, often relating to timbre and pitch. We explore how lyric-based features can facilitate a model's inherent disentanglement between acoustic and linguistic content, which leads to more convincing conversions. This section closes with a brief discussion on the ethical concerns and responsibilities that should be considered in this area.

This tutorial caters especially to new researchers with an interest in lyrics and singing voice modeling, or those involved in improving lyrics alignment and transcription methodologies. It can also inspire researchers to leverage lyrics for improved performance on tasks like singing voice separation, music and singing voice generation, and cover song and emotion recognition.

Biographies of the Presenters

Daniel Stoller is a research scientist at MIQ, the music intelligence team at Spotify. He obtained his PhD from Queen Mary University in 2020, before researching causal machine learning at the German center for neurodegenerative diseases (DZNE). Experienced in audio source separation as well as generative modeling and representation learning, he develops machine learning models and techniques scalable to high-dimensional data such as raw audio signals, publishing in both machine learning and audio-related venues. With a special passion for music, he also worked extensively on lyrics alignment, and singing voice processing including separation, detection and classification.

Emir Demirel is a Senior Data Scientist at Music.ai / Moises, leading projects on lyrics and vocal processing. He obtained his Ph.D. at Queen Mary University of London, as a fellow to the "New Frontiers in Music Information Processing" project under EU's Marie Curie/Skladowska Actions. After completing his Ph.D, he joined Spotify's Music Intelligence team, enhancing his expertise before moving to Music.ai. His research interests span lyric transcription and alignment, speech recognition, and natural language processing, along with generative AI models.

Kento Watanabe is a senior researcher at the National Institute of Advanced Industrial Science and Technology (AIST), Japan. He received his Ph.D. from Tohoku University in 2018, and his work focuses on Lyrics Information Processing (LIP), natural language processing, and machine learning. He aims to bridge the gap between humans and computers in the field of music and language, and to improve interactions through advanced algorithms.

Brendan O'Connor has worked in music as a performer, composer, producer, teacher, and sound installation artist. He earned his Bachelor's in classical music at the MTU Cork School of Music (Ireland), followed by his Master's in music technology at the University of West London, specialising in the voice as the principal instrument in electroacoustic compositions. He then worked towards his Ph.D. in the field of singing voice conversion via neural networks at Queen Mary University of London. His research interests include the disentanglement of scarcely labelled vocal attributes, such as singing techniques. After completing his PhD, Brendan began working for a startup company in voice conversion, allowing him to continue working in his area of expertise with other researchers of the same field using SOTA machine learning techniques.

Special Session: Remembering Don Byrd

Host: Zhiyao Duan

Don Byrd, the General Chair of the very first ISMIR conference in 2000, has recently left us. This session remembered and celebrated the vision and contributions of this legend of the ISMIR community.

Creative Practice Sessions

Hosts: Cynthia Liem, Tomas Peire

On the occasion of ISMIR 2024's special focus on 'Bridging Technology and Musical Creativity', a dedicated Creative Practices track was initiated to especially focus on these bridging aspects. With this, we wanted to facilitate dialogues and collaborations on the bridge between technology and creativity, addressing the mutual needs and interests of artists, technologists, and industry professionals.

Leading up to ISMIR, a community call was published, allowing anyone with interest in music, technology and creativity to propose topics they wanted to collaborate on. In response, 15 proposals were submitted by a wide and international audience, including artists and a high school student.

At the main conference, the Creative Practices track organized several activities. First of all, two panel discussions were organized, featuring invited guests that have professionally been active on the bridge between technology and creative practice:

- **Session 1** largely featured artists who have been working with technology:
 - **Carlos Mosquera** - Musician, Programmer, CEO at CM MEDIA LLC
 - **Michelle Alexander** - Musician, Music Analyst & Mood Specialist at Pandora
 - **Mark Goldstein** - Percussionist, programmer, teacher, inventor; with an interest in the nexus of musical gesture, sound, and expression.
- **Session 2** featured guests (several having an artistic background) with strong connections to commercial and practical applications:
 - **Eyal Amir** - musician, software developer and musical instruments creator. Co-founder and CTO of the audio plugins company Modalics.
 - **Ben Cantil** aka Encanti - electronic music producer, software designer, educator, and scholar
 - **Seth Forsgren** - Amateur Musician, CEO at Riffusion
 - **Spencer Salazar** - Principal Engineer (formerly CTO) at Output

At the start of session 1, special attention was drawn to the community proposals, where proposers physically present at ISMIR also presented their projects/proposals:

- **Nicole Brady** - Music Performer, music producer, composer, music teacher. Project: Memory of Sun: Inspired by Anna Akhmatova's poignant poem "Memory of Sun," this project is a multisensory installation that explores themes of memory, transformation, and fading light through light, audio-visual, and interactive technologies
- **Jin Ha Lee, Michele Newman, Lidia Morris** - a team of researchers at the University of Washington, coming from a background of information science, with interest in MIR research. Interested in dialogues on Understanding the Creative Processes of Human-AI Music Collaboration, with particular interest in the role and viewpoint of the human creator.

The intention is to keep advertising for the community proposals until Spring 2025, where interested parties can directly engage with proposers. From resulting collaborations, we wish to draw practical lessons on how collaborations can be effective, and where vocabulary matches or does not yet match across disciplinary viewpoints.

Finally, a Creative Practice social was hosted by Riffusion at the Riffusion HQ and Phonobar, providing an informal networking platform, technical demos, and live performances. Apart from invited artists, there also were performances by the long-time ISMIR community members Dadabots and panelist Encanti.

The Creative Practice track was well received, with ISMIR audience members indicating they appreciated both the community call and the elements programmed at the conference. As for points to keep in mind for possible future continuations, it will be good to be aware early on that a track like this will especially attract an audience (both in terms of invitees to panels, as well as proposers to the community call) that would not normally attend ISMIR. As such, these attendees are not obviously having resources for conference attendance, nor will they naturally 'speak the language and know the culture' of ISMIR. It will thus be good to provide support for them, both in terms of travel/registration support, but also on a more translational note with regard to the work presented at the conference (this year's backchannel in the Slack community may have served that purpose). Furthermore, it will be good to more proactively rotate the community call early on and directly link possible collaborators, although this will need considerable chairing capacity and good knowledge of the ISMIR network.

Industry Sessions

Hosts: Brandi Frisbie, Minz Won

Sponsor presentation

Eleven ISMIR sponsors gave presentations on their vision, goals, research topics, demos, and potential job opportunities in two sessions.

Panel discussion

Five panelists and a moderator engaged in a discussion about the past, present, and future of music and technology. The session included insights from the panelists, followed by a Q&A and an opportunity for social networking. This event was open to the public.

Theme: Bridging Technology and Musical Creativity

Moderator: Jessica Powell (CEO and Co-founder of AudioShake)

Panelists:

- Stephen White (CPO of EMPIRE)
- Douglas McCausland (TAC Studio Manager and Faculty Lecturer at SFCM)
- Tony Brooke (Independent Consultant for Music Data Companies)
- Heidi Trefethen (Adjunct Professor at SFCM TAC, FOH/Monitor Engineer at SF Jazz)
- Cheng-Zhi Anna Huang (Assistant Professor at MIT)

Live Online Sessions

Mindfulness Sessions

Virtual Mindfulness Sessions were programmed during onsite lunch breaks (in live and replay time zones) to give online participants a chance to unwind between plenary sessions. The experiences presented in the ISMIR 2024 Mindfulness sessions were designed to help guide attendees into a deeper sense of Calm. All visuals were naturally produced and captured, with no AI intervention.

The ISMIR 2024 Organizers thank soundBrilliance (<https://www.soundbrilliance.com/>) for creating custom audiovisual experiences for these sessions.

soundBrilliance™

Online Social Events

Participants met over Zoom for informal conversation and contributed to the ISMIR 2024 Collaborative Playlist.

Online Special Sessions

Host: Vino Alluri

Five invited researchers from MIR and neighboring communities joined virtual sessions to give short presentations of their work and engage in informal conversation with attendees.

Kathleen Rose Agres: Affective Music Generation for Emotion Regulation in Listeners

There has been a surge of interest in automatic music generation in recent years, particularly in affective music generation. Numerous systems now offer controllable AI-based affective music generation (AI-AMG), as highlighted in Dash & Agres (2024). While these systems have been developed for various applications—including soundtrack creation in gaming and virtual reality, co-creativity, and health and well-being—this talk focuses on the use of AI-AMG to support emotion regulation in listeners. One such system, AffectMachine (Agres, Dash, & Chua, 2023), is designed to generate affective music in real time, and is capable of composing in both classical and pop-music styles. Recent findings across several studies demonstrate AffectMachine’s efficacy in producing music perceived as emotional and capable of inducing emotions, as shown by subjective emotion ratings and physiological responses. This talk will explore the implications of systems like AffectMachine for supporting emotion self-regulation.

Bio: Dr. Kat Agres is an Assistant Professor at the Yong Siew Toh Conservatory of Music at the National University of Singapore (NUS), and Founding Director of the Centre for Music and Health, the first dedicated research centre in Southeast Asia to spearhead evidence-based research leveraging the efficacy of music for health and well-being. Kat received her PhD in Cognitive Psychology from Cornell University and completed her postdoctoral fellowships in Music Cognition and Computational Creativity at the University of London. She also holds a bachelor’s degree in Cello Performance and Psychology from Carnegie Mellon University. Kat’s research explores music interventions and technologies for healthcare and well-being, music perception and cognition, and computational creativity. She has received numerous grants to support her research in Singapore, the US, and UK. Kat has presented her research in over twenty countries around the world, and has also performed professionally as a cellist.

Kaustuv Kanti Ganguli: Harmonic Convergence: Orchestrating the Synergy of Human Intuition and Machine Intelligence in Music

In the rapidly evolving landscape of computational musicology, we stand at a fascinating crossroads where human perception intertwines with machine-driven analysis. This convergence offers unprecedented opportunities to unravel the

complexities of musical structures, particularly in rich non-Eurogenetic traditions such as Indian art music. By harmonizing human cognition with artificial intelligence, we can decode the intricate artifacts of audio signal processing, revealing new dimensions in our understanding of music. This approach not only enhances our appreciation of musical nuances but also challenges us to rethink the boundaries between human creativity and computational analysis.

As we navigate this confluence, we must consider the profound implications for music education, composition, and appreciation. How can we leverage machine learning to augment human musical intuition? What new insights into musical cognition can emerge from this synthesis? By exploring these questions, we open doors to innovative pedagogical tools, more nuanced music recommendation systems, and perhaps even new forms of musical expression. The future of music analysis lies not in choosing between human expertise and artificial intelligence but in orchestrating a symphony where both play in perfect harmony, each enhancing the other's strengths and compensating for limitations.

Bio: Dr. Kaustuv Kanti Ganguli is an Associate Professor of Artificial Intelligence at Zayed University and a Scholar at New York University Abu Dhabi Scholar, spearheading computational musicology and machine learning research. His innovative work bridges AI and music, focusing on Arabian Gulf and South Indian repertoires. Dr. Ganguli develops AI models that enhance music understanding, preservation, and education by combining engineering approaches with human cognition. A President's Gold Medal recipient and accomplished Hindustani vocal performer, his expertise spans machine learning, virtual reality, and audio processing. His groundbreaking projects include Raga/Makam characterization, multi-sensory perception, and crossmodal correspondence that collectively foster a deeper appreciation for diverse musical traditions through the lens of artificial intelligence. Kaustuv envisions blending humanistic and computational methods in a cross-disciplinary environment within a liberal arts framework, focusing on cutting-edge research and sustainable, innovative teaching.

Martin Hartmann: Music and Movement: Exploring Social and Multimodal Dimensions of Rhythmic Entrainment

The talk addresses key challenges in the field of music and movement through two ongoing studies at the Centre of Excellence in Music, Mind, Body and Brain at the University of Jyväskylä. The first challenge explores rhythmic-social entrainment within the context of free dyadic dance. We present a study that examines the relationship between rhythmic-social entrainment and social as well as musical affiliation during adolescence, using markerless motion capture technology. Following a 2x2 factorial design, participants dance freely in dyads with a friend and with a stranger to music of their choice and to music selected by us. The second challenge focuses on the multimodality of rhythmic-social entrainment. We discuss a study that employs motion capture and surface electromyography to investigate the impact of visual cues and performed activities on acoustic features, physiological responses, and kinematic responses in choir singing. The goal is to understand how the visibility of other choir members and the performed activities (chat, homophony, polyphony, and musical improvisation) influence different types of individual and group responses. In addition to exploring the social aspects of rhythmic entrainment in dance and its multimodal nature in choir singing, we emphasize the extraction of musical features and individual and social acoustic and kinematic features. We also consider potential take-home messages from these studies for the music information retrieval community and beyond.

Bio: Martin Hartmann is an Assistant Professor of Musicology at the University of Jyväskylä, where he works for the Centre of Excellence in Music, Mind, Body, and Brain and for the European Research Council project MUSICCONNECT. His research encompasses music and movement, perception, information retrieval, and therapy. Currently, he specializes in the computational modeling of multimodal interactions in music and dance contexts. He is an executive group member of the Finnish Doctoral Network for Music Research and the local coordinator of the EU-funded FORTHEM Alliance Lab for Arts and Aesthetics in Contemporary Society. He led the project "Interaction in Music Therapy for Depression", maintains the MoCap (Motion Capture) Toolbox for MATLAB, and holds editorial roles for the journals Music Perception and Psychology of Music.

Amanda Krause: Can We Categorise Younger Adult Listeners?

The evolution of digital listening technologies continues to impact the way we think about music consumption and music listening practices. Krause and North's (2016) findings suggest that, in addition to demographic characteristics, psychological constructs should be considered when investigating listening practices and technology use. The present study uses latent profile analysis (LPA), which is a statistical technique that focuses on identifying latent subgroups within a population based on a set of variables. With this study, LPA affords us the opportunity to attempt to categorise types of music listeners. To explore this possibility, we draw on data collected from a sample of 584 younger adults residing in Australia

(Age = 19.62; 74.10% female). Participants were asked to complete an online questionnaire that included demographics, the musicianship module of the MUSEBAQ (Chin, et al., 2018), the Music Engagement Test (MET; Greenberg & Rentfrow, 2015), Langford's (2003) Big Five proxy personality scale, Krause and Hargreave's (2013) Music Self-Images Questionnaire, and Krause and Brown's (2021) format use measure. With analyses underway, preliminary indications suggest that format use and MET scores may differentiate listener typologies. Study findings further our theoretical understanding of how individuals consume music in everyday life.

Bio: Dr. Amanda E. Krause is a Senior Lecturer (Psychology) in the College of Healthcare Sciences at James Cook University (Queensland, Australia). As a music psychology scholar based at James Cook University, she studies how we experience music in our everyday lives.

Her passion for researching the social and applied psychology of music has led her to give guest lectures and public talks and serve as President of the Australian Music & Psychology Society (AMPS). She is the author of numerous peer-reviewed academic publications and has spoken on her research to academics and industry leaders at conferences around the world. Her research has made significant contributions to understanding how listening technologies influence people's experiences and how musical engagement impacts well-being. Dr Krause's current programs of research concern how everyday music and radio experiences influence people's well-being.

Sebastian Stober: Generative AI Training and Copyright Law

Training generative AI models requires extensive amounts of data. A common practice is to collect such data through web scraping. In the USA, AI developers rely on "fair use" and in Europe, the prevailing view is that the exception for "Text and Data Mining" (TDM) applies. In a recent interdisciplinary tandem-study with a legal expert, I have argued in detail that this is actually not the case because generative AI training fundamentally differs from TDM. In this talk, I will share our main findings and the implications for both public and corporate research on generative models. I will further discuss how the phenomenon of training data memorization leads to copyright issues independently from the "fair use" and TDM exceptions. Finally, I would like to outline how the ISMIR could contribute to the ongoing discussion about fair practices with respect to generative AI that satisfy all stakeholders.

Bio: Sebastian Stober is professor for Artificial Intelligence at the Otto-von-Guericke-University Magdeburg, Germany. He studied computer science with focus on intelligent systems in Magdeburg until 2005 and received his PhD with distinction on the topic of adaptive methods for user-centered organization of music collections in 2011. From 2013 to 2015, he was postdoctoral fellow at the Brain and Mind Institute in London, Ontario where he pioneered deep learning techniques for studying brain activity during music perception and imagination. Afterwards, he was head of the Machine Learning in Cognitive Science Lab at the University of Potsdam, before returning to Magdeburg in 2018. In his current research, he investigates and develops generative models for music and speech as well as methods to better understand what an artificial intelligence has learned and how it solves specific problems. To this end, he combines the fields of artificial intelligence and machine learning with cognitive neuroscience and music information retrieval.

TISMIR Presentations

This year marks the first time we have included presentations from the journal *Transactions of the International Society for Music Information Retrieval* (TISMIR) in the ISMIR conference program. This initiative aims to enhance the visibility of TISMIR and encourage more submissions and participation in the journal.

Any TISMIR paper published between June 1, 2023 and May 31, 2024 qualified for presentation, provided the authors registered during the regular registration period. We are pleased to announce that 10 TISMIR papers were registered and presented at this year's ISMIR conference (see list below).

These 10 papers were integrated into the Oral and Poster sessions, ensuring they received the same level of attention and relevance as regular ISMIR papers.

The ISMIR 2024 General Chairs extend our thanks to the TISMIR Editors-in-Chief and the ISMIR Board for their support and collaboration in making this possible.

The papers below are not included in the ISMIR 2024 proceedings, but can be accessed by visiting the TISMIR website (<https://transactions.ismir.net/>).

Papers presented (sorted alphabetically by first author):

1. Edwards, D., Dixon, S. and Benetos, E., 2023. PiJAMA: Piano Jazz with Automatic MIDI Annotations. TISMIR, 6(1), p.89–102.
2. Fabbro, G., Uhlich, S., Lai, C.-H., Choi, W., Martínez-Ramírez, M., Liao, W., Gadelha, I., Ramos, G., Hsu, E., Rodrigues, H., Stöter, F.-R., Défossez, A., Luo, Y., Yu, J., Chakraborty, D., Mohanty, S., Solovyev, R., Stempkovskiy, A., Habruseva, T., Goswami, N., Harada, T., Kim, M., Lee, J.H., Dong, Y., Zhang, X., Liu, J. and Mitsufuji, Y. (2024) 'The Sound Demixing Challenge 2023 – Music Demixing Track', TISMIR, 7(1), p. 63–84.
3. Maia, L.S., Rocamora, M., Biscainho, L.W.P. and Fuentes, M. (2024) 'Selective Annotation of Few Data for Beat Tracking of Latin American Music Using Rhythmic Features', TISMIR, 7(1), p. 99–112.
4. Özer, Y., Schwär, S., Arifi-Müller, V., Lawrence, J., Sen, E. and Müller, M. (2023) 'Piano Concerto Dataset (PCD): A Multitrack Dataset of Piano Concertos', TISMIR, 6(1), p. 75–88.
5. Peter, S.D., Cancino-Chacón, C.E., Foscarin, F., McLeod, A.P., Henkel, F., Karystinaios, E. and Widmer, G. (2023) 'Automatic Note-Level Score-to-Performance Alignments in the ASAP Dataset', TISMIR, 6(1), p. 27–42.
6. Plaja-Roglans, G., Nuttall, T., Pearson, L., Serra, X. and Miron, M. (2023) 'Repertoire-Specific Vocal Pitch Data Generation for Improved Melodic Analysis of Carnatic Music', TISMIR, 6(1), p. 13–26.
7. Schwär, S., Krause, M., Fast, M., Rosenzweig, S., Scherbaum, F. and Müller, M. (2024) 'A Dataset of Larynx Microphone Recordings for Singing Voice Reconstruction', TISMIR, 7(1), p. 30–43.
8. Uhlich, S., Fabbro, G., Hirano, M., Takahashi, S., Wichern, G., Le Roux, J., Chakraborty, D., Mohanty, S., Li, K., Luo, Y., Yu, J., Gu, R., Solovyev, R., Stempkovskiy, A., Habruseva, T., Sukhovei, M. and Mitsufuji, Y. (2024) 'The Sound Demixing Challenge 2023 – Cinematic Demixing Track', TISMIR, 7(1), p. 44–62.
9. Weiß, C., Arifi-Müller, V., Krause, M., Zalkow, F., Klauk, S., Kleinertz, R. and Müller, M. (2023) 'Wagner Ring Dataset: A Complex Opera Scenario for Music Processing and Computational Musicology', TISMIR, 6(1), p. 135–149.
10. Zhang, Y., Zhou, Z., Li, X., Yu, F. and Sun, M. (2023) 'CCOM-HuQin: An Annotated Multimodal Chinese Fiddle Performance Dataset', TISMIR, 6(1), p. 60–74.

Papers – Session I

FORMAL MODELING OF STRUCTURAL REPETITION USING TREE COMPRESSION

Zeng Ren¹
EPFL

Yannis Rammos
EPFL

Martin Rohrmeier
EPFL

zeng.ren@epfl.ch yannis.rammos@epfl.ch martin.rohrmeier@epfl.ch

ABSTRACT

Repetition is central to musical structure as it gives rise both to piece-wise and stylistic coherence. Identifying repetitions in music is computationally not trivial, especially when they are varied or deeply hidden within tree-like structures. Rather than focusing on repetitions of musical events, we propose to pursue repeated structural *relations* between events. More specifically, given a context-free grammar that describes a tonal structure, we aim to computationally identify such relational repetitions within the derivation tree of the grammar. To this end, we first introduce the *Template*, a grammar-generic structure for generating trees that contain structural repetitions. We then approach the discovery of structural repetitions as a search for optimally compressible *Templates* that describe a corpus of pieces in the form of production-rule-labeled trees. To make it tractable, we develop a heuristic, inspired by tree compression algorithms, to approximate the optimally compressible *Templates* of the corpus. After implementing the algorithm in Haskell¹, we apply it to a corpus of jazz harmony trees, where we assess its performance based on the compressibility of the resulting *Templates* and the music-theoretical relevance of the identified repetitions.

1. INTRODUCTION

Repetition has been widely recognized as an essential means of establishing effects of coherence in music [1–5]. In Western music, at least, it operates at multiple levels of structure—whether within individual compositions or across different pieces—and takes the shape of musical motifs, themes [6–9], and sectional forms [10–14], among others.

In formalizing repetition in, we need to clarify the *object*, the *means*, and the *scope* of repetition. The *object* specifies the kind of entities being repeated—for instance concrete pitch events or abstract relations between them. By *means* of repetition we refer to processes via which the


objects are repeated. The *scope* specifies the musical context in which any repetitions are identified.

This paper focuses on *structural repetitions* whose *objects* are not the musical events themselves but rather relations among them. By way of a musical example, consider the first eight bars in the jazz standard “Satin Doll.” In common form-theory [10] terms, is sentential; its first half (“presentation”) establishes and repeats a “basic idea” one step higher, whereas the second half (“continuation”) accelerates the harmonic rhythm towards a closing gesture.² A parse tree of the piece (see Figure 1a) based on the jazz harmony grammar by Rohrmeier [15] clarifies the chord dependency and constituency. Here we may observe that the varied repetition of the basic idea in the presentation is reflected in the parallelism between the respective tree components.³

One might think that the tree topology suffices to capture the musical intuition of the parallel structures within this phrase. Further observation reveals that the equality in topology is at most a necessary but not sufficient condition for parallelism. Equality between tree topologies can only express sameness of grouping structure (“constituency”), irrespective of group contents. This is not enough to describe common pattern-like tonal structures such as sequences. The essence of the phenomenon in this example lies also in the equality of the *relations* (edge labels). Listeners familiar with the genre would identify the repeating *objects* as “ii-V”-type motions. To make the relations explicit, we construct a production-rule-labeled tree as shown in Figure 1b. this representation, the parallelism is reflected in the equality of labeled tree components. This notion of repetition, now construed as equality of abstract relations, is crucial in formally capturing parallelism in music.

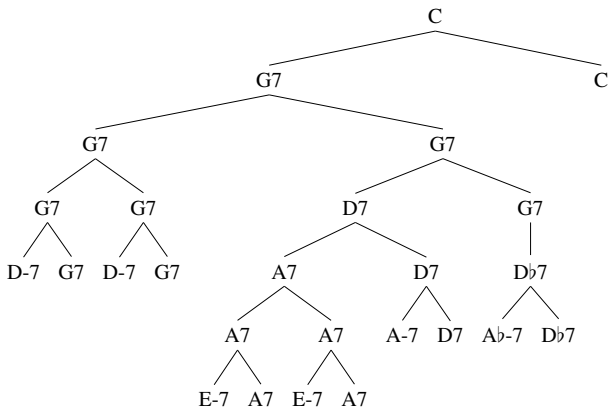
The *means* of repetition can be informally understood as the coloring present in the rule-labeled tree, which demarcates different rule types. This coloring imposes on the derivation of the piece a repetition constraint that is recursively constrained: indeed, one challenge is to express repetition not just between tree leaves or sub-trees, but also between connected subgraphs of the tree. This generalization would enable us to express deep-level repetitions of tonal frameworks despite non-parallel operations close to

¹ <https://github.com/ren-zeng/formal-modeling-of-structural-repetition.git>

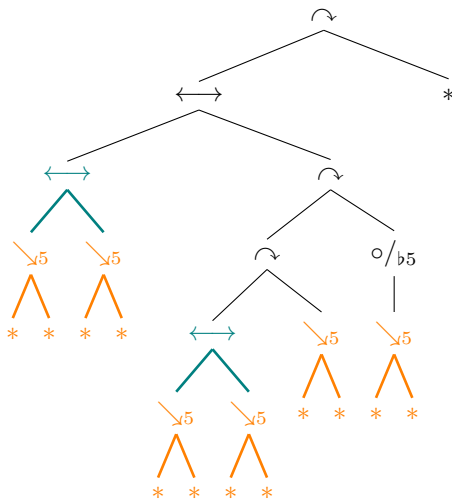
 © Z. Ren, Y. Rammos, and M. Rohrmeier. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Z. Ren, Y. Rammos, and M. Rohrmeier, “Formal Modeling of Structural Repetition using Tree Compression”, in *Proc. of the 25th Int. Society for Music Information Retrieval Conf.*, San Francisco, United States, 2024.

² This hearing becomes obvious after considering the parallelism within the melody (not shown in the figure).

³ “Tree component” here refers not just to subtrees but also to connected subgraphs within the tree.



(a) Harmony tree of Satin Doll mm. 1-8.



(b) The rule-labeled tree corresponding to the harmonic analysis. “ \sim ” indicates applied dominant relation. “ \longleftrightarrow ” indicates prolongation. “ \searrow_5 ” indicates diatonic descending fifth relation. “ o/b_5 ” indicates tritone substitution.

Figure 1: Hierarchical harmonic analysis of “Satin Doll” mm. 1-8 using the jazz harmony grammar [15] under two representations.

the surface (e.g. ornaments). A constructive definition of repetition *means* is provided in section 3.

Finally, we distinguish two kinds of *scopes* for repetition: piece-wise and corpus-wise. This distinction is important for characterizing musical styles. For instance, the “ii-V-I” chord progression in jazz is a recurrent harmonic device **across the corpus**. In contrast, the two descending thirds arranged one descending step apart, which open Beethoven’s Fifth Symphony, establish themselves as a motive through specific processes of repetition **within the piece** (and not outside it).

2. RELATED WORK

Leaving aside historical texts, a plethora of contemporary theoretical studies have examined the phenomenon of repetition in music [16], as well as the distinction between—borrowing Eugene Narmour’s terms [17]—“style structures” [18, 19] and piece-specific “idiostructures” [7, 20].

To computationally model the repetition phenomena in

the “Satin Doll” example of section 1, we coordinate two kinds of hierarchical structures: one that explains musical entities in terms of a context-free grammar (CFG), and another that explains the repetition of grammar rule applications. Hierarchical models of tonal structure are not new [21–25]. The same can be said about the hierarchical understanding of repetition itself such as the String Pattern-Induction Algorithm (SPIA) [26]. Methods for repetition identification have primarily focused on the repeated material itself by searching for exact or inexact successions [27–32] (also see [33] for an overview of this body of research). However, to the best of our knowledge, and outside the sphere of purely music-theoretical contributions, little attention has been paid to computational models of repetition whose objects are relations (generative procedures). Variation, for example, is often understood as a departure from exact repetition by means of ad-hoc or systematic transformations on a concrete entity. In this paper, by contrast, we understand variation as a different elaboration of the same *abstract* entity. Building upon the work by Finkensiep et al. on repetition structure inference [34], we extend its notion of “formal prototypes” to accommodate non-exact repetitions.

In the field of computer science, grammar-based compression algorithms aim to compress data by factoring out repeated information and storing it only once. Grammar-based compression algorithms have been developed and studied both for strings [35] and trees [36–40]. With string-like input data, the Sequitur algorithm [41] encodes a compressed string by constructing a straight-line grammar—a subclass of CFGs—whose language size is equal to one. When the input data have tree or forest form, algorithms such as [35–37, 39, 40] construct a straight-line tree program by iteratively constructing repeated units using “digrams,” which assemble a unit of repetition from two adjacent units. In the case of tree patterns, a digram consists of a root plus one of its children. Among the related research, the *TreeRepair* algorithm by Markus Lohrey et al. [37] is relevant as it is specifically concerned with the notion of “deep-level repetition”, i.e. of connected graphs within a tree.

In this paper, we formalize repetition patterns as functions operating on generic abstract syntax trees. Using a new approach that is based on tree compression algorithms and formal prototypes, we introduce a model that can discover piece-specific and stylistic patterns from a forest of abstract syntax trees.

3. FORMALISM

Meta-rule. A *meta-rule* is a list of symbols in $\mathbb{N}^+ \cup \{_, \star\}$ where “_” denotes a new symbol, “ \star ” denotes the recurrence of the parent symbol (“parent repeat”), and “ n ” denotes the recurrence of the n -th argument (“sibling repeat”). In the rest of the paper, we use \mathcal{M} to denote the space of *meta-rules*. Each $m \in \mathcal{M}$ induces a repetition function $rep_m : (X, X^n) \rightarrow X^{m+k}$ where the first argument represents the root of a sub-tree and the second represents its children. Here are two examples of the workings

of the rep_m function:

$$rep_{\langle _, _, 1, \star \rangle}(5, [3, 1]) = [3, 1, 3, 5]$$

$$rep_{\langle \star, _, 2, _, _, 4 \rangle}(t, [a, b, c]) = [t, a, a, b, c, b].$$

Intuitively, a *meta-rule* encodes the atomic “means” of repetition within a tree structure without specifying the *object* of repetition.⁴ In addition, for $m \in \mathcal{M}$, we define $sizeIn(m)$ to be the number of “ $_$ ” symbols and the $sizeOut(m)$ to be the length of the *meta-rule*.

Template. Given a context-free Grammar G where P_G is the set of production rules, we define its corresponding *Template* \mathcal{T}_G constructively using following axioms:

1. (Rule Lifting)

$$\forall f \in P_G. f \in \mathcal{T}_G$$

2. (Composition)⁵

$$\forall f, g \in \mathcal{T}_G, i \leq \text{arity}(g). (i, g, f) \in \mathcal{T}_G$$

3. (Replication)

$$\forall g \in \mathcal{T}_G, m \in \mathcal{M}, \vec{f} \in \mathcal{T}_G^{sizeIn(m)}. (g, m, \vec{f}) \in \mathcal{T}_G$$

\mathcal{T}_G is effectively a context-free grammar that generates P_G -labeled trees. The template is a structured representation of a rule-labeled tree; each template can be mapped to a unique rule-labeled tree but each rule-labeled tree can be assigned to multiple templates. Each rule-labeled tree can be embedded as a template in a trivial way using the axioms *Rule Lifting* and *Replication* with the *meta-rules* containing only “ $_$ ”.

We view the problem of discovering *objects* and *means* of repetition as one of inferring optimally compressible *templates* that generate the given production-rule-labeled trees. Given a collection of rule-labeled trees, we want to parse them as *templates*, so that the total size of the list of *templates* is minimized under memoization.⁶

3.1 Atomic parsing operations for \mathcal{T}_G

P-rewrite. Given a pattern of form (i, f, x_i) , referred to as a “digram” in [37], the rewrite procedure operates as depicted in Figure 2 in a post-order fashion, replacing all non-overlapping instances of the pattern within the tree. Through iterative application, *P-rewrite* facilitates the abstraction of a connected subgraph of a tree in the form of a single node.⁷

⁴ Visualizing meta-rules as strings of literal symbols, for instance “ABAT”, would be more intuitive and readable. Here we opt for a referential representation as it facilitates the computational implementation.

⁵ Here the notion of composition extends single-argument function composition to multiple arguments. $g \circ_i f$ denotes the function obtained by passing the output of f to the i -th argument of g . For *composition* and *replication* we also require the functions involved to have compatible types. The *arity* of a template, informally speaking, is the number of arguments needed to form a complete tree. For a template lifted from a production rule, its *arity* is the number of child symbols. For templates constructed using *composition*, $\text{arity}((i, f, g)) = \text{arity}(f) + \text{arity}(g) - 1$. For templates constructed using *replication*, $\text{arity}((g, m, \vec{f})) = \sum_{x \in rep_m(g, \vec{f})} \text{arity}(x)$.

⁶ This connection to memoization is inspired by [42].

⁷ *P-rewrite* mirrors the “replacement step” described in [37]. To the best of our knowledge, this is the first application of this notion in the analysis of musical structure.

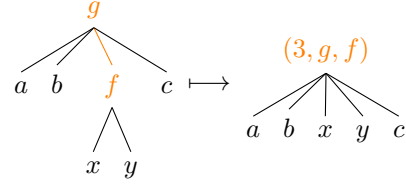


Figure 2: A single step of *P-rewrite* using the template $(3, g, f)$ arise from the *composition* axiom.

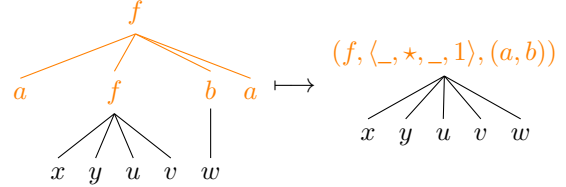


Figure 3: A single step of *R-rewrite* using the meta-rule $m = \langle _, \star, _, 1 \rangle$.

R-rewrite. *R-rewrite* is responsible for abstracting *means* of repetition. Given a $m \in \mathcal{M}$, *R-rewrite* applies to the tree whose first-level children are of the form $rep_m(g, \vec{f})$ for some g and \vec{f} . For example if $m = \langle _, \star, _, 1 \rangle$, then the rewrite procedure is the operation shown in Figure (3). *R-rewrite* corresponds to the *replication* case of \mathcal{T}_G .

Figure 4 demonstrates the relationship between a rule-labeled tree and the template that generates it. Note that at the most abstract level (Figure 4d) the template shows that the entire dominant region of the *Satin Doll* theme follows a “AABB” repetition pattern (indicated by the *meta-rule* $m_3 = \langle _, 1, _, 2 \rangle$) where “A” and “B” are templates T_2 and T_1 respectively. It is worth noting that the *objects* of this particular repetition pattern do not appear at the same structural level in the original rule-labeled tree. In general, the *Template* formalism allows us to coordinate tonal structure and repetition structure in a single hierarchical framework. It is the *Composition* axiom that makes this possible, since it can abstractly represent a connected graph of a tree as a single node. This mechanism is related to tree adjunction and substitution in Tree Adjoining Grammar (TAG) [43, 44].⁸

4. ALGORITHMS

The algorithm draws insights from [37] and [34], in particular their methods for tree and repetition pattern extraction. Departing from [37], our proposed algorithm introduces an additional replacement (*R-rewrite*) step to summarize repetition configurations. In comparison to [34], which operates with strings rather than trees, we introduce a mechanism that handles structural variations, and also derive *meta-rules* from data instead of prescribing a collection thereof. Furthermore, our algorithm differentiates

⁸ A *template* of the form (i, g, f) simulates tree adjunction when g is a non-trivial template, and tree substitution when f is a non-trivial template.

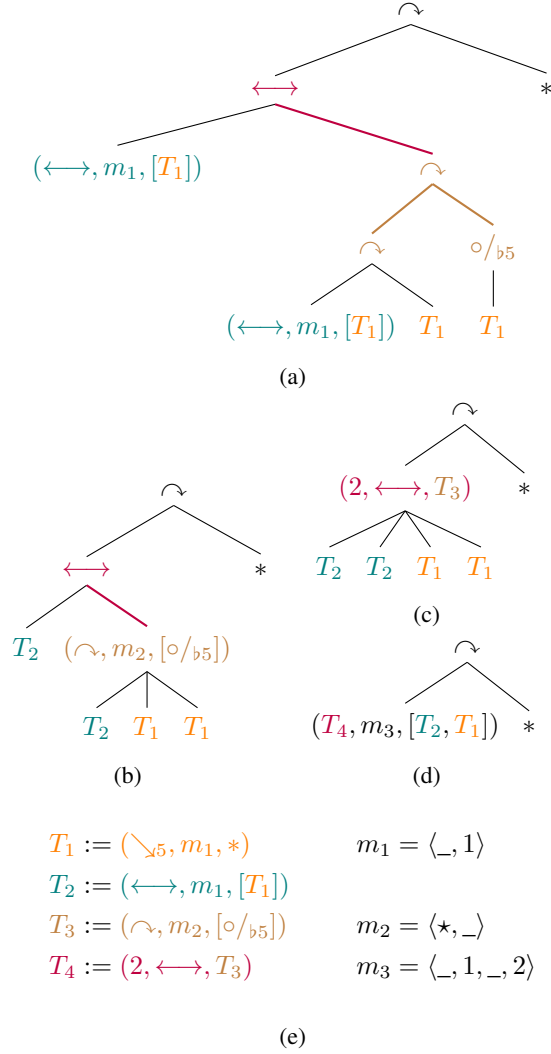


Figure 4: One possible parse of the rule-labeled tree in Figure 1b. Figures (a-d) demonstrate parsing steps (in bottom-up direction) using P and R rewrites, while figure (e) shows the definition of binded variables, which are colored accordingly.

repetition patterns by their scopes.

The algorithm works by incrementally constructing a table of mined patterns, whether *templates* or *meta-rules*. Its shape is identical to that of Table 1. Given a forest of rule-labeled trees, a pattern is *global* if it occurs at least in two trees, and *local* if it occurs at least twice in a single tree (but not in any other tree).

4.1 The Compression Algorithms

The goal is to find the minimal encoding achievable through a series of *P-Rewrite* and *R-Rewrite* operations. An exact solution would require us to try out all the possible rewrite steps (including all possible choices of patterns or meta-rules to write on), with each rewrite generating a new state dependent on the previous state of the program. Even with dynamic programming techniques, such an approach would be computationally intractable. To tame the computational complexity of the optimization problem, we

define two ‘greedy’ heuristics that help find rewrite candidates: a Local Compression for single trees (Algorithm 1) and a Global Compression for forests (Algorithm 2). The complete compression algorithm consists in an iteration of Algorithms 1 and 2 until a fixed-point is reached (when the result convergence).

Algorithm 1 The Local Compression Algorithm (a Single Step)

1: **Input**

t : Rule-labeled tree

dP : Dictionary from symbols to *templates*

dM : Dictionary from symbols to *meta-rules*

2: **function** $compress_L(t, dP, dM)$

3: $(o_P, o_r) \leftarrow occurrence_L(t)$

4: $c \leftarrow bestCandidate(o_P, o_r)$

5: **if** $c = Nothing$ **then**

6: **return** (t, tP, dM)

7: **else if** $c \in o_P$ **then**

8: $t' \leftarrow p\text{-Rewrite}(c, t)$

9: $dP' \leftarrow update(dP, c)$

10: $dM' \leftarrow dM$

11: **else if** $c \in o_r$ **then**

12: $t' \leftarrow r\text{-Rewrite}(c, t)$

13: $dP' \leftarrow dP$

14: $dM' \leftarrow update(dM, c)$

15: **end if**

16: **return** (t', dP', dM')

17: **end function**

The function $occurrence_L/occurrence_G$ constructs a dictionary of all the patterns and their locations in the tree/forest for potential rewrite. The function $bestCandidate$ in both algorithms is defined by comparing the *net memory savings* of the rewrite.⁹ Given a tree t , the local frequency of a *composition template* of the form $p = (i, g, f)$, $Freq_L^P(t, p)$, is equal to the maximal non-overlapping occurrences of the pattern within a tree. Its global frequency within a forest T , denoted as $Freq_G^P(T, p)$ is the number of pieces where it is present; if it occurs multiple times in a tree of the forest, it still contributes 1 to the global frequency.

The local frequency of $m \in \mathcal{M}$, $Freq_L^R(m)$ is the sum of all local frequencies of the patterns that match a *replication template* (g, m, \vec{f}) within a tree.¹⁰ The global frequency $Freq_G^R(T, m)$ simply counts the number of pieces where it occurs.

$$Freq_L^R(t, m) = \sum_{g \in label(t)} Freq_L^P((g, m, children(g))) \quad (1)$$

⁹ A net memory saving of a pattern defined as its unit memory saving multiplied by its number of occurrence minus the storage cost. If a pattern is already in the dictionary, then the storage cost is zero.

¹⁰ Similarly with the “non-overlapping” constraint of derivation patterns, we do not count re-occurrences of a meta-rule in a node if the same meta-rule also occurs in any of its children on a repeating position.

Algorithm 2 The Global Compression Algorithm (a single step)

```

1: Input

     $dE$  : Dictionary from piece-id to  $(t, dP, dM)$ 
     $dP_G$  : Dictionary from symbols to global templates
     $dM_G$  : Dictionary from symbols to global meta-rules

2: function  $compress_G(dE, dP_G, dM_G)$ 
3:    $(d_P^o, d_M^o) \leftarrow occurrence_G(dE)$ 
4:    $c \leftarrow bestCandidate(d_P^o, d_M^o)$ 
5:   if  $c = Nothing$  then
6:      $dE' \leftarrow compress_L$  over  $dE$ , only update
7:     local tables if the rule is not in  $dP_G$  or  $dM_G$ 
8:      $dP'_G \leftarrow dP_G$ 
9:      $dM'_G \leftarrow dM_G$ 
10:  else if  $c \in d_P^o$  then
11:     $dE' \leftarrow$  apply  $p$ -Rewrite( $c$ ) over trees in  $dE$ 
12:     $dP'_G \leftarrow update(dP_G, c)$ 
13:     $dM'_G \leftarrow dM_G$ 
14:  else if  $c \in d_M^o$  then
15:     $dE' \leftarrow$  apply  $r$ -Rewrite( $c$ ) over trees in  $dE$ 
16:     $dP'_G \leftarrow dP_G$ 
17:     $dM'_G \leftarrow update(dM_G, c)$ 
18:  end if
19:  return  $(dE', dP', dM')$ 
20: end function
    
```

As an example, in the tree in Fig 1b, the *meta-rule* $m_1 = \langle _ , 1 \rangle$ occurs six times¹¹ while the *template* $T_1 = (_ \searrow_5, m_1, *)$ occurs four times.¹²

The size for a *composition template* (i, g, f) is always 3 when g and f are stored in memory, because we only need two symbols to represent them, and an integer to specify the location at which they are composed. The *unitSave* of a *composition template* is always 1, as it replaces two nodes in the tree with one, thus decreasing the tree size by 1. The size of a *replication template* (g, m, \vec{f}) is $2 + sizeIn(m)$ for similar reason. The *unitSave* of meta-rule m is $sizeOut(m) - sizeIn(m)$ as the difference represents the number of symbol repeats.

5. EXAMPLE APPLICATION: REPETITION MINING ON THE JAZZ HARMONY TREE BANK

To exemplify an application of the proposed model, we turned to the Jazz Harmony Tree Bank dataset [45], which contains 150 pieces with annotated harmony labels. The annotations are in accordance with the formal-grammar trees of [15]. Since our focus is on modeling repeated relations among chords, rather than the chords alone, we need to transform our input from chord-labeled trees to rule-labeled trees in the same fashion as in Figure 1b. To this end, for every node in the tree we match chord labels with abstract production rules of [15]. The resulting rule-

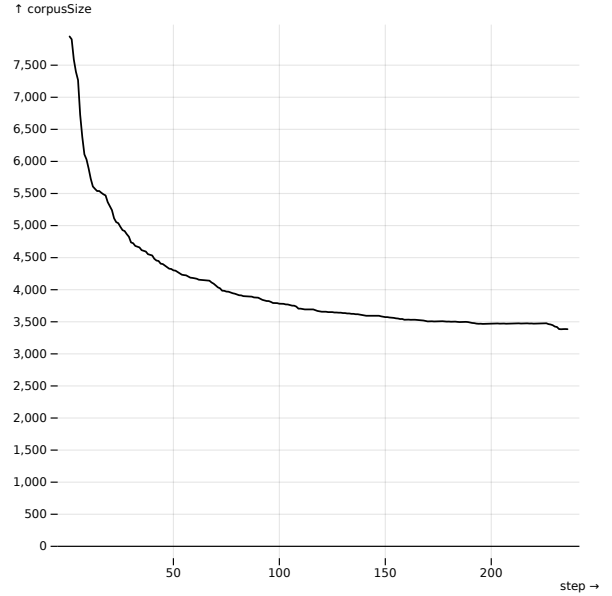


Figure 5: Corpus size plotted against the number of global compression steps. The overall reduction in corpus size is from 7948 to 3385.

labeled trees are ranked trees,¹³ which is a required property for the compression algorithms (see [37]).

5.1 Results

In global compression, the decrease in tree size in the corpus is partly offset by the expansion of the global-rule table.¹⁴ In initial iterations, this trade-off is highly advantageous, as only few instances of certain patterns in the corpus are needed to offset the cost of storing rules. Thereafter, as shown in Figure 5, the compression size rapidly converges, demonstrating the efficiency of this process. It is worth noting a slight upward trend in the curve towards the end. This is because the algorithm does not mandate a reduction of the corpus size at each step; rather, it extracts patterns as long as they occur twice, whether locally or globally.

Following global compression, each piece is represented in a significantly more condensed format, utilizing the global-rule table. As shown in Figure 6, all pieces are compressed to at least 2/3 of their original size. In particular, four pieces are compressed to the size of one.¹⁵ These pieces and their changes are the following: “Equinox” (23 to 1), “Mr. P.C.” (23 to 1), “Subconscious Lee” (63 to 1), and “Hot House” (63 to 1). Notably, the first two pieces and the last two have the same rule-labeled tree representations respectively, indicating that they are derived in the same manner despite differences in chord labels. They compress to size 1, because their entire “piece” patterns occur at least twice and the algorithm therefore identifies

¹³ A ranked tree guarantees that the same symbol has the same arity. In a chord-labeled tree, a chord symbol can occur in both branching node (*arity* > 0) or leaf (*arity* = 0); such a tree is thus not ranked.

¹⁴ The size of a piece is defined by the sum of the size of the template nodes in the tree.

¹⁵ Only two of the pieces whose compressed size is equal to one are visible in the plot due to overlaying.

¹¹ as opposed to eight times because of the non-overlapping constraint

¹² This is made more explicit by the reduced tree in Figure 4a.

them as global *templates*. Table 1 summarizes the number of rules obtained after global compression of the corpus. The majority of the rules extracted are global, suggesting many common derivation patterns and meta-rules repeated in multiple pieces. To our surprise, all meta-rules found are global.¹⁶

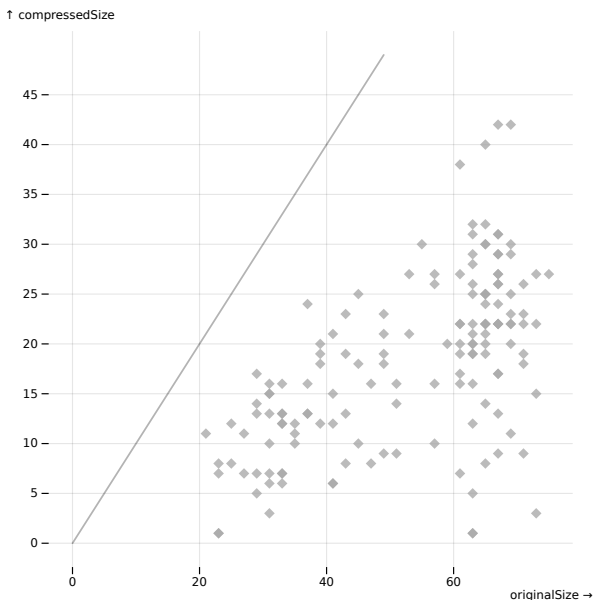


Figure 6: Distribution of piece size before (x -axis) and after (y -axis) global compression. Each dot corresponds to a piece in the corpus. The indicator line represents 1-1 compression rate. Sizes refer to the individual compressed trees alone, not counting the size of global rule tables.

	Global (stylistic)	Local (piece-specific)
<i>Template</i>	198	20
<i>Meta-rule</i>	36	0

Table 1: The numbers of rules after global compression.

6. DISCUSSION AND CONCLUSION

We have presented a formal description and computational model of structural repetition in music, which also accounts for variations. In seeking a compressed representation of a forest of abstracted syntax trees, our goal has been to unveil *what* is actually repeated in a fundamental sense, and *how* entities recur within a specific style. To this end, we proposed a forest compression algorithm based on two rewrite operations, each catering to distinct musical abstractions.

The discovered global *meta-rules* (see Figure 7) include some of the hand-coded *meta-rules* outlined in [34] in line with musical intuition: $\alpha\alpha$ (M_1), $\alpha\alpha\beta$ (M_{64}) and $\alpha\alpha\beta\alpha$ (M_{62}). Also discovered are meta-rules such as $\alpha\beta\alpha\gamma$ (M_{20}), akin to (but not necessarily identical with) the “period” in standard contemporary form theory [10],

¹⁶ We think this is due to the abstract, general nature of *meta-rules*, which makes their recurrence in 150 pieces highly probable.

$M_1 = \langle _ , 1 \rangle$	$M_4 = \langle \star , _ \rangle$	$M_9 = \langle _ , _ , 1 , 2 \rangle$
$M_{13} = \langle \star \rangle$	$M_{16} = \langle _ , \star \rangle$	$M_{17} = \langle _ , 1 , 1 , 1 \rangle$
$M_{20} = \langle _ , _ , 1 , _ \rangle$	$M_{27} = \langle _ , _ , _ , 2 \rangle$	$M_{62} = \langle _ , 1 , _ , 1 \rangle$
$M_{63} = \langle _ , 1 , _ , _ \rangle$	$M_{64} = \langle _ , 1 , _ \rangle$	$M_{65} = \langle _ , _ , _ , 1 \rangle$
$M_{100} = \langle _ , _ , 1 , 1 \rangle$	$M_{101} = \langle _ , _ , _ , 3 \rangle$	$M_{148} = \langle _ , _ , 2 , _ \rangle$

Figure 7: The first 15 discovered global *meta-rules* (whose length is less than 5) out of total 36. The index n indicates the n -th discovered global pattern, including both *template* and *meta rules*.

as well as $\alpha\beta\gamma\alpha$ (M_{65}), which resembles an expanded ternary structure. *Meta-rules* with parent-child repeats (e.g. M_4, M_{13}, M_{16}) emerge quite early in the compression process. The *meta-rule* $\langle \star \rangle$ is the simplest way to nest a pattern. For example, applying it to the template “V region followed by I chord” results in the template “V/V region followed by V chord followed by I chord.”¹⁷ We believe such recursive repetitions of the same pattern are, in general, highly meaningful in music. By analyzing the compression rate of the individual pieces after global compression, one could argue that pieces with higher compression rates are generally likely to correspond to more “conventional” expressions of a style. In future research, considering the compression rate of individual pieces could shed light onto their stylistic attributes and patterns of interaction between style and structure.

We consider the distinction between global and local abstraction meaningful both for music interpretation and its computational representation. Global abstractions enable the creation of more efficient representations of an entire corpus in comparison with intra-piece, local compression. From a music-theoretical perspective, intertextual study is inextricable from the notion of style. For instance, while a ternary form may appear only once within a piece, analysts would still recognize it as a conventional entity because it recurs across the style. Galant schemata [18] can similarly be thought of as collections of stylistic patterns. Form archetypes such as *AABA* and *ABA* can likewise be seen as global meta-rules.

By integrating additional types of constraints, the model has the potential, with certain extensions, to express more sophisticated repetitions. For example, Schoenberg’s notion of “liquidation” [46] could be recast as the repetition of abstract relations constrained by decreasing elaboration depth. The notion of “fragmentation” [10] could also be modeled as repetition with a constraint on ordering, so that fragments appear only after the initial, integral structure. Our framework could also find use in algorithmic music generation under grammatical constraints. For instance, one could generate a piece in top-down fashion by sampling patterns and meta-rules discovered within a stylistically homogeneous corpus.

¹⁷ “Region” here indicates non-terminal symbol while “chord” indicates terminal symbol.

7. ACKNOWLEDGMENTS

This project has received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation program under grant agreement No 760081 – PMSB. We thank Claude Latour for supporting this research through the Latour Chair in Digital Musicology. We thank the team of the Digital and Cognitive Musicology Lab (DCML), particularly Xinyi Guan, Gabriele Cecchetti and Ran Tamir, for their helpful comments on this paper. We would also like to express our gratitude towards the conference reviewers for their constructive feedbacks.

8. REFERENCES

- [1] F. Salzer, *Structural hearing: Tonal coherence in music*. Courier Corporation, 1962, vol. 1.
- [2] L. M. Zbikowski, “Musical coherence, motive, and categorization,” *Music Perception*, vol. 17, no. 1, pp. 5–42, 1999.
- [3] —, *Conceptualizing Music: Cognitive Structure, Theory, and Analysis*. Oxford University Press, 2002.
- [4] E. T. Hall and M. T. Pearce, “A model of large-scale thematic structure,” *Journal of New Music Research*, vol. 50, no. 3, pp. 220–241, 2021.
- [5] E. H. Margulis, *On Repeat: How Music Plays the Mind*. Oxford University Press, 2014.
- [6] P. C. Van den Toorn, “What’s in a motive? Schoenberg and Schenker reconsidered,” *The Journal of Musicology*, vol. 14, no. 3, pp. 370–399, 1996.
- [7] D. Beach, “Motivic repetition in Beethoven’s Piano Sonata op. 110 part i: The first movement,” *Intégral*, pp. 1–29, 1987.
- [8] C. Burkhart and H. Schenker, “Schenker’s” motivic parallelisms,” *Journal of Music Theory*, vol. 22, no. 2, pp. 145–175, 1978.
- [9] A. Cadwallader and W. Pastille, “Schenker’s high-level motives,” *Journal of Music Theory*, vol. 36, no. 1, pp. 119–148, 1992.
- [10] W. E. Caplin, *Classical Form: A Theory of Formal Functions for the Instrumental Music of Haydn, Mozart, and Beethoven*, 1st ed. Oxford Univ. Press, 2001.
- [11] Y. Greenberg, *How Sonata Forms: A Bottom-up Approach to Musical Form*. Oxford University Press, 2022.
- [12] D. Smyth, “Balanced Interruption” and the Formal Repeat,” *Music Theory Spectrum*, vol. 15, no. 1, pp. 76–88, 1993.
- [13] R. Ivanovitch, “What’s in a theme? On the nature of variation.” *Gamut: The Online Journal of the Music Theory Society of the Mid-Atlantic*, vol. 3, no. 1, 2010.
- [14] W. Frisch, *Brahms and the principle of developing variation*. University of California Press, 1990, no. 2.
- [15] M. Rohrmeier, “The Syntax of Jazz Harmony: Diatonic Tonality, Phrase Structure, and Form,” *Music Theory and Analysis (MTA)*, vol. 7, no. 1, pp. 1–63, 2020-04-30. [Online]. Available: <https://www.ingentaconnect.com/content/10.11116/MTA.7.1.1>
- [16] J.-J. Nattiez, *Music and Discourse: Toward a Semiology of Music*. Princeton University Press, 1990.
- [17] E. Narmour, “Some major theoretical problems concerning the concept of hierarchy in the analysis of tonal music,” *Music Perception*, vol. 1, no. 2, pp. 129–199, 1983.
- [18] R. Gjerdingen, *Music in the galant style*. OUP USA, 2007.
- [19] G. Sanguinetti, *The art of partimento: history, theory, and practice*. OUP USA, 2012.
- [20] J. F. Boss, “Schenkerian-Schoenbergian analysis’ and hidden repetition in the opening movement of Beethoven’s Piano Sonata op. 10, no. 1,” *Music Theory Online*, vol. 5, no. 1, 1999.
- [21] J. Yust, *Organized Time: Rhythm, Tonality, and Form*, ser. Oxford Studies in Music Theory. Oxford University Press, 2018.
- [22] P. Westergaard, *An Introduction to Tonal Theory*. Norton, 1975.
- [23] M. Rohrmeier, “Towards a generative syntax of tonal harmony,” *Journal of Mathematics and Music*, vol. 5, no. 1, pp. 35–53, 2011-03. [Online]. Available: <http://www.tandfonline.com/doi/abs/10.1080/17459737.2011.573676>
- [24] —, “Towards a formalization of musical rhythm.” in *ISMIR*, 2020, pp. 621–629.
- [25] C. Finkensiep, R. Widdess, and M. A. Rohrmeier, “Modelling the syntax of north indian melodies with a generalized graph grammar,” in *Proceedings of the 19th International Society for Music Information Retrieval Conference*. ISMIR, 2019, pp. 462–269.
- [26] E. Cambouropoulos, “Towards a general computational theory of musical structure,” Ph.D. dissertation, The University of Edinburgh, 1998.
- [27] I. Y. Ren, H. V. Koops, A. Volk, and W. Swierstra, “In search of the consensus among musical pattern discovery algorithms,” in *Proceedings of the 18th ISMIR*. ISMIR press, 2017, pp. 671–678.

- [28] O. Melkonian, I. Y. Ren, W. Swierstra, and A. Volk, "What constitutes a musical pattern?" in *Proceedings of the 7th ACM SIGPLAN International Workshop on functional art, music, modeling, and design*, 2019, pp. 95–105.
- [29] A. Laaksonen and K. Lemström, "Transposition and time-warp invariant algorithm for detecting repeated patterns in polyphonic music," in *Proceedings of the 6th International Conference on Digital Libraries for Musicology*, 2019, pp. 38–42.
- [30] J.-L. Hsu, C.-C. Liu, and A. L. Chen, "Discovering nontrivial repeating patterns in music data," *IEEE Transactions on multimedia*, vol. 3, no. 3, pp. 311–325, 2001.
- [31] C. Wang, J. Li, and S. Shi, "N-gram inverted index structures on music data for theme mining and content-based information retrieval," *Pattern recognition letters*, vol. 27, no. 5, pp. 492–503, 2006.
- [32] I. Karydis, A. Nanopoulos, and Y. Manolopoulos, "Finding maximum-length repeating patterns in music databases," *Multimedia Tools and Applications*, vol. 32, pp. 49–71, 2007.
- [33] B. Janssen, W. B. De Haas, A. Volk, and P. Van Kranenburg, "Finding repeated patterns in music: State of knowledge, challenges, perspectives," in *Sound, Music, and Motion: 10th International Symposium, CMMR 2013, Marseille, France, October 15-18, 2013. Revised Selected Papers 10*. Springer, 2014, pp. 277–297.
- [34] C. Finkensiep, M. Haerberle, F. Eisenbrand, M. Neuwirth, and M. A. Rohrmeier, "Repetition-structure inference with formal prototypes," in *ISMIR 2023 Hybrid Conference*, 2023.
- [35] E. Lehman and A. Shelat, "Approximation algorithms for grammar-based compression," in *Proceedings of the thirteenth annual ACM-SIAM symposium on Discrete algorithms*. Citeseer, 2002, pp. 205–212.
- [36] T. Akutsu, "A bisection algorithm for grammar-based compression of ordered trees," *Information Processing Letters*, vol. 110, no. 18-19, pp. 815–820, 2010-09. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0020019010002085>
- [37] M. Lohrey, S. Maneth, and R. Mennicke, "Tree Structure Compression with RePair," in *2011 Data Compression Conference*. IEEE, 2011-03, pp. 353–362. [Online]. Available: <http://ieeexplore.ieee.org/document/5749493/>
- [38] M. Lohrey, "Grammar-Based Tree Compression," in *Developments in Language Theory*, I. Potapov, Ed. Springer International Publishing, 2015, vol. 9168, pp. 46–57. [Online]. Available: https://link.springer.com/10.1007/978-3-319-21500-6_3
- [39] P. Bille, I. L. Gørtz, G. M. Landau, and O. Weimann, "Tree compression with top trees," *Information and Computation*, vol. 243, pp. 166–177, 2015-08. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0890540114001643>
- [40] A. Gascón, M. Lohrey, S. Maneth, C. P. Reh, and K. Sieber, "Grammar-Based Compression of Unranked Trees," *Theory of Computing Systems*, vol. 64, no. 1, pp. 141–176, 2020-01. [Online]. Available: <http://link.springer.com/10.1007/s00224-019-09942-y>
- [41] C. G. Nevill-Manning and I. H. Witten, "Identifying Hierarchical Structure in Sequences: A linear-time algorithm," *Journal of Artificial Intelligence Research*, vol. 7, pp. 67–82, 1997-09-01. [Online]. Available: <https://www.jair.org/index.php/jair/article/view/10192>
- [42] T. J. O'Donnell, *Productivity and Reuse in Language: A Theory of Linguistic Computation and Storage*. The MIT Press, 2015.
- [43] A. K. Joshi and Y. Schabes, "Tree-adjoining grammars," in *Handbook of Formal Languages: Volume 3 Beyond Words*. Springer, 1997, pp. 69–123.
- [44] A. K. Joshi, "An introduction to tree adjoining grammars," *Mathematics of language*, vol. 1, pp. 87–115, 1987.
- [45] D. Harasim, C. Finkensiep, P. Ericson, T. J. O'Donnell, and M. Rohrmeier, "The jazz harmony treebank," in *21st ISMIR, Montréal, Canada, October 11-16, 2020*, 2020, pp. 207–215.
- [46] A. Schoenberg, G. Strang, and L. Stein, *Fundamentals of Musical Composition*. Faber & Faber, 1999.

SARAGA AUDIOVISUAL: A LARGE MULTIMODAL OPEN DATA COLLECTION FOR THE ANALYSIS OF CARNATIC MUSIC

Adithi Shankar Genís Plaja-Roglans Thomas Nuttall
Martín Rocamora Xavier Serra

Music Technology Group, Universitat Pompeu Fabra, Barcelona, Spain

adithishankar.sivasankar@upf.edu

ABSTRACT

Carnatic music is a style of South Indian art music whose analysis using computational methods is an active area of research in Music Information Research (MIR). A core, open dataset for such analysis is the Saraga dataset, which includes multi-stem audio, expert annotations, and accompanying metadata. However, it has been noted that there are several limitations to the Saraga collections, and that additional relevant aspects of the tradition still need to be covered to facilitate musicologically important research lines. In this work, we present Saraga Audiovisual, a dataset that includes new and more diverse renditions of Carnatic vocal performances, totalling 42 concerts and more than 60 hours of music. A major contribution of this dataset is the inclusion of video recordings for all concerts, allowing for a wide range of multimodal analyses. We also provide high-quality human pose estimation data of the musicians extracted from the video footage, and perform benchmarking experiments for the different modalities to validate the utility of the novel collection. Saraga Audiovisual, along with access tools and results of our experiments, is made available for research purposes.

1. INTRODUCTION

In recent years, there has been an increasing emphasis on representing non-Western classical music styles within computational musicology [1, 2], an interdisciplinary research area involving musicology and computer science. To facilitate this research, many repertoire-specific datasets have been proposed that take into account the melodic, rhythmic and structural complexities of these traditions. Several of them are consolidated within the scope of Dunya, a collection of large music corpora dedicated to fuelling research of five major non-Western music traditions: Carnatic music, Hindustani music, Turkish Makam, Beijing Opera and Arab-Andalusian music [1].

One style of particular interest is Carnatic music, for which there has been numerous computational musicological studies carried out using the Saraga dataset, a subset of the Dunya corpora dedicated to the Indian Art Music (IAM) traditions of Hindustani and Carnatic music [3–6]. The Carnatic portion of this dataset comprises performance audio, expert/automatically extracted annotations, and associated relevant metadata [7].

The audio data includes the mixture and, for a number of recordings, multi-track signals for all instrument sources except the *tānpūrā*. Since Carnatic music is primarily performed and enjoyed in a live performance setting, the audio recordings gathered for the Saraga dataset are all recorded in this context, and hence contain some leakage interference in the individual stem signals.

Alongside these audio recordings, Saraga provides automatically extracted annotations, such as the predominant pitch track of the vocalist’s melody and rhythmic beats, and manual annotations, such as melodic patterns and musical sections. Finally, the dataset includes editorial metadata such as performer names, concert/composition titles, and musical tags such as melodic (*rāga*) and rhythmic (*tāla*) modes, which are crucial for this repertoire.

Whilst Saraga has proven to be a valuable resource for the analysis of IAM, there are nonetheless many challenges and important research questions for Carnatic music for which Saraga is insufficient. Some of these deficiencies – such as representativeness (e.g., instrument diversity, number of *rāgas*, demographics), completeness of annotations, and data access – have been pointed out in its open peer-review [8]. However, no new version of the dataset addressing said problems has been made available, and hence such deficiencies persist. Furthermore, Saraga contains automatically extracted features, which although may have been state-of-the-art at the time, could well be improved using more modern algorithms [9] and models [10].

In this work, we introduce *Saraga Audiovisual*, a new dataset built according to the principles of the original Saraga, that encompasses 42 new concerts totalling more than 60 hours of Carnatic music recordings. By including new artists, compositions, *rāgas*, and *tālas*, we improve the diversity and representativeness of the data. The new collection comprises multi-track audio, video recordings, and human pose estimation data, the latter two of which are entirely new modalities which are currently not considered in the first Saraga dataset. We hope that this multimodal data



© A. Shankar, G. Plaja-Roglans, T. Nuttall, M. Rocamora, and X. Serra. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** A. Shankar, G. Plaja-Roglans, T. Nuttall, M. Rocamora, and X. Serra, “Saraga Audiovisual: a large multimodal open data collection for the analysis of Carnatic Music”, in *Proc. of the 25th Int. Society for Music Information Retrieval Conf.*, San Francisco, United States, 2024.

will power further research of musically relevant problems in Carnatic music and encourage the development of underexplored research strands, particularly in music’s visual and kinetic aspects [11–13]. We also improve documentation, access, and tools, considering the issues raised for Saraga [8], and provide a detailed description of the new dataset regarding musical metadata and coverage.

To showcase the value of the new dataset we present two benchmarking experiments to (a) demonstrate that the features extracted from the new audio-visual data are useful for the analysis of codependencies between performer body movement and vocalisations, an active research area in IAM analysis [14–21]; and (b) show that the novel multi-track audio is valuable for Music Source Separation (MSS) in Carnatic music, a low-level feature extraction task for which distributed pre-trained models in the literature do not generalize [22].

2. BACKGROUND AND RELATED WORK

Digital technology has brought new research methods to musicology [23, 24]. With digital archives and computer science techniques, researchers can study music corpora more systematically and quantitatively [25, 26]. Hence, creating appropriate datasets and research corpora for different music traditions is a fundamental concern in music information research (MIR) [27–32]. Computational musicological studies have used various data sources: scanned sheet music, symbolic scores, audio/video recordings, and motion capture data [11–13, 21, 33–36].

With few exceptions [36, 37],¹ almost all openly available datasets in the literature for Carnatic and Hindustani music are compiled from the IAM corpora in CompMusic [28], and more recently, from the Saraga dataset, for which multi-track audio recordings and manual and automatically extracted annotations are available [7].

Dataset distribution is a major concern in the music information research community [38], in which data plays a key role, especially given the advent of DL models. Saraga is currently accessed through Python notebooks, but the process is complex, not standardized, and hindered by bugs and dependency incompatibilities. Such a distribution method requires regular maintenance, which is expensive and time-consuming. A unified and functioning access point for the canonical version of the dataset, and a documented toolkit to browse through the recordings and annotations are not available.

One other important limitation of Saraga is that it contains only audio recordings. However, music is not only an auditory experience; multiple lines of research have demonstrated that the visual and kinetic aspects are all part of what music fundamentally is [39–41]. Thus, a comprehensive study of music performance requires auditory, visual and kinetic components [11].

In the case of Carnatic music, visual cues like hand/head gestures and performer gaze can provide the artists con-

textual information for an improved dynamic on stage, whilst also playing a more individualistic expressive role. This has been investigated in various IAM studies using motion capture data and pose estimation extracted from video [14–21]. For this reason, a dataset of Carnatic music should ideally include as much of this multimodality as possible, which we are enabling through the contribution of the video recordings in the proposed dataset.

3. DATASET DESCRIPTION

Saraga Audiovisual aims to address some of the aforementioned issues attributed to the first version of Saraga. In this section, we present the proposed improvements, which are mainly based on fixes, new recordings, and the novel visual modality. Although the new dataset falls entirely in the Carnatic repertoire, the proposed pipeline could be extended to Hindustani Music in the future.

3.1 New concerts

A total of 42 new concerts are released as part of Saraga Audiovisual, including multi-track audio and video for all concerts. The multi-track audio covers three main stems: vocals, violin, and mṛdaṅgam for all renditions, with the addition of ghaṭam and tānpūrā for 9 other concerts. The audios are all stereophonic and encoded at 44.1 kHz. Since the audio is recorded during live performances, the individual stems contain interference from the other sources.

These 42 concerts consist of a total of 235 individual performances of 223 unique compositions from 131 lead and accompanying artists. All performances include manually annotated section annotations. 10 distinct tālas and 113 distinct rāgas are represented, an increase of 55 on the existing Saraga dataset. Figure 1 shows the combined statistics for the case of the frequency of occurrence of the same rāga performances over Saraga and Saraga Audiovisual. Our aim is to increase the representation of existing rāgas whilst including unrepresented rāgas.

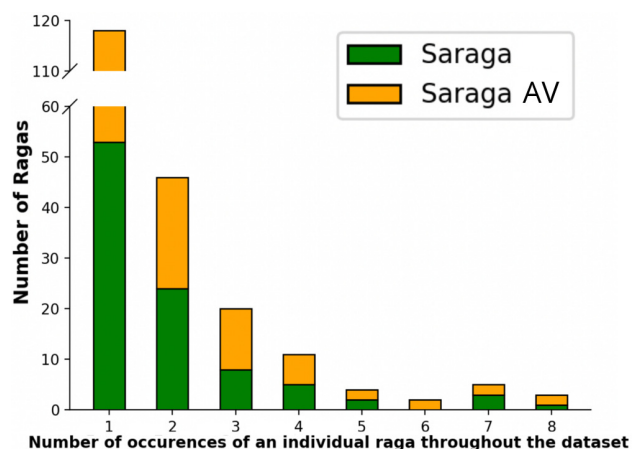


Figure 1. Number of occurrences of individual rāgas combining the two datasets. X-axis represents number of occurrences, whilst Y-axis indicates how many rāgas there are with 1, 2, ..., 8 occurrences.

¹ Using the IEMP North Indian Rāga: <https://osf.io/ks325/>, and Karnatak ālāpāna multimodal dataset: <https://osf.io/6huvd/> respectively

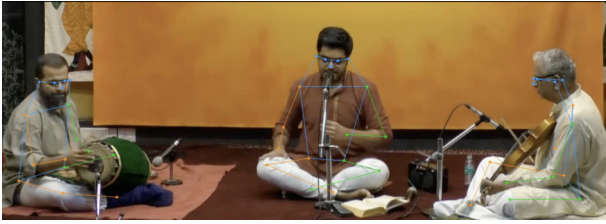


Figure 2. A video frame fragment from Saraga Audiovisual. The lead singer is VR Raghava Krishna, the violinist is VV Ravi and the mridangist is Guru Raghavendra.

Content	Saraga	Saraga AV
Total number of recordings	249	235
Total number of artists	64	131
Number of compositions	202	223
Total number of rāgas	110	235
Unique rāgas in collection	96	113
Total number of tālas	10	10
Total duration of the dataset	52.7	64.8

Table 1. Content comparison of the Carnatic subset of Saraga and the Saraga Audiovisual dataset (Saraga AV).

The most popular performance format today is vocal-led, either by a single or multiple vocalists. Despite the fair criticism of the shortage of instrumental recordings [8], we decide to consider only concerts led by a singer in Saraga Audiovisual. Moreover, the singing voice is extensively explored in the MIR literature, with numerous models designed to address various problems and research questions, offering opportunities for leveraging, training, and fine-tuning functional systems. We refine the statement around representativeness of Saraga to clarify that our dataset is intentionally vocal-centered.

3.2 Video recordings and human pose estimation

The videos corresponding to the concerts are rendered at 1080p and have a frame rate of 25 fps. They are recorded with a fixed wide-view position to frame all performing artists throughout the concert.

Figure 2 depicts the recording setting. The videos are recorded in a traditional concert set up with microphones occluding the view of the artists at most times. For example, if we observe a singer in this setting, they are in a seated position with the microphone directed towards their mouth. Consequently, the microphone head occludes the mouth, and the stand hampers the view of the singer’s hands in several instances. In general, occlusions can hinder human pose estimation by a very large margin. After careful examination of several human pose estimation models, we choose MMPose [42], a DL model which performs extremely well, given the tricky setting. We extract human skeletons with 17 key points through its 2-D model. See Figure 3 for an example of the gesture estimation on a Saraga Audiovisual example video recording.

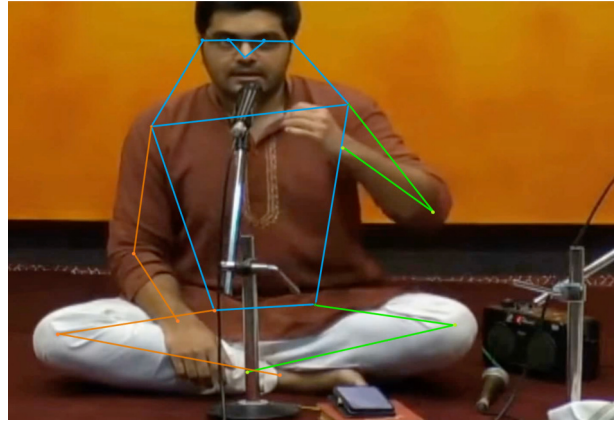


Figure 3. Gesture extraction with MMPose. The artist depicted in the figure is VR Raghava Krishna

3.3 Improving dataset access

Hassle-free and canonical access to the Saraga audio, annotations, and also metadata is an issue raised by the community [8]. We implement a `mirdata`² loader for Saraga Audiovisual to download, load, and browse through the canonical dataset and easily filter the data by musically important aspects such as rāga, tāla, artist, and tonic. These functions are also available through `compIAM`³, where models and algorithms for the computational analysis of Carnatic music are also available.

3.4 Further improvements

Note that some of the features in Saraga are automatically extracted. Despite not being manually collected, these allow for faster, consistent, and reproducible research as we bypass the need to compute them multiple times. Since Saraga was first published, much research has been carried out by the MIR community, and new models to more reliably extract such features are available. Within the context of this work we compute the melody curves of the novel recordings using the Carnatic-optimized FTA-Net [10].

4. EXPERIMENTS

In this section we present two experiments using the audio and visual components of Saraga Audiovisual.

4.1 Multimodal study

There exists various studies that demonstrate the relationship between gesture and musical motifs in an IAM context [14–21]. Whilst most focus on the North Indian, Hindustani style, a recent study by Pearson et al. presents a quantitative attempt at characterising codependencies between the body movement and vocalisations of Carnatic performers using a combination of predominant pitch tracks extracted from audio, and motion tracking data captured using an inertial measurement system on the body during performance [37]. In an effort to demonstrate the value of

² <https://github.com/mir-dataset-loaders/mirdata>

³ <https://github.com/MTG/compIAM>

Performer	Rāga	Dur.
Ashwin Srikant	Śīṃhēndramadhyamam	09:17
Raghava Krishna	Śīṃhēndramadhyamam	05:36
Aditi Prahlad	Pūrvīkalyāṇī	08:15
Prithvi Harish	Pūrvīkalyāṇī	08:01

Table 2. Saraga Audiovisual performances used for multimodal experiment in Section 4.1. Duration’s (mm:ss) correspond to the rāga ālāpana section of the performance, the rest of the performance is not used for analysis.

Saraga Audiovisual in supporting such studies, we reproduce a part of Pearson et al’s analysis here using data extracted from the proposed multimodal dataset. In the original study, performer gesture data is extracted using motion capture equipment, since here we rely on inferring this information from video, we make some changes to that part of the process, outlined in the following section. All other steps remain identical to the original study and we refer the reader to the paper for more details.

4.1.1 Experimental setup

We reproduce Pearson et al’s Analysis 1: *Do sonic motif DTW distances covary with spatiotemporal patterns of gesture?* Our sonic data for such analysis is extracted from 4 performances (Table 2) in Saraga Audiovisual, from which we extract 4 time series corresponding to the rāga ālāpana section of the performance audio; (1) f_0 – the predominant vocal melodic line, measured in cents above the performer tonic, extracted using a Carnatic-specific methodology [10], (2) Δf_0 – the first derivative of f_0 , (3) loudness, L , computed as $L = 10 \cdot \log_{10} \frac{S}{ref}$, where S is the power spectrum of the raw audio signal and ref is its maximum value, and (4) the spectral centroid of the raw audio signal. Our gestural data is extracted using MMPose and limited to the performer’s left and right hands (see Section 3.1, and Figure 3), from which we compute the first and second derivatives to obtain two subsequent time series of velocity and acceleration, respectively, resulting in 6 gestural time series. The 6 time series are resampled so as to have identical sampling rates of 24 Hz, and all 10 time series are smoothed using a 2nd-order Savitzky-Golay filter with a window length of 125 ms.

In each of the 4 performances, we identify regions of repeated melodic motifs using a Carnatic-specific methodology [4]. For each motif, we isolate the corresponding segment in our 4 sonic and 6 gestural time series, and discard the gestural time series corresponding to the non-dominant hand. The dominant hand of the performer for each pattern is determined as that which has the highest kinetic energy, $K.E$, computed from the velocity tracks, v , where $K.E = \frac{mv^2}{2}$ and m is the mass of the moving body part, assumed equal for both sides. We note that for over 98% of the identified motifs, the ratio in $K.E$ between the dominant and non-dominant hand is greater than 1.2, i.e. there is almost always a clear dominant hand. 70% of motifs are identified as left-handed and 30% as right-handed. The

gesture space for each motif is transformed such that left and right-hand gestures occur in the same space by mirroring all right-handed gestures in the y-axis, and such that the gestural space origin corresponds to the centroid of the body of the performer. This centroid is determined for each motif as the centroid of the trapezoid corresponding to the performer’s body, provided by MMPose and visible in Figure 3.2. The result is a selection of 269 non-overlapping motifs across the 4 performances, each represented by 4 sonic time series (f_0 , Δf_0 , loudness and spectral centroid) and 3 gestural time series (position, velocity and acceleration of the dominant hand).

For each pairwise combination of our 269 motifs, we compute the dynamic time warping (DTW) distance between each of their 6 sonic time series and 4 gestural time series, i.e. f_0 compared to f_0 , hand position compared to hand position etc... Motifs are not compared to themselves and as such this constitutes 36,046 motif pairs. This analysis is concerned with whether there exists a relationship between the DTW distances of sonic and gestural features (sonic features: f_0 , Δf_0 , loudness and spectral centroid; gestural features: hand position, velocity and acceleration). For each combination of sonic to gestural features, we compute Spearman’s rank correlation coefficient to quantify this relationship, both on a performer level and across all performers.

4.1.2 Results

The correlation analysis results are presented in Figure 4. Tests for which the p-value is greater than our significance level of 0.0001 are excluded and replaced with a grey square in the heatmap. It is not within the scope of this paper to discuss the results of this analysis in detail, nor do we consider the size of the data analysed sufficient to make any meaningful conclusions (0.5 hours of performance compared to 3.8 in the original study). We do, however, emphasize that even with this limited scope, we are able to identify significant relationships between sonic motif distances and spatiotemporal patterns of gesture using the Saraga Audiovisual dataset, corroborating the results of Pearson et al.’s study. As in that study, we show loudness as having the strongest correlation with gestural features across the performers and demonstrate how distinct the individual performer gesturing styles are, with great variation in the extent to which performers’ gestures correlate with f_0 and spectral centroid.

4.2 Fine-tuning MSS models with data with bleeding

The current state-of-the-art MSS models are based on DL architectures which are trained using multi-track recordings. Some models that are widely used, namely Spleeter [43] or Demucs [44], are trained with multi-track stems available through datasets like MUSDB18HQ [45] or MoisesDB [46], mainly including Western pop styles or related genres. Although many research works on the analysis of IAM use the available Spleeter model for source separation [13,47,48], these models do not generalize well for Carnatic music due to its varied instrumentation and

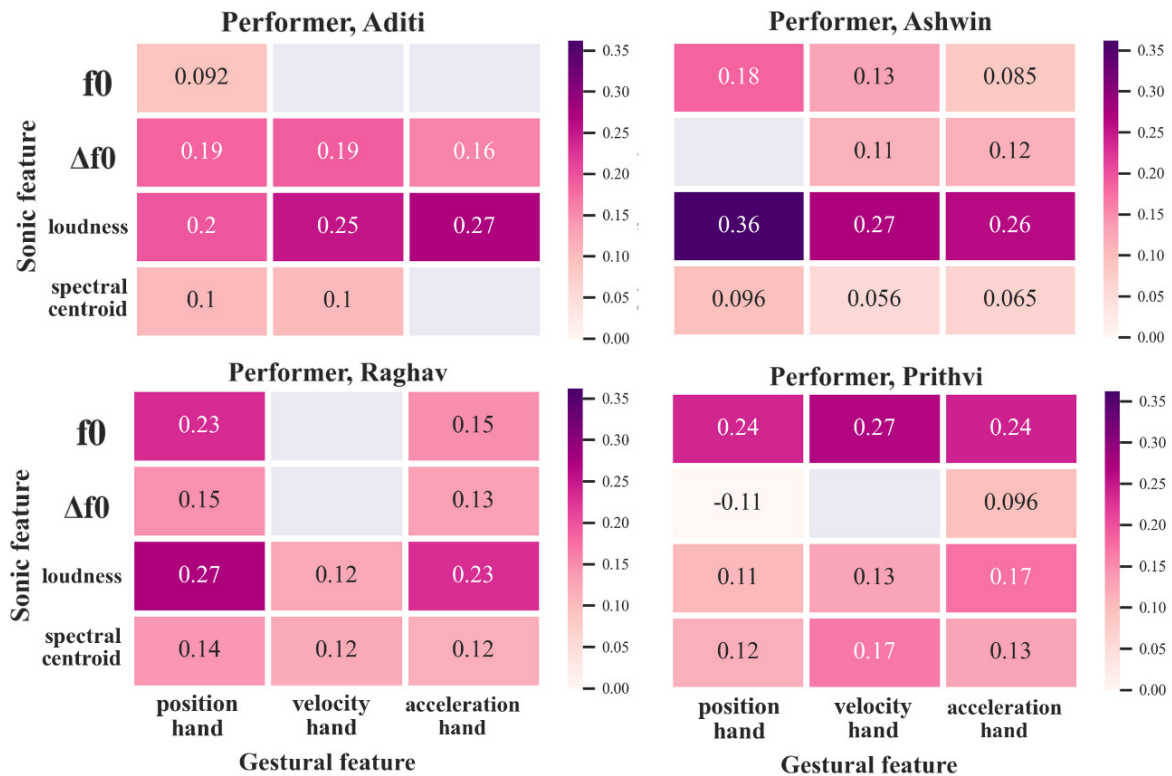


Figure 4. Spearman’s rank correlation coefficient for all performers and on an individual level. Insignificant test results are represented by a grey square.

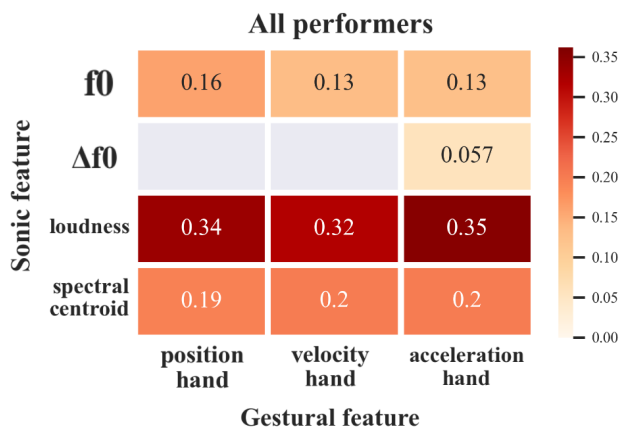


Figure 5. Spearman’s rank correlation coefficient for all performers in the dataset. Insignificant test results are represented by a grey square.

idiosyncratic singing technique. In Carnatic music, the violin usually replicates or closely follows the melody of the singing voice. The tānpūrā provides an ambient canvas, and the mṛdaṅgam, a pitched percussion instrument, is strongly present. Existing MSS models are unfamiliar with such music styles and struggle to give a clean, separated singing voice stem.

There have been some efforts by the community to improve MSS for the use case of Carnatic music [22], since much research is done on top of vocals. Therefore the availability of isolated vocal recordings is highly valuable

for the computational research of Carnatic music. Datasets like Saraga offer multi-track audio data, but given the fact that these recordings are from live concerts, there is leakage between sources that are part of the ensemble. The lack of clean multi-tracks for these music styles has been reported consistently, but Carnatic music is normally performed and recorded in a live setting. Therefore, recording musicians separately is not representative of how the tradition is generally performed. However, there has been some research on training MSS models with data with bleeding and some methods have been proposed to show the utility of data with bleeding [22].

Spleeter is a model based on a U-Net architecture that operates on the time-frequency domain. It is composed of a 6-layer encoder-decoder structure with skip-connections. Similar to most spectrogram-based separation models, Spleeter estimates n separation masks that are multiplied by the input mixture spectrogram to separate the sources. The official implementation of Spleeter provides a framework to fine-tune the available pre-trained models in order to adapt the system to a specific domain [43].

In an ideal case, clean Carnatic multi-track stems would be essential to fine-tune Spleeter. However, we utilize the data with leakage that is part of the Saraga Audiovisual dataset in an attempt to set up a baseline for bespoke Carnatic vocal separation. We use the provided 2-stem Spleeter model, trained on a private dataset of 25k samples of 30s. We fine-tune Spleeter using Saraga and Saraga Audiovisual, aiming also to study the effect of the newly collected multi-track data. The models are fine-tuned for

600k steps with a constant learning rate of $1e-5$. The fine-tuning process takes about a week in a TITAN XP GPU.

4.2.1 Experimental setup

Perceptual tests for MSS have gained interest in the research community, as objective metrics in [49] have been reported to not always correlate with the perceptual quality of MSS estimations [50]. Moreover, there is not a standardized and completely clean testing set for Carnatic separation. For that reason, we run a listening test with human subjects, including separations from recordings that we randomly collect from the Dunya dataset.

The listening test is based on the MUSHRA framework [51]. Subjects are asked to evaluate the vocal quality and the intrusiveness of other sources in separate stages. The scores are given on a scale from 1 to 5, with 5 being the maximum score. In each example, the subject is shown the original mixture as the reference stimuli, and the separations are shown unnamed and in a randomized order. The proposed subjective evaluation follows closely the ITU-T P.835 recommendation. We select and separate 6 Carnatic music concerts, ensuring diversity in audio quality and singer gender. Then, we randomly select a rendition from each concert, from which we collect a 30s chunk starting at a random point in time [22].

4.2.2 Results

We collect the results of the perceptual experiments and report the Mean Opinion Scores (MOS) per each model. We also report the Confidence Intervals (CIs) with $\alpha = 0.05$. A total of 20 subjects participated in the survey. Results are given in Table 3. While Spleeter samples are rated as having better vocal quality, Spleeter-FT-Sar improves over interference removal, while Spleeter-FT-SarAV is the most balanced solution among the three.

From the perceptual experiment, we conclude that Spleeter can better preserve the quality of the singing voice over the fine-tuned models. Using noisy data to fine-tune may be causing the model to lose some ability to properly discriminate the singing voice components. On the other hand, as the fine-tuned models improve on interference removal, we argue that it is possible for the pre-trained model to learn the instrumentation and vocal concepts of Carnatic music while preserving the knowledge to estimate separation masks for clean sources. In this particular experiment, Spleeter-FT-SarAV provides a balanced trade-off between artifacts and interferences. However, the overall performance is comparable to the other systems, suggesting that the multi-stem recordings have been obtained following the same peer-reviewed process in Saraga [7]. While establishing the baseline for Carnatic vocal separation, we also observe that leakage-aware systems such as [22] are still to be explored to take complete advantage of the multi-track data with leakage in both Saraga and Saraga Audiovisual, and outperform out of domain pre-trained models.

	Artifacts	Interferences
Spleeter [43]	3.89 _[3.75,4.04]	2.17 _[2.04,2.30]
Spleeter-FT-Sar	2.76 _[2.60,2.93]	3.80 _[3.68,3.93]
Spleeter-FT-SarAV	3.41 _[3.27,3.57]	2.88 _[2.74,3.02]

Table 3. MOS rating comparison between the default Spleeter [43] and a fine-tuned Spleeter using Saraga (FT-Sar) and Saraga Audiovisual (FT-SarAV). The higher, the better; 5 is the maximum rating. 95% CIs are also reported.

5. CONCLUSIONS

In this paper, we introduce Saraga Audiovisual, a multi-modal dataset for the analysis of Indian Art Music, specifically of the Carnatic style. The dataset includes multi-track audio, fundamental frequency extractions from that audio, performance videos, and human pose estimation extracted from the video footage. The dataset also includes metadata like rāga, tāla, composition, and structural annotations like the ālāpana, kalpanā svara, niraval, and thaṇi āvartana. The dataset is made available as a mirdata dataloader for easy and standardized access.

We perform two benchmarking experiments using the extracted audio and video features: (1) a multimodal analysis investigating the relationship between performer gesture and vocalisation, and (2) a fine-tuning experiment for Carnatic vocal source separation with audio data induced with leakage. Both experiments demonstrate the value of this data for music analysis in spite of imperfections in the automatically extracted feature data, such as audio leakage in the isolated instrument stems, or instability in the extracted pose estimations.

We expect Saraga Audiovisual to be a valuable resource for future work on tasks such as both vocal and instrument based leakage-aware source separation or predominant pitch extraction; and further multimodal studies of Carnatic music.

6. ETHICS STATEMENT

The Saraga Audiovisual dataset was recorded at the Arkay Convention Centre in Chennai. All of the artists appearing in the dataset gave informed consent for the dissemination of the data for research purposes through a consent form, and the Arkay Convention Centre was paid for their work in gathering the recordings.

With the release of this dataset, we wish to honour the cultural heritage of Carnatic music and safeguard its traditions. We recognize that to understand the distinctive intricacies of this culture and tradition, it is essential to go beyond computational methods alone. This understanding should not be oversimplified or broadly generalized.

7. ACKNOWLEDGEMENTS

This work was carried under the "IA y Música: Cátedra en Inteligencia Artificial y Música (TSI-100929-2023-1)",

funded by the Secretaría de Estado de Digitalización e Inteligencia Artificial, and the European Union-Next Generation EU, under the program Cátedras ENIA 2022 para la creación de cátedras universidad-empresa en IA. Special thanks to Suhit Chiruthapudi for actively helping with the preparation of the dataset. We would like to acknowledge Arkay Convention Centre, Chennai and all the musicians included in the dataset. Special thanks to Aaditya Rangan Raghavan, Samiksha Sreekanthan, Adithya Srinivasan and R Sarang for their invaluable contributions to the dataset. We would also like to extend our thanks to Dr. Lara Pearson for her valuable insights on the paper. We would like to thank Marius Rodrigues for helping with the pose estimation of the videos and Serafin Schweinitz for his contribution to the dataset. Finally, we would like to acknowledge the 20 participants who agreed to undertake the perceptual test.

8. REFERENCES

- [1] X. Serra, “A multicultural approach in music information research,” in *Proc. of the 12th Int. Society for Music Information Retrieval Conf. (ISMIR)*, Miami, USA, October 24-28 2011, pp. 151–156.
- [2] E. B. Maria Panteli and S. Dixon, “A review of manual and computational approaches for the study of world music corpora,” *Journal of New Music Research*, vol. 47, no. 2, pp. 176–189, 2018.
- [3] T. Nuttall, G. Plaja-Roglans, L. Pearson, and X. Serra, “The matrix profile for motif discovery in audio—an example application in carnatic music,” in *Int. Symposium on Computer Music Multidisciplinary Research*, Tokyo, Japan, 2021, pp. 228–237.
- [4] Nuttall, Thomas and Plaja-Roglans, Genís and Pearson, Lara and Serra, Xavier, “In search of sañcāras: tradition-informed repeated melodic pattern recognition in carnatic music,” in *Proc. of the 23rd Int. Society for Music Information Retrieval Conf. (ISMIR)*, Bengaluru, India, 2022, pp. 337–344.
- [5] T. Nuttall, X. Serra, and L. Pearson, “Svara-forms and coarticulation in carnatic music: an investigation using deep clustering,” in *Proc. of the 11th International Conference on Digital Libraries for Musicology (DLFM)*, Stellenbosch, South Africa, 2024, pp. 15–22.
- [6] S. Paschalidou and I. Miliaresi, “Multimodal deep learning architecture for Hindustani raga classification,” *Sensors and Transducers*, vol. 260, no. 2, pp. 77–86, 06 2023.
- [7] A. Srinivasamurthy, S. Gulati, R. C. Repetto, and X. Serra, “Saraga: Open datasets for research on indian art music,” *Empirical Musicology Review*, vol. 16, no. 1, pp. 85–98, 2021.
- [8] L. Pearson, “Cultural specificities in carnatic and hindustani music: Commentary on the saraga open dataset,” *Empirical Musicology Review*, vol. 16, no. 1, pp. 166–171, 2021.
- [9] P. Alonso-Jiménez, D. Bogdanov, J. Pons, and X. Serra, “Tensorflow audio models in essentia,” in *Proc. of the Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Barcelona, Spain, 2020, pp. 266–270.
- [10] G. Plaja-Roglans, T. Nuttall, L. Pearson, X. Serra, and M. Miron, “Repertoire-specific vocal pitch data generation for improved melodic analysis of carnatic music,” *Transactions of the Int. Society for Music Information Retrieval*, vol. 6, no. 1, pp. 13–26, 2023.
- [11] C. E. Cancino-Chacón and I. Pilkov, “The rach3 dataset: Towards data-driven analysis of piano performance rehearsal,” in *Int. Conf. on Multimedia Modelling*, Amsterdam, The Netherlands, 2024, pp. 28–41.
- [12] S. Nadkarni, S. Roychowdhury, P. Rao, and M. Clayton, “Exploring the correspondence of melodic contour with gesture in raga alap singing,” in *Proc. of the 24th Conf. of the Int. Society for Music Information Retrieval (ISMIR)*, Milano, Italy, 2023.
- [13] M. Clayton, P. Rao, N. N. Shikarpur, S. Roychowdhury, and J. Li, “Raga classification from vocal performances using multimodal analysis,” in *Proc. of the 23rd Int. Society for Music Information Retrieval (ISMIR)*, Bengaluru, India, 2022, pp. 283–290.
- [14] M. Rahaim, *Musicking bodies: Gesture and voice in Hindustani music*. Wesleyan University Press, 2013.
- [15] M. Clayton, J. Li, A. Clarke, and M. Weinzierl, “Hindustani raga and singer classification using 2d and 3d pose estimation from video recordings,” *Journal of New Music Research*, pp. 1–16, 2024.
- [16] G. A. Fatone, M. Clayton, L. Leante, and M. Rahaim, “Imagery, melody and gesture in cross-cultural perspective,” in *New perspectives on music and gesture*. Routledge, 2016, pp. 203–220.
- [17] L. Leante, “The lotus and the king: imagery, gesture and meaning in a hindustani rāg,” in *Ethnomusicology forum*, vol. 18, no. 2. Taylor & Francis, 2009, pp. 185–206.
- [18] M. Charulatha, “Gesture in musical declamation: An intercultural approach,” *Musicologist*, vol. 1, no. 1, pp. 6–31, 2017.
- [19] P.-S. Paschalidou, “Effort in gestural interactions with imaginary objects in hindustani dhrupad vocal music,” Ph.D. dissertation, Durham University, 2017.
- [20] S. Paschalidou, T. Eerola, and M. Clayton, “Voice and movement as predictors of gesture types and physical effort in virtual object interactions of classical indian singing,” in *Proc. of the 3rd Int. Symposium on Movement and Computing (SMC)*, Thessaloniki, Greece, 2016, pp. 1–2.

- [21] L. Pearson and W. Pouw, “Gesture–vocal coupling in karnatak music performance: A neuro–bodily distributed aesthetic entanglement,” *Annals of the New York Academy of Sciences*, vol. 1515, no. 1, pp. 219–236, 2022.
- [22] G. Plaja-Roglans, M. Miron, A. Shankar, and X. Serra, “Carnatic singing voice separation using cold diffusion on training data with bleeding,” in *24th Int. Society for Music Information Retrieval Conf. (ISMIR)*, Milano, Italy, 2023.
- [23] E. Clarke and N. Cook, Eds., *Empirical musicology: Aims, methods, prospects*. Oxford University Press, 2004.
- [24] T. Crawford and L. Gibson, Eds., *Modern methods for musicology: prospects, proposals, and realities*. Routledge, 2016.
- [25] M. Müller, *Information retrieval for music and motion*. Springer Berlin, Heidelberg, 2007.
- [26] D. Meredith, Ed., *Computational Music Analysis*. Springer Cham, 2016.
- [27] R. Caro Repetto and X. Serra, “Creating a corpus of Jingju (beijing opera) music and possibilities for melodic analysis,” in *Proc. of the 15th Conf. of the Int. Society for Music Information Retrieval (ISMIR)*, Taipei, Taiwan, 2014.
- [28] A. Srinivasamurthy, G. K. Koduri, S. Gulati, V. Ishwar, and X. Serra, “Corpora for music information research in indian art music,” in *Proc. of the Int. Computer Music Conf. (ICMC)*, Athens, Greece, 2014.
- [29] B. Uyar, H. S. Atli, S. Şentürk, B. Bozkurt, and X. Serra, “A corpus for computational research of Turkish Makam music,” in *Proc. of the 1st Int. Workshop on Digital Libraries for Musicology (DLFM)*, London, United Kingdom, 2014, pp. 1–7.
- [30] M. Sordo, A. Chaachoo, and X. Serra, “Creating Corpora for Computational Research in Arab-Andalusian Music,” in *Proc. of the 1st Int. Workshop on Digital Libraries for Musicology (DLFM)*, London, United Kingdom, 2014, p. 1–3.
- [31] R. C. Repetto, N. Pretto, A. Chaachoo, B. Bozkurt, and X. Serra, “An open corpus for the computational research of Arab-Andalusian music,” in *Proc. of the 5th Int. Conf. on Digital Libraries for Musicology (DLFM)*, Paris, France. New York, NY, USA: Association for Computing Machinery, 2018, p. 78–86. [Online]. Available: <https://doi.org/10.1145/3273024.3273025>
- [32] M. Clayton, S. Tarsitani, R. Jankowsky, L. Jure, L. Leante, R. Polak, A. Poole, M. Rocamora, P. Albornò, A. Camurri, T. Eerola, N. Jacoby, and K. Jakubowski, “The interpersonal entrainment in music performance data collection,” *Empirical Musicology Review*, vol. 16, no. 1, pp. 65–84, 2021.
- [33] F. C. Moss, M. Neuwirth, D. Harasim, and M. Rohrmeier, “Statistical characteristics of tonal harmony: A corpus study of beethoven’s string quartets,” *PLoS One*, vol. 14, no. 6, p. e0217242, 2019.
- [34] C. Weiß, M. Mauch, S. Dixon, and M. Müller, “Investigating style evolution of Western classical music: A computational approach,” *Musicae Scientiae*, vol. 23, no. 4, pp. 486–507, 2019.
- [35] H. Schreiber, C. Weiß, and M. Müller, “Local key estimation in classical music recordings: A cross-version study on schubert’s winterreise,” in *Proc. of the Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Barcelona, Spain, 2020, pp. 501–505.
- [36] M. Clayton, K. Jakubowski, and T. Eerola, “Interpersonal entrainment in indian instrumental music performance: Synchronization and movement coordination relate to tempo, dynamics, metrical and cadential structure,” *Musicae Scientiae*, vol. 23, no. 3, pp. 304–331, 2019.
- [37] L. Pearson, T. Nuttall, and W. Pouw, “Landscapes of coarticulation: The co-structuring of gesture-vocal dynamics in karnatak music performance,” 2024. [Online]. Available: osf.io/preprints/psyarxiv/npm96
- [38] R. M. Bittner, M. Fuentes, D. Rubinstein, A. Jansson, K. Choi, and T. Kell, “mirdata: Software for reproducible usage of datasets,” in *Int. Society for Music Information Retrieval Conf. (ISMIR)*, Delft, The Netherlands, 2019.
- [39] E. Clarke, “Meaning and the specification of motion in music,” *Musicae Scientiae*, vol. 5, no. 2, pp. 213–234, 2001.
- [40] R. I. Godøy, “Motor-mimetic music cognition,” *Leonardo*, vol. 36, no. 4, pp. 317–319, 2003. [Online]. Available: <http://www.jstor.org/stable/1577332>
- [41] Z. Eitan and R. Y. Granot, “How Music Moves: : Musical Parameters and Listeners Images of Motion,” *Music Perception*, vol. 23, no. 3, pp. 221–248, 02 2006.
- [42] M. Contributors, “Openmmlab pose estimation toolbox and benchmark,” <https://github.com/open-mmlab/mmpose>, 2020.
- [43] R. Hennequin, A. Khelif, F. Voituret, and M. Moussallam, “Spleeter: a fast and efficient music source separation tool with pre-trained models,” *Journal of Open Source Software*, pp. 1–4, 2020.
- [44] A. Défossez, N. Usunier, L. Bottou, and F. Bach, “Music source separation in the waveform domain,” 2019. [Online]. Available: <http://arxiv.org/abs/1911.13254>
- [45] Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimilakis, and R. Bittner, “MUSDB18-HQ - an uncompressed version of musdb18,” Dec. 2019. [Online]. Available: <https://doi.org/10.5281/zenodo.3338373>

- [46] I. Pereira, F. Araújo, F. Korzeniowski, and R. Vogl, “Moisesdb: A dataset for source separation beyond 4-stems,” in *Proc. of the 24th Int. Society for Music Information Retrieval Conf. (ISMIR)*, Milan, Italy, 2023.
- [47] N. N. Shikarpur, A. Keskar, and P. Rao, “Computational analysis of melodic mode switching in raga performance.” in *Proc. of the 22nd Int. Society for Music Information Retrieval Conf. (ISMIR)*, Online, 2021, pp. 657–664.
- [48] D. P. Shah, N. M. Jagtap, P. T. Talekar, and K. Gawande, “Raga recognition in Indian Classical Music using deep learning,” in *Artificial Intelligence in Music, Sound, Art and Design: 10th Int. Conf. (EvoMUSART)*, Sevilla, Spain, 2021, pp. 248–263.
- [49] F. R. Stöter, A. Liutkus, and N. Ito, “The 2018 Signal Separation Evaluation Campaign,” *Lecture Notes in Computer Science*, vol. 10891, pp. 293–305, 2018.
- [50] E. Cano, D. Fitzgerald, and K. Brandenburg, “Evaluation of quality of sound source separation algorithms: Human perception vs quantitative metrics,” in *Proc. of the 24th European Signal Processing Conf. (EUSIPCO)*, Budapest, Hungary, 2016, pp. 1758–1762.
- [51] M. Schoeffler, S. Bartoschek, F.-R. Stöter, M. Roess, S. Westphal, B. Edler, and J. Herre, “webMUSHRA — a comprehensive framework for web-based listening tests,” *Journal of Open Research Software*, vol. 6, no. 1, 2018.

X-COVER: BETTER MUSIC VERSION IDENTIFICATION SYSTEM BY INTEGRATING PRETRAINED ASR MODEL

Xingjian Du¹ Mingyu Liu¹ Pei Zou¹
Xia Liang¹ Zijie Wang¹ Huidong Liang² Bilei Zhu¹
¹ ByteDance Inc.
² Univeristy of Oxford
xingjian.du97@gmail.com

ABSTRACT

Methods based on deep learning have emerged as a dominant approach for cover song identification (CSI) literature over the past years, among which ByteCover systems have consistently delivered state-of-the-art performance across major CSI datasets in the field. Despite its steady improvements along previous generations from audio feature dimensionality reduction to short query identification, the system is found to be vulnerable to audios with noise and ambiguous melody when extracting musical information from constant-Q transformation (CQT) spectrograms. Although some recent studies suggest that incorporating lyric-related features can enhance the overall performance of CSI systems, this approach typically requires training a separate automatic lyric recognition (ALR) model to extract lyric-related features from music recordings. In this work, we introduce X-Cover, the latest CSI system that incorporates a pre-trained automatic speech recognition (ASR) module, Whisper, to extract and integrate lyrics-related features into modelling. Specifically, we jointly fine-tune the ASR block and the previous ByteCover3 system in a parameter-efficient fashion, which largely reduces the cost of using lyric information compared to training a new ALR model from scratch. In addition, a bag of tricks is further applied to the training of this new generation, assisting X-Cover to achieve strong performance across various datasets.

1. INTRODUCTION

In the rapidly evolving field of music information retrieval, Cover Song Identification (CSI), which aims to identify different versions of a specific musical composition within a large database, remains a complex and computationally challenging task [1, 2]. This problem has received consid-

erable interest for its wide-ranging applications such as intellectual property management and enhancing music recommendation systems [3, 4].

With the advancements in deep learning, CSI systems based on neural networks gradually replace traditional models based on handcrafted features [5, 6] and become a new paradigm for real-world deployment. Existing methods typically frame CSI as either a classification problem [7–9], a metric learning problem [10], or a combination of both [11, 12]. On the other hand, the proliferation of social video platforms like TikTok has also led to a surge in short-form videos, which often contain remixed or covered segments of original compositions and hence involve copyright infringement issues. Unfortunately, as most of the existing works above contain a global pooling layer to directly aggregate the information from all time sections, they are found to suffer from identifying these seconds-long short segments.

To address this problem, the latest ByteCover3 system [13] first splits each full audio track into a set of short spectrogram chunks and then uses a neural-network-based extractor (i.e. the ResNet-IBN module introduced in the first ByteCover generation [11]) to encode them into latent embeddings. These low-dimensional embeddings are later sent to calculate a Local Alignment Loss (LAL) that uses the matching of local embeddings to identify short queries against full songs.

Despite the progress in detecting short cover songs, the ByteCover3 system is still found to be vulnerable to non-musical information in real-world scenarios. For instance, musical segments in short videos are frequently overlaid with ambient noise, speech, or poorly composed user-generated melodies. The presence of these non-musical elements that mask or distort the musical elements can mislead the feature extraction phase of ByteCover3, leading to inaccurate or ambiguous representations of the audio content. This misrepresentation can degrade the system’s ability to correctly match the audio sample against its database of known songs, reducing both the accuracy and reliability of the system in operational settings.

To enhance the robustness of existing CSI systems, an intuitive approach is to incorporate more discriminative information into model training. Recent research [14, 15] has demonstrated that lyric-related features are less susceptible to being masked by unrelated noise sources and can serve



© X. Du, M. Liu, P. Zou, X. Liang, Z. Wang, H. Liang, and B. Zhu. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** X. Du, M. Liu, P. Zou, X. Liang, Z. Wang, H. Liang, and B. Zhu, “X-Cover: Better Music Version Identification system by integrating pretrained ASR model”, in *Proc. of the 25th Int. Society for Music Information Retrieval Conf.*, San Francisco, United States, 2024.

as auxiliary inputs to bolster CSI systems. For example, one recent work [14] utilizes latent embeddings from an automatic lyrics recognition model (ALR) as lyric-related features for CSI. However, this approach requires training an extra ALR module from the beginning and employs Dali [16] as its training dataset, which implies that it only supports the recognition of English lyrics.

In this paper, we extend the current ByteCover3 system by leveraging a pre-trained Automatic Speech Recognition (ASR) model to jointly model both lyric and musical features for short-query identification, which eliminates the need for training an additional ALR model from scratch. Specifically, we use Whisper [17] as the pre-trained ASR module for its inherent scalability and robust recognition capabilities across multiple languages. However, the training is still quite challenging in terms of GPU memory consumption and inference time given the large parameter size of Whisper and its autoregressive nature at the text decoding stage. To counter these issues, we employ a prefix-tuning fashion that adapts the output of Whisper to CSI training. A trainable prefix latent is added before each text decoder block in Whisper to reprogram the model to extract features specific to CSI task without extensive retraining of the entire model. Finally, the fusion model is trained using the local alignment loss (LAL) scheme introduced in ByteCover3, and together with a bag of new techniques, we further improve the current ByteCover system to be more efficient and accurate in CSI tasks with controllable training GPU memory and reasonable inference time.

2. PRELIMINARIES

This paper builds on ByteCover series [11–13] training framework and model structure. The primary motivation of ByteCover series is to develop a highly accurate, robust and efficient cover song detection system for real-world industrial-level tasks with various query types and large music corpora, beyond typical laboratory settings.

ByteCover1 [11] introduced a streamlined framework designed to train a neural network that extracts version-related embeddings from the CQT spectrogram of input audio recordings. This model utilized Instance Batch Normalization (IBN) layer [18] within its ResNet architecture, which enhances the model’s capability to learn invariant features while preserving discrimination and is critical for handling diverse musical styles. Additionally, a Generalized Mean (GeM) pooling layer was employed to compress local features into a global feature, optimizing the model’s training objectives. Furthermore, ByteCover1 adopted a multi-loss training paradigm that combined classification loss and triplet loss, fostering a more robust representation and improved accuracy.

For improved throughput, the authors of ByteCover2 [12] identified an anisotropy in the embedding distribution of ByteCover1, which led to inefficient utilization of the embedding dimension size. To address this issue, ByteCover2 introduced a Principal Component Analysis - Fully Connected (PCA-FC) layer. The weights of this layer are informed by the transformation matrix de-

rived from a PCA analysis of the original ByteCover1 embeddings. This strategic adjustment effectively alleviated the anisotropy problem, enabling ByteCover2 to match the performance of ByteCover1 while only requiring one-eighth of the dimension size. Consequently, this reduction drastically accelerated the retrieval of query embeddings and linearly decreased the storage costs of the embedding database relative to the magnitude of the dimension size reduction.

With the emergence of short video content, more queries in CSI systems appear to be short audio clips. However, the authors of ByteCover3 [13] observed that state-of-the-art CSI methods performed suboptimally on these short queries, where the accuracy of previous deep learning CSI models [10–12, 19] degraded significantly as the duration of query audio decreased. This issue was linked to the global pooling modules employed in previous works that often neglected local features, which complicates the task of matching segments of songs to complete tracks. Traditional audio matching algorithms, such as the "shingling" method referenced in [20], slice inputs into segments and extract features separately to preserve local details. Unfortunately, this straightforward strategy struggled when applied to deep learning-based CSI methods, which are inherently data-driven rather than handcrafted. Consequently, there is a misalignment between training and inference objectives: while softmax and triplet losses focus on matching pairs of embedding vectors, the model deals with sequences of local embeddings at the inference phase.

To bridge this gap, ByteCover3 introduced a novel training paradigm known as Local-Aware Losses (LAL) for metric learning on sequence data. This approach extends softmax and triplet losses into a more general form that directly optimizes the metrics between two sequences of embeddings. Specifically, ByteCover3 employs the MaxMean Measure to assess the similarity between two sequences of vectors, consequently ensuring better alignment of the training and inference targets and enhancing performance. This measure calculates the similarity between each segment in the query audio and the most similar segment in the candidate song, averaging these similarities to produce a final similarity score. This method is computationally efficient and differentiable. With the use of LAL and the MaxMean measure, ByteCover3 demonstrated significant improvements in retrieval capabilities with 30-second queries.

The subsequent method section, Section 3, describes the details of X-Cover and is organized as follows:

- Subsection 3.1 introduces the overall framework of X-Cover, which retains a high-level similarity to ByteCover3.
- Subsection 3.2 discusses the incorporation of the pretrained ASR model, Whisper [17], to enhance the robustness of the CSI system. It also describes a non-autoregressive decoding method designed to accelerate Whisper’s decoding. Compared to previous methods that relied solely on cover song data,

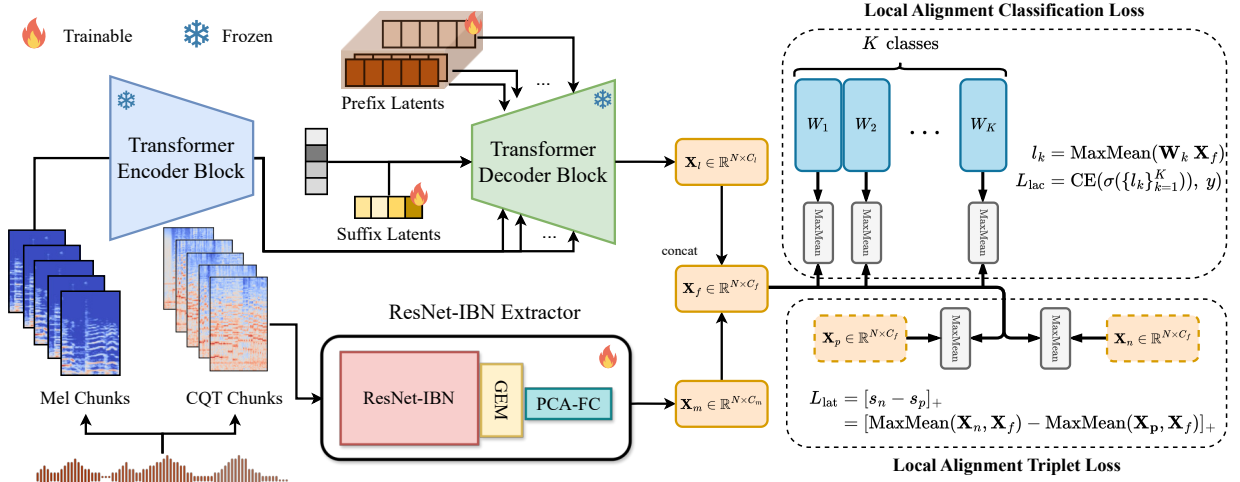


Figure 1. The input CQT spectrogram and Mel spectrogram are split into n equal-sized chunks with overlap in temporal dimension. Subsequently, the ResNet-IBN extractor derived from [13] generates an embedding \mathbf{X}_m that contains N local features corresponding to N input chunks with the CQT spectrogram. The optimized Whisper variant generates an embedding \mathbf{X}_l that contains N local features corresponding to N input chunks with the log Mel spectrogram. Then, we optimize the model with a multi-loss objective that consists of a classification loss L_{lac} and a triplet loss L_{lat} using the *MaxMean* measure, where \mathbf{X}_p and \mathbf{X}_n represent the positive sample and negative sample in triplet loss.

X-Cover achieved superior performance without the necessity of pretraining on additional lyric data. This simplification and streamlining of the training process significantly enhance the efficiency of model preparation.

- Subsection 3.3 introduces a bag of tricks to improve the performance of the CSI model, covering aspects such as data augmentation, model architecture, and loss function design details.

3. X-Cover

In this section, we delineate the architecture and training paradigms of X-Cover, emphasizing its novelties: efficient adaption of Whisper and a bag of tricks for improving ByteCover3 in Cover Song Identification (CSI) tasks. The overall architecture is depicted in Fig. 1.

3.1 Overall Framework of X-Cover

The architecture of X-Cover is an evolutionary extension of ByteCover3, delineated in Section 3. X-Cover retains the multi-objective learning paradigm and the ResNet-Based Feature Extractor from ByteCover3. To incorporate lyric-related features, X-Cover introduces a pretrained ASR model, Whisper, alongside the existing ResNet-IBN architecture [11].

In ByteCover3, local features are extracted by initially resampling the audio to 22,050 Hz and partitioning it into N overlapping segments of 20 seconds each, with a 10-second hop. These segments are subsequently transformed into CQT spectrograms, serving as the input to the ResNet-IBN model. The model outputs a 4-D embedding, which

undergoes GeM pooling to yield a compact final local embedding $\mathbf{X}_m \in \mathbb{R}^{N \times C_m}$, comprising N local features.

To facilitate the integration of Whisper as an additional feature extractor, mel spectrograms are extracted from the input audio. To temporally align the features from both branches, the same chunking strategy as in ByteCover3 is employed. Post chunking, the specialized Whisper model for CSI utilizes these mel spectrograms to extract lyric-related embeddings $\mathbf{X}_l \in \mathbb{R}^{N \times C_l}$. Upon obtaining the two sets of latent embeddings, \mathbf{X}_m and \mathbf{X}_l , a straightforward feature-dimensional concatenation is performed to generate the fused embedding $\mathbf{X}_f \in \mathbb{R}^{N \times (C_m + C_l)}$, for the simplicity, we redefine it as $\mathbf{X} \in \mathbb{R}^{N \times C}$ where $C = C_f = C_m + C_l$. This is feasible due to the temporal alignment and identical lengths of \mathbf{X}_m and \mathbf{X}_l . X-Cover leverages the Local Alignment Loss (LAL) methodology, originally proposed in ByteCover3, to enhance local segments matching capabilities. The LAL comprises a classification loss \mathcal{L}_{lac} and a triplet loss \mathcal{L}_{lat} , defined as follows:

$$\text{logit}_k = \text{MaxMean}(\mathbf{X}, \mathbf{W}_k), \quad (1)$$

$$\mathcal{L}_{lac} = \text{CE}(\sigma(\{\text{logit}_k\}_{k=1}^K), y), \quad (2)$$

$$\mathcal{L}_{lat} = [\text{MaxMean}(\mathbf{X}_n, \mathbf{X}) - \text{MaxMean}(\mathbf{X}_p, \mathbf{X})]_+, \quad (3)$$

where $\text{CE}(\cdot, \cdot)$ is the cross entropy, $\sigma(\cdot)$ is the softmax function, $\mathbf{X}_n, \mathbf{X}_p \in \mathbb{R}^{N \times C}$ is the fused embedding of the negative sample and positive sample while calculating triplet loss. $\mathbf{W} \in \mathbb{R}^{K \times L \times C}$ is a trainable weight matrix in the linear layer before softmax, and $\mathbf{W}_k \in \mathbb{R}^{L \times C}$ denotes the proxy representation for class k . L is a hyperparameter which is set to 9 in ByteCover3 and X-Cover. These loss functions serve as a robust optimization objective, fortify-

ing the model’s performance and adaptability for emerging challenges in music information retrieval. The subsequent subsections will expound upon these enhancements and their contributions to the overarching efficacy of X-Cover.

3.2 Efficient Adaption of Whisper for CSI

Whisper [17] serves as a state-of-the-art ASR framework, exhibiting robust scalability through its Transformer-based encoder-decoder architecture. Its efficacy in ASR tasks has been corroborated by numerous studies [21, 22]. The encoder processes normalized spectrograms via an initial stem, consisting of two convolutional layers, before routing the output through multiple Transformer blocks employing pre-activation residual connections.

On the decoder end, learned positional embeddings are integrated with tied input-output token representations to generate the final transcript. To maintain architectural coherence, the encoder and decoder are structured to have an identical number of Transformer blocks and the same width.

In the Whisper decoder, an autoregressive approach is adopted, similar to the GPT series of language models. The probability each token is sequentially determined based on the preceding context. Nonetheless, extensive parameter count of Whisper renders it challenging for downstream applications. Conventional fine-tuning strategies are computationally expensive, leading us to adopt prefix-tuning [23], a more efficient alternative that utilizes a smaller set of trainable parameters.

Motivated by these advances, we introduce an optimized Whisper variant for lyric-based feature extraction from audio recordings. The detailed structure of it is shown in Figure 2. The audio encoder ingests chunked log mel spectrograms $\mathbf{S}_{mel} \in \mathbb{R}^{N \times F \times T}$ and comprises blocks with self-attention and MLP layers. Residual connections are employed in both layers, culminating in audio features $\mathbf{X}_{ac} \in \mathbb{R}^{N \times C_{ac}}$.

For every segment, the text decoder process the corresponding audio feature $\mathbf{X}_{ac}(i) \in \mathbb{R}^{C_{ac}}$ independently for transcribing. The text decoder starts with four initial tokens $\mathbf{t}_{init} \in \mathbb{R}^4$, which serve specific purposes such as indicating the start of prediction, speech presence, task specification, and timestamp prediction. These tokens pass through an embedding layer to yield embedded tokens $\mathbf{E}_{init} \in \mathbb{R}^{4 \times C_l}$, defined during the pretrain phase. A trainable suffix latent $\mathbf{E}_e \in \mathbb{R}^{L_e \times C_l}$ is appended to these initial tokens. Additionally, before each text decoder block, a trainable prefix latent $\mathbf{E}_{p_j} \in \mathbb{R}^{L_p \times C_l}$ is added, forming the input for the first text decoder block as $\mathbf{E}_{in_1} = [\mathbf{E}_{p_1}; \mathbf{E}_{init}; \mathbf{E}_e] \in \mathbb{R}^{(L_p+4+L_e) \times C_l}$. The prefix embeddings are learned to overwrite the instruction carried by init tokens, to reprogram Whisper to extract features that assist in the CSI task. Compared with the audio encoder block, the text decoder block incorporates an additional cross-attention layer. The decoder block consists of self-attention, cross-attention, and MLP layers, sequentially connected. The self-attention layer accepts the out-

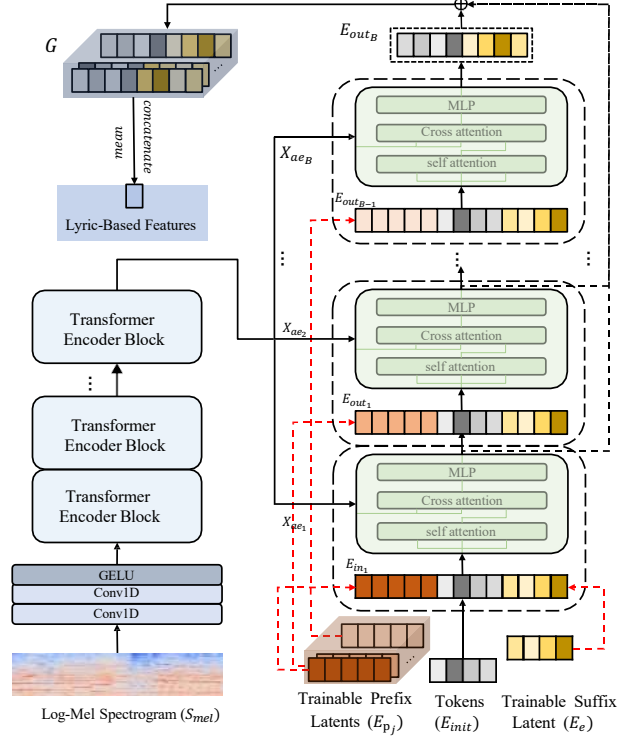


Figure 2. The detailed structure of modified Whisper Model.

put of the preceding text decoder block, while the cross-attention layer uses both the audio encoder output $\mathbf{X}_{ac}(i)$ as its key and value and the output of self-attention layer as its query. The MLP layer subsequently processes the cross-attention output. After processing through the first text decoder block and subsequent truncation, we obtain $\mathbf{E}_{out_1} \in \mathbb{R}^{(4+L_e) \times C_l}$. The same process is repeated for subsequent text decoder blocks, each time adding a new set of trainable prefix latents. Since the first L_p output embeddings of each text decoder block are truncated, the input dimensions of each decoder block remain consistent even after adding a new set of prefix latents. The outputs of each text decoder block are stored to form an lyric-based feature group $\mathbf{G} = [\mathbf{E}_{out_1}, \mathbf{E}_{out_2}, \dots, \mathbf{E}_{out_B}] \in \mathbb{R}^{((4+L_e) \times B) \times C_l}$. Finally, We select the last 70% to 80% of the audio-lyric feature group and the mean across the second dimension is taken to get the audio-lyric feature $\mathbf{X}_l(i) \in \mathbb{R}^{C_l}$. The medium version of the whisper we adopted contains 24 text decoder blocks, and the hidden size is 1024.

In terms of computational efficiency, the introduction of prefix latents enables the Whisper model to generate multiple embeddings in a single forward pass, thereby obviating the need for autoregressive operations during the decoding phase. This can be regarded as a Non-Autoregressive (NAR) method for efficient inference [24]. Despite this, certain studies [24] indicate that NAR performance is generally inferior to that of Autoregressive (AR) methods. To explore the performance upper bound of Whisper in CSI tasks, we also propose an AR-based feature extraction method. Importantly, in this AR approach, there are no modifications required to the Whisper model, and its

parameters are kept frozen throughout the process. The Whisper model transcribes each audio segment and collects the final hidden states from each inference pass. These states are then aggregated using a mean operation to form the lyric-based feature $\mathbf{X}_l(i)$. Since the shape of the embeddings outputted by both NAR and AR methods is consistent, they can be used interchangeably. In section 4, we refer to the strategy using efficient Whisper adaptation as "E" and the original Whisper-based strategy as "AR". Lastly, we introduce a trainable linear projection layer subsequent to the Whisper output to map the embedding dimensions C_l to 512. For clarity, the symbol C_l continues to denote the dimensions of these projected embeddings.

3.3 A Bag of Tricks for Improving ByteCover3

We enhance our model’s performance by incorporating three techniques: Sparse Softmax, Non-local Operations, and Grid Distortion. Due to the crowdsourced nature of the labels in the SHS100k dataset [9], mislabeling of version categories is inevitable. Additionally, the SHS100k dataset contains 8,858 cliques of music, indicating a high number of classes to distinguish when training models using softmax loss. Sun et al. [25] demonstrated that an increase in the number of classes raises the risk of overfitting with softmax loss, particularly when incorrect labels are present, as it may further degrade model performance by fitting these erroneous labels. Therefore, in X-Cover, we have opted to use Sparse Softmax [25] loss instead of the original softmax loss. Sparse Softmax loss retains only the top-K logits and the logits for the ground truth class during probability computation, effectively reducing the number of classes to distinguish to $K + 1$. However, a smaller K value during the initial phase of training might impair the model’s ability to fit. Through tuning, we found that fine-tuning with $K = 1024$ after training with the original softmax loss achieves optimal results.

As demonstrated in [26], capturing global spectrogram information is crucial for understanding the complex compositions in music, characterized by varied spectral characteristics over time. However, the limited receptive field size of conventional CNNs restricts their ability to capture long-range dependencies across different parts of the spectrogram effectively. To address this limitation, we integrate Non-Local modules [27] into the ResNet-IBN architecture. These modules utilize the strength of Non-Local operations to compute interactions directly between any two positions in the input data, irrespective of their physical distance. This feature is particularly advantageous for analyzing music tracks, where distant sections may share thematic but transformed material, a characteristic common in cover songs.

Finally, Grid Distortion is borrowed from computer vision field, involves random scaling transformations in both the frequency and time dimensions of the spectrogram to simulate time-stretching and pitch shift.

4. EXPERIMENTS

We evaluated X-Cover using two publicly available datasets: SHS100K, which consists of 8,858 cover groups and 108,523 individual recordings [9], and Covers80, featuring 160 recordings that include two covers of each of the 80 songs [5]. The training and test division of SHS100K adheres to previous work [8, 11–13], while Covers80 serves exclusively for testing. We convert all audio to CQT and Mel spectrograms before training. For CQT, we set the bins per octave to 12 and use a Hann window during extraction with a hop size of 512. All audio is resampled to 22,050 Hz before CQT conversion. We then downsample the CQT temporally by averaging over 100 adjacent frames to enhance computational efficiency and reduce latency. For Mel spectrograms, we follow the configuration in Whisper [17], resampling audio to 16,000 Hz and computing an 80-channel log-magnitude Mel spectrogram with 25-millisecond windows and a 10-millisecond stride. X-Cover’s training phase uses weights from ByteCover3 for initialization. For the Whisper branch, we adopt configurations from the Whisper-Medium setup [17], and its initial weights are also sourced from pre-trained models.

Model	#Dims. ↓	mAP ↑	MR1 ↓
Covers80			
Me+Ha+Ly [14]	1536	0.993	1.02
ByteCover3 [13]	512	0.927	3.32
X-Cover-E	2560	0.992	1.04
X-Cover-AR	2560	1.000	1.00
SHS100K-TEST			
MOVE [10]	512	0.519	154.5
Me+Ha+Ly [14]	1536	0.794	39.3
ByteCover3 [13]	512	0.824	37.0
Whisper-E	512	0.437	150
Whisper-AR	512	0.708	145.4
ByteCover3.5	2048	0.857	22.7
X-Cover-E	2560	0.889	14.9
X-Cover-AR + PCA-FC [12]	512	0.924	14.7
X-Cover-AR	2560	0.924	14.9

Table 1. Performance on different datasets.

4.1 Comparison on Performance and Efficiency

As shown in Table 1, we benchmark the performance of various models, including our X-Cover, on two datasets: Covers80 and SHS100K-TEST. Metrics reported include the number of dimensions (Dims.), mean Average Precision (mAP), and Mean Rank 1 (MR1). For comparative analysis, we include MOVE [10], Me+Ha+Ly [14], and ByteCover3 [13]. Our X-Cover incorporate Whisper models trained with LAL loss, with and without efficient adaptation. By integrating these Whisper-based models with ByteCover3.5, we produce two hybrid solutions: X-Cover-E and X-Cover-AR. Notably ByteCover3.5 is an improved version over ByteCover3 which incorporates a series of enhancements described in Section 3.3. Addition-

	ByteCover3.5	Whisper-E	Whisper-AR	X-Cover-E	X-Cover-AR
Feature Extraction(ms)	21	402	10467	452	10522

Table 2. The average inference time per audio

ally, the Me+Ha+Ly model [14] included in the comparison is not a strict replication of the original work which is a composite system comprised with a melody extraction model, the ALR model, and the MOVE [10]. Due to limited information in the original paper and the smaller size of ALR model compared to Whisper, our version of Me+Ha+Ly uses Basic-Pitch [28] and our Whisper-AR, ensuring a more equitable comparison. We will release the implementation publicly. All models are trained with the Adam Optimizer and a batch size of 128. Table 2 presents the average inference time for X-Cover variants on SHS100K-TEST using an NVIDIA A100 GPU.

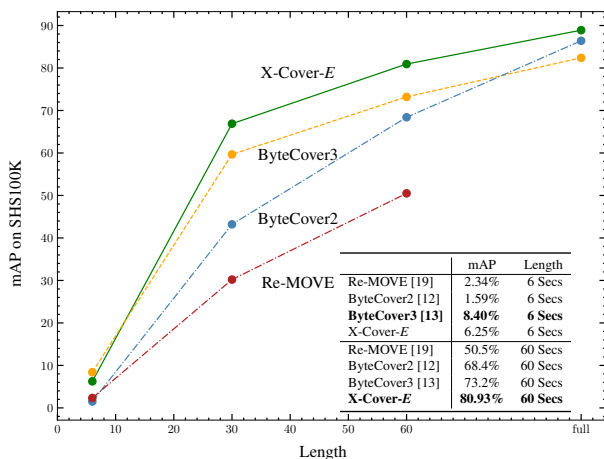

Figure 3. Length of Queries vs. Performance.

Figure 3 displays the mAP results of X-Cover-E for different query lengths on SHS100K-TEST, using Re-MOVE [19], ByteCover2 [12] and ByteCover3 [13] as compared methods. As illustrated in the figure, our X-Cover-E model achieves the best mAPs for all query lengths except for the 6-second length. This clearly indicates the effectiveness of X-Cover-E. The poorer performance at the 6-second scenario is likely due to the shortness of the queries, which may not contain enough meaningful lyric information, thus not highlighting the strengths of X-Cover-E.

Overall, X-Cover-AR consistently outperforms other models on both datasets. Remarkably, X-Cover-AR, X-Cover-E and Me+Ha+Ly achieve nearly 100% accuracy on Covers80. Both methods employ Whisper-AR, making this high accuracy expected given Covers80 which is dataset with a limited size and predominantly consisting of recordings with vocal components. Therefore, we believe SHS100K-TEST is a more robust test of model performance. On this dataset, both X-Cover-AR and X-Cover-E achieve state-of-the-art performance. The performance gap between X-Cover-E and X-Cover-AR can be attributed to the differences in Whisper-AR and Whisper-E. The com-

parable performance and significantly faster speed of X-Cover-E validate our efficient adaptation. Surprisingly, Whisper-AR, cloned from an ASR model except for the final linear layer, shows comparable performance to other SoTA methods on SHS100K, highlighting the potential of large-scale pretrained ASR models in CSI tasks. However, the second-level inference time for individual samples in X-Cover-AR poses a challenge for its practical deployment in real-life scenarios. Finally, we employ the PCA-FC dimensionality reduction module from ByteCover2 to compress X-Cover-AR features from 2560 to 512 dimensions, finding negligible performance loss. This suggests that the performance gains in X-Cover variants are not due to increased feature size.

5. CONCLUSION

This paper has enhanced the robustness and efficiency of the ByteCover3 system in CSI by integrating lyric-related features using a pre-trained Automatic Speech Recognition model, Whisper. This integration addresses the issue of non-musical elements that distort musical characteristics essential for accurate CSI, without the need for training an additional lyrics recognition module from scratch.

The use of Whisper, adapted via prefix-tuning, significantly reduces the computational demands typically associated with large-scale ASR systems, thereby improving efficiency in both training and inference stages. Our results demonstrate improved accuracy and reliability of CSI, particularly in handling short queries against full songs.

In conclusion, our approach contributes to the development of more robust, efficient, and scalable CSI systems, enhancing both intellectual property management and music recommendation systems, especially in social video platforms. Future work will aim to further optimize these methods and explore additional features to increase resilience against noise in practical applications.

6. REFERENCES

- [1] M. Müller, F. Kurth, and M. Clausen, “Audio matching via chroma-based statistical features.” in *ISMIR*, vol. 2005. Citeseer, 2005, p. 6.
- [2] F. Yesiler, G. Doras, R. M. Bittner, C. J. Tralie, and J. Serrà, “Audio-based musical version identification: Elements and challenges,” *IEEE Signal Processing Magazine*, vol. 38, no. 6, pp. 115–136, 2021.
- [3] M. F. Yesiler *et al.*, “Data-driven musical version identification: accuracy, scalability and bias perspectives,” 2022.

- [4] J. Serra, “Identification of versions of the same musical composition by processing audio descriptions,” *Department of Information and Communication Technologies*, 2011.
- [5] D. P. Ellis and G. E. Poliner, “Identifying cover songs with chroma features and dynamic programming beat tracking,” in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2007.
- [6] J. Serra, E. Gómez, P. Herrera, and X. Serra, “Chroma binary similarity and local alignment applied to cover song identification,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 6, pp. 1138–1151, 2008.
- [7] Z. Yu, X. Xu, X. Chen, and D. Yang, “Temporal pyramid pooling convolutional neural network for cover song identification,” in *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 2019, pp. 4846–4852.
- [8] Z. Yu, X. Xu, X. Chen, and D. Yang, “Learning a representation for cover song identification using convolutional neural network,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 541–545.
- [9] X. Xu, X. Chen, and D. Yang, “Key-invariant convolutional neural network toward efficient cover song identification,” in *IEEE International Conference on Multimedia and Expo*, 2018, pp. 1–6.
- [10] F. Yesiler, J. Serrà, and E. Gómez, “Accurate and scalable version identification using musically-motivated embeddings,” in *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 21–25.
- [11] X. Du, Z. Yu, B. Zhu, X. Chen, and Z. Ma, “Bytecover: Cover song identification via multi-loss training,” in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 551–555.
- [12] X. Du, K. Chen, Z. Wang, B. Zhu, and Z. Ma, “Bytecover2: Towards dimensionality reduction of latent embedding for efficient cover song identification,” in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 616–620.
- [13] X. Du, Z. Wang, X. Liang, H. Liang, B. Zhu, and Z. Ma, “Bytecover3: Accurate cover song identification on short queries,” in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.
- [14] M. Abrassart and G. Doras, “And what if two musical versions don’t share melody, harmony, rhythm, or lyrics?” in *ISMIR 2022*, 2022.
- [15] A. Vaglio, R. Hennequin, M. Moussallam, and G. Richard, “The words remain the same: Cover detection with lyrics transcription,” in *22nd International Society for Music Information Retrieval Conference ISMIR 2021*, 2021.
- [16] G. Meseguer-Brocal, A. Cohen-Hadria, and G. Peeters, “Dali: A large dataset of synchronized audio, lyrics and notes, automatically created using teacher-student machine learning paradigm,” *arXiv preprint arXiv:1906.10606*, 2019.
- [17] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, “Robust speech recognition via large-scale weak supervision,” in *International Conference on Machine Learning*. PMLR, 2023, pp. 28 492–28 518.
- [18] X. Pan, P. Luo, J. Shi, and X. Tang, “Two at once: Enhancing learning and generalization capacities via ibn-net,” in *Proceedings of the european conference on computer vision (ECCV)*, 2018, pp. 464–479.
- [19] F. Yesiler, J. Serrà, and E. Gómez, “Less is more: Faster and better music version identification with embedding distillation,” in *Proc. of the Int. Soc. for Music Information Retrieval Conf. (ISMIR)*, 2020.
- [20] F. Zalkow, J. Brandner, and M. Müller, “Efficient retrieval of music recordings using graph-based index structures,” *Signals*, vol. 2, no. 2, pp. 336–352, 2021.
- [21] K. Li, Z. Liu, T. He, H. Huang, F. Peng, D. Povey, and S. Khudanpur, “An empirical study of transformer-based neural language model adaptation,” in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.
- [22] Q. Yue, W. Shi, Y. He, J. Chu, Z. Han, and X. Han, “An improved speech recognition system based on transformer language model,” in *2021 International Conference on Control, Automation and Information Sciences (ICCAIS)*, 2021.
- [23] X. Liu, K. Ji, Y. Fu, W. L. Tam, Z. Du, Z. Yang, and J. Tang, “P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks,” *arXiv preprint arXiv:2110.07602*, 2021.
- [24] Y. Xiao, L. Wu, J. Guo, J. Li, M. Zhang, T. Qin, and T. Liu, “A survey on non-autoregressive generation for neural machine translation and beyond,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.
- [25] S. Sun, Z. Zhang, B. Huang, P. Lei, J. Su, S. Pan, and J. Cao, “Sparse-softmax: A simpler and faster alternative softmax transformation,” *arXiv preprint arXiv:2112.12433*, 2021.
- [26] X. Du, B. Zhu, Q. Kong, and Z. Ma, “Singing melody extraction from polyphonic music based on spectral

correlation modeling,” in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 241–245.

- [27] X. Wang, R. Girshick, A. Gupta, and K. He, “Non-local neural networks,” *CVPR*, 2018.
- [28] R. M. Bittner, J. J. Bosch, D. Rubinstein, G. Meseguer-Brocal, and S. Ewert, “A lightweight instrument-agnostic model for polyphonic note transcription and multipitch estimation,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Singapore, 2022.

HARMONIC AND TRANSPOSITION CONSTRAINTS ARISING FROM THE USE OF THE ROLAND TR-808 BASS DRUM

Emmanuel Deruty

Sony Computer Science Laboratories, Paris, France; Aalborg University, Denmark
emmanuel.deruty@sony.com

ABSTRACT

The study investigates hip-hop music producer Scott Storch’s approach to tonality, where the song’s key is transposed to fit the Roland TR-808 bass drum instead of tuning the drums to the song’s key. This process, involving the adjustment of all tracks except the bass drum, suggests significant production motives. The primary constraint stems from the limited usable pitch range of the TR-808 bass drum if its characteristic sound is to be preserved. The research examines drum tuning practices, the role of the Roland TR-808 in music, and the sub-bass qualities of its bass drum. Analysis of TR-808 samples reveals their characteristics and their integration into modern genres like trap and hip-hop. The study also considers the impact of loudspeaker frequency response and human ear sensitivity on bass drum perception. The findings suggest that Storch’s method prioritizes the spectral properties of the bass drum over traditional pitch values to enhance the bass response. The need to maintain the unique sound of the TR-808 bass drum underscores the importance of spectral formants and register in contemporary popular music production.

1. INTRODUCTION

In popular music, a common practice is to tune the drums to the song’s key [1]. However, in a 2007 interview [2], R&B producer Scott Storch suggests that during the production of music involving a Roland TR-808 drum machine, it may be beneficial to do the opposite and transpose the song’s key to fit the 808 bass drum [3]. The process involves transposing all the tracks but the bass drum. Storch’s motive for undertaking such a potentially time-consuming set of operations is to conserve the characteristic sound of the 808 bass drum.

The present study investigates aspects of the music production process that may explain Storch’s position. In Section 2, we shortly address the issue of drum tuning in popular music. In Section 3, we provide an overview of the importance and usage of the Roland TR-808 in popular music production, focusing on its bass drum voice.

In Section 4.1, we analyze the content of TR-808 bass drum samples. In Section 4.2, we relate spectral features of TR-808 bass drum samples to a diachronic analysis of the power spectrum in popular music. In Section 5, we can understand Storch’s position by involving the frequency response of loudspeakers and the sensitivity of the human ear. Finally, in Section 6, we discuss how the practice suggested by Storch may be a particular case of how properties of the spectrum might be considered more important than pitch values.

2. DRUMS AND TUNING

The musical signal has been divided into two categories: “percussion has a short temporal duration and is rich in noise, while harmonic elements have a long temporal duration with most of the signal energy concentrated in pitch spikes” [4]. “The harmonic and percussive components of music signals have much different structures in the power spectrogram domain, the former is horizontal, while the latter is vertical” [5]. These observations are the basis for source separation methods distinguishing “drums” from “pitched instruments” [6, 7].

Yet, drums can contain pitched content [8]. In drum sounds, relations between eigenfrequencies are not necessarily harmonic [9]. “The tonal elements in drums are usually not structured like partials in a harmonic series. Instead, their frequency relationship can range from inharmonic to chaotic” [10]. From a music producer’s perspective, “drums make several different notes simultaneously” [11].

Recent source separation methods don’t involve prior hypotheses. They’re based on models trained on actual data. Listening to the audio output stemming from such technology indicates that drum stems extracted from popular music do contain pitch. Demonstrations of Steinberg’s SpectraLayers [12], Native Instruments’ iZotope RX 8 [12], iZotope RX 9 [13], and StemRoller [14], provide relevant examples.

If drums contain pitched content, they can be tuned. In popular music, the “[i]ntricate tuning of acoustic drums can have a significant impact on the quality and contextuality of the instrument” [1]. There is no consensus on how to tune drums: “[t]alk to ten different drummers and you’ll get ten different ways to tune drums [...] there’s actually no wrong or right way to tune a drum, or right or wrong pitches to tune it to” [15].



Scott Storch is an American record producer and songwriter. Storch has been referred to as a “producer that changed the R&B game” [16], a “superproducer” [17], *i.e.* a wave of artists “who have established a new degree of visibility for the rap producer, earning star billings virtually equal in prominence to the artists that they produce” [18]. For Scott Storch, drum tuning is an integral part of the music production process:

“I know there’s a lot of producers [who will] put an 808 in the song, and there will be chords and stuff clashing with it, and [...] if [...] your ears are really in tune with that stuff, you realize it’s just like [“not so convincing” kind of gesture]... Sometimes, it actually does something cool to the track, but [...] I like to [...] get into that and tune the kick to match [...] the bass line or whatever the chords are doing [...], I just try different stuff... then... [even when there is] not an incredible amount of tune that carries over regular kicks, like short kicks, and I find myself sometimes at least even trying to tune [...] a regular [...] kick drum sound, and get it close to where most of the chords are in the song...” [2, 0:30]

In the above, Storch mentions the Roland TR-808 bass drum and testifies to tuning bass drums to match the music’s key.

3. THE ROLAND TR-808

The Roland TR-808 Rhythm Composer is an analog drum machine manufactured between 1980 and 1983 [19]. It is “one of the most influential and unique drum machines of its time” [20]. “To this day, the 808 remains a benchmark against which all other analog drum machines are measured” [21]. It can be found in many music genres. The TR-808’s distinctive presets are classic sounds in hip-hop, techno, electro, R&B, and house music [22]. The 808 “play[ed] a central role in the development of acid house” [21]. Pop music star Phil Collins used it throughout his entire career [23, 1:21:28]. It is “a fixture in hip-hop culture, not only as a tool for producers but as a defining sound of the genre” [19]. According to Scott Storch, in modern trap music, producers “live in an 808 world” [24]. One reason for the success of the 808 resides in the fact that “it sounded like nothing else [...] and this is what made it so distinctive” [25]. Perhaps as a result, the 808 has been seen not only as a drum machine but as an “instrument in its own right” [23, 0:06:51].

One notable voice of the 808 is its “long and velvet deep, almost subsonic” bass drum [25], which can be made into a “multi-second-long decaying pseudo-sinusoid with a characteristic sighing pitch” [21]. According to producer Pharrell Williams, the 808 bass drum “filled a massive void in the sound spectrum that wasn’t there [...] once the 808 started to occupy that space, it became like something that you missed if you didn’t have it” [23, 1:20:52].

Over time, the 808 bass drum became used as both kick drum and bass. According to producer Remi Kabaka Jr., “the kick drum would play the bass at the same time [...] there was drums and there was bass, but now the two were sort of fused, so the fill was not just complex and rhythmical, but it was also tonal” [23, 1:11:11]. Musician and writer Alex Lavoie notes that “[i]n most contemporary music genres, especially in trap and hip-hop, the 808 often carries the bassline, providing both the low-end foundation and outlining the harmonic progression of the song” [26]. Musician and producer Charles Burchell writes that the TR-808 “brings a sound closer to a traditional bass line while retaining the power of a drum [...] In many cases, producers will not use a kick drum sample. Instead, they program drum patterns with a tuned 808 as the kick drum” [27].

As a tonal instrument, the 808 bass drum can be tuned: as Lavoie states, “[a]n 808 kick, particularly when it has a long decay, effectively functions as a bass instrument. That’s why tuning your 808s is so crucial” [26]. Lavoie warns that “[i]f the pitch of your 808 kick doesn’t match the key of your song, it can create a dissonant effect” [26].

4. THE 808 BASS DRUM

4.1 Signal analysis of 808 bass drum samples

Figure 1 shows the waveform corresponding to the “TR808 BD Bass Drum Long 01” preset. All samples considered in this paper originate from the TR-808 Trisample library [28]. The waveform confirms that the sample is tonal. The tonal aspect derives from the TR-808 generation technique, during which an oscillator produces a sawtooth wave that is filtered to make it close to a sine wave [29].

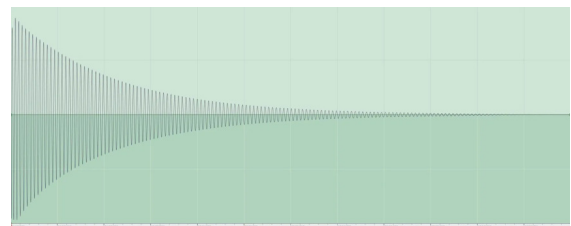


Figure 1. “TR808 BD Bass Drum Long 01” sample, waveform.

Figure 2 shows the STFT for the same sample. Harmonics are present near the start of the sample and then fade out. The sample’s pitch value is briefly higher near the beginning, then decreases to a stable value. A study of the 37 “long” samples from the Trisample library shows that the median range for the initial frequency sweep is close to one half-tone.

The Tristar library features “driven” samples (a reference to the slang term “drive” for “overdrive”, *i.e.* “distortion”). Figure 3 shows the STFT for one “driven” sample. The threshold conditioning the display of the partials as red lines is the same as in Figure 2, which indicates that the distortion boosts the level of the overtones.

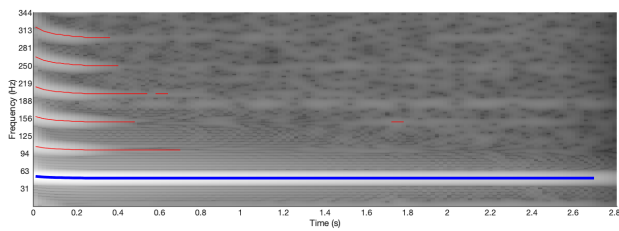


Figure 2. “TR808 BD Bass Drum Long 01” sample, STFT. The horizontal lines follow the fundamental and harmonics. The blue line stops when the energy of the corresponding bin is lower than 0.7 times the peak energy of all bins. The red lines stop when the energy of the corresponding bin is lower than 0.5 times the peak energy of all bins.

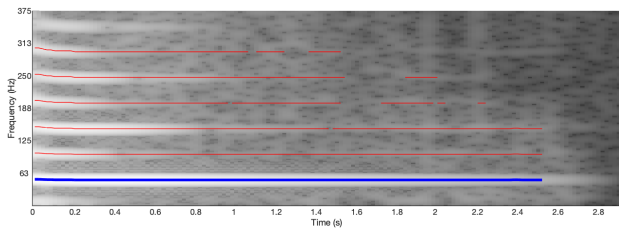


Figure 3. “TR808 BD Bass Drum Driven 01” sample, STFT.

Figure 4 shows the STFT for an extract from the 2017 song “Mask Off”, by the American rapper Future. The track has been described as an example of heavy 808 use [19]. The initial frequency sweep on each bass drum occurrence is similar to the samples shown in Figures 2 and 3. The vertical distribution of high energy values at the beginning of each bass drum occurrence suggests that the 808 is superimposed with a noisier kick drum. The 808 samples are tuned to the song’s tonality (D minor). The pitch values (D1 and B \flat 0) are very low: they stand one minor second and one perfect fourth above the piano’s lowest note. The corresponding frequency range (ca. 40Hz) recalls the “almost subsonic” aspect of the 808 bass drum samples [25].

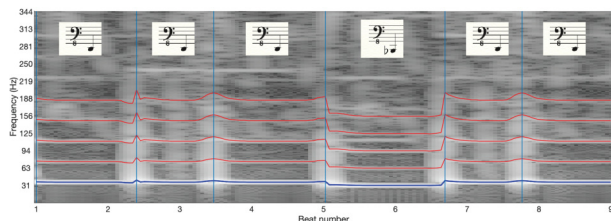


Figure 4. Future, “Mask Off”, 8 beats from 0’25 to 0’30, STFT. The vertical lines denote the kick drum’s onsets. The horizontal lines follow the TR’s fundamental and harmonics. The corresponding pitch values are shown at the top.

4.2 Sub-bass frequencies and the 808 bass drum

Producers recognize three distinct regions of sub-bass: the “boom” (ca. 30Hz), the “thump” (ca. 50Hz) and the

“punch” (ca. 80Hz) [30, pp. 88–118] [31, p. 282]. Figure 5 confirms that 50Hz (the “thump”) is the “frequency range occupied by the Roland TR-808 analog kick” [31].

Before the advent of digital audio, low frequencies were attenuated to protect amplifiers and speakers from the adverse effects of mechanical noise and harmonic distortion [31, p. 282] [32,33]. Musical information in this frequency range only became possible by using digital audio as a medium. Figure 6 shows the evolution of the power spectrum in popular music. The measures were derived from a dataset containing 30435 tracks released between 1961 and 2022. The choice of the tracks stems from the “Best Ever Albums” website, a review aggregator that proposes the best-rated albums for each year of production [34]. For each year, we select the best-rated albums. The overall spectral profile is consistent with Pestana’s results [35]. The increase of energy in the lower band, also testified by Hove et al. [36], is concomitant to the advent of digital audio.

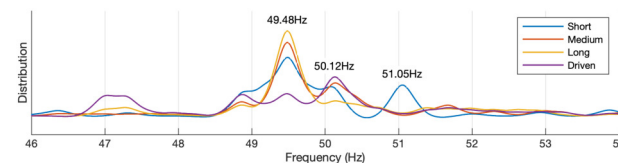


Figure 5. Distribution of fundamental frequencies of TR-808 bass drum samples. The fundamental frequencies are evaluated on 0.2-second windows. The contribution of each window is weighted according to the energy at the fundamental frequency. In the non-“driven” presets, the maximum of the distribution corresponds to $f_0 = 49.48$ Hz. The f_0 values for the “driven” presets are higher. “Short” presets involve a secondary local maximum ($f_0 = 51.05$ Hz) corresponding to the samples’ earliest windows.

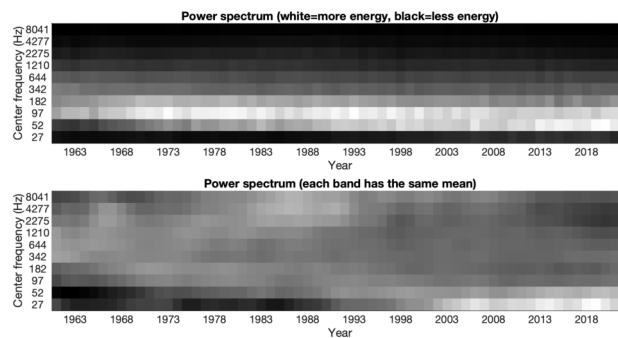


Figure 6. Evolution of the power spectrum in popular music. Top, raw energy values. Bottom, values for each frequency band are normalized to the same mean.

The analysis results shown in Figure 2 indicate that after the initial 0.4s-long attack, “long” 808 samples are based on a single low-frequency sine wave. The sine wave’s frequency is ca. 50 Hz according to Figure 5. The results shown in Figures 3 and 4 suggest that this very low frequency remains an essential component of the 808 bass

drum with added harmonics. Confronting these observations with the power spectrum evolution in popular music (Figure 6), it follows that the sound of the 808 bass drum wasn't fully reproduced before the end of the '90s, even though the machine itself was sold between 1980 and 1983.

After 2009, “the characteristic 808-kick drums [...] started entering mainstream music in general”, and trap music, a “tradition of rap that developed during the 1990s”, an “808 world” according to Scott Storch (see Section 3), “began to reach strong presence on the mainstream Billboard music charts” [37]. So strong is the presence of trap in the charts that this formerly underground genre has been qualified as “pop”, in the sense that “[p]eople’s ears have adjusted” to it [38].

The extended bandwidth provided by the emergence of digital audio made possible the faithful restitution of the entire spectrum of the 808 bass drum, which favored the birth and rise of a music genre that became mainstream and influenced popular music in general.

5. TUNING THE SONG’S KEY TO THE TR-808 BASS DRUM

In Section 2, Scott Storch describes how he tries to tune the bass drum (808 in particular) to the music’s key. Later in the same interview, Storch suggests that instead of tuning the 808 bass drum sample to the song’s tonality, one can do the opposite and adjust the song’s key to the 808 bass drum sample:

“[S]ometimes, producers will program a song in a certain key, and they’ll try to program an 808 under it, and it’s like the key of the song is almost too low to really let speakers do what they need to do with the bass so, I recommend [...] modulating the song up, transposing it up a couple of keys, and you’ll be surprised how much more level you can get out of the song. [Because] anything really below [...] a low E [...], it’s like the speakers are gonna not, let you turn it up, you don’t feel the bass response.” [2, 1:37]

Storch describes a situation in which a producer previously set the key for a song, tunes an 808 bass drum to make it fit the key, and, as a result, the 808 bass drum does not sound “right”.

5.1 Transposition of the TR-808 bass drum: effect on the lowest partial

Let us consider an example where the song’s key is D, as in the extract from Figure 4. We focus on the fundamental, the only lasting component in samples from the “long” type (Figure 2). As seen in Figure 5, the f_0 of an 808 bass drum is ca. 49.5Hz, corresponding to a G1. The producer, therefore, transposes the 808 bass drum one perfect fourth down (5 semitones) to a D1 – one tone below the “low E”

mentioned by Storch. Storch states that the loudspeakers may not reproduce the bass correctly in such a situation.

Professional mixing engineers mainly use near-field monitors [39, p. 3]. With such monitors, they can produce “masters which ‘travel’ well to their use by the record buyers” [40]. The use of near-field monitors extends to producers. Nigel Godrich testifies that during the production of Radiohead’s “OK Computer”, he always used near-field monitors, but never the studios’ main monitors, which “don’t relate to anything” and are “fairly useless” [41]. In the many videos documenting his work, Storch can be seen using near-field monitors.

Newell et al. [40] provide the frequency response for 36 near-field monitoring loudspeakers. Figure 7 graphs the median frequency response for these loudspeakers against the median f_0 for the 808 bass drum samples (49.5Hz / G1) and the TR bass drum median frequency transposed down one perfect fourth (37Hz / D1). The downward transposition results in a gain loss of 6.3 dB. Following Storch’s suggestion and transposing up the song key instead of transposing down the 808 sample would avoid the 6.3dB loss. In Storch’s terms, transposing the song up may “let speakers do what they need to do with the bass”.

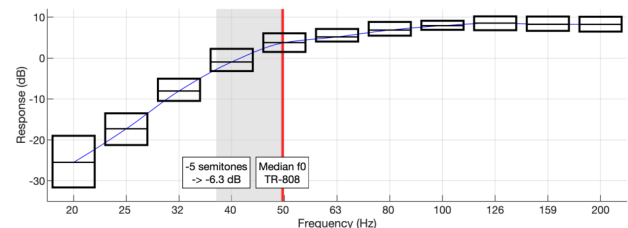


Figure 7. near field loudspeaker responses as a function of frequency. The center horizontal line in each box represents the median, and the two surrounding horizontal lines represent the 25th and 75th percentiles. The blue line shows the smoothed median response. The red vertical line represents the median f_0 for the 808 bass drum as shown in Section 4.2, Figure 5. The gray rectangle denotes a -5 semitone transposition of the median f_0 . The textual representation displays the difference in the response that occurs.

Loudspeakers are not the only frequency-dependent transducers involved in the listening process. The human ear is also sensitive to frequency. In particular, as the frequencies get closer to the lower limit of human hearing, a sine wave with the same sound pressure level but a lower frequency will be perceived as less loud. The phenomenon is accounted for by equal loudness contours, representing the sound pressure levels at different frequencies that are perceived as equally loud [42]. Figure 8 graphs the ISO226-2003 [43] equal loudness contours against the median 808 bass drum f_0 , and the same frequency transposed down one perfect fourth. If we choose a loudness of 60 phon, a +5.5dB gain would be required so that the transposed f_0 remains at the same loudness. Therefore, considering the human ear as one of the transducers in the signal path, the gain it applies to the signal when transposing down the original median f_0 is ca. -5.5dB. As a result,

the overall gain loss following the downward transposition originating from both the loudspeakers and the ear can be estimated to be ca. 11.8 dB.

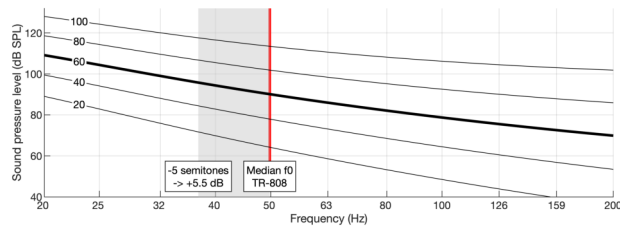


Figure 8. Equal-loudness contours according to [43]. The numbers superimposed on each contour indicate the loudness value corresponding to the contour (in phon). The red vertical line represents the median f_0 for the 808 bass drum as shown in Section 4.2, Figure 5. The gray rectangle indicates a -5 semitone transposition of the median f_0 (one perfect fourth down). The textual representation displays the gain that would be required so that the transposed f_0 remains at the same 60-phon loudness.

If different 808 bass drum notes result in different gains, then a sequence of different 808 bass drum notes will result in gain changes within the sequence. Quoting Storch, “for 808s [...] I try to stay in the comfort zone of the speaker, so I don’t [...] have the volumes jumping out for different notes” [44]. In other words, 808 bass drum parts’ pitch should remain largely static to achieve a stable gain. In turn, largely static bass pitch values may result in a limited variety of chords. The phenomenon illustrates how loudness stability may take precedence over harmonic complexity.

5.2 Transposition of the TR-808 bass drum: involvement of the harmonics

The 808 bass drum samples corresponding to Figures 3 and 4 involve lasting harmonics. Figure 9 shows the combined response deriving from both the near field loudspeakers and the ear’s sensitivity at 60 phon. The lower the frequency, the greater the influence of transposition on the overall gain. The gain loss diminishes with each harmonic. It is almost zero for the fifth harmonic.

We generate a 49.5Hz five-partial harmonic complex tone. The amplitudes of the partials are the same as in the “TR808 BD Bass Drum Driven 01” sample when the frequency values reach a static regime (see Figure 3). The overall power change following a 5-semitone downward transposition is -4.5dB. It is much less than the -11.8 dB gain brought by the downward transposition of the lone fundamental. The result suggests that the issues mentioned by Scott Storch (gain conservation and gain stability) mainly concern the fundamental or, at least, the lowest harmonics. In other words, Storch is specifically concerned with the audibility and stability of the 808 bass drum’s bottom partials.

The phenomenon known as the “missing fundamental” [45] suggests that even if a negative gain is applied to lower

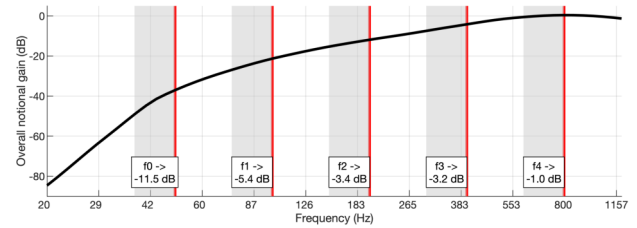


Figure 9. The black line shows the combined response deriving from both the near field loudspeakers and the ear’s sensitivity at 60 phon (Figures 7 and 8). The red vertical lines represent the median values for the 808 bass drum’s fundamental and harmonics. The textual representations display the difference in the response that occurs from a -5 semitone transposition.

harmonics, the perceived pitch remains unchanged due to the auditory system’s temporal pitch processing. Only timbre is affected. In the case of the sub-bass register, *i.e.* frequencies lower than 100Hz according to Fink [31, p. 281], another perceptual aspect may be mentioned. In relation to findings by Takahashi et al. [46], Fink et al. [30, pp. 88-118] suggest that one aspect of the perceptual effects of bass stems from small body surface displacements. According to the author, each sub-bass range can be associated with a body region in which the corresponding frequencies are imaginatively felt. The “boom” (ca. 30Hz) is “the semi-audible vibration in the gut felt during the deepest drops in dancehall and dubstep”. The “thump” (ca. 50Hz) is felt in the stomach, and the “punch” (ca. 80Hz) in the chest. Even when listeners use headphones, bass frequencies may be associated with a “tactile sensation” [47]. Fink [31] and Hove et al. [47]’s views suggest that low frequencies may play a role beyond pitch and timbre, in this case, a haptic role.

A downward transposition and the resulting negative gain applied to these frequencies may affect both the resulting timbre and bodily sensation. As a result, they may be prejudicial to at least some music genres, independently from the presence of upper harmonics.

6. PITCH AND REGISTER

Scott Storch’s advice according to which a song’s key may be adjusted to the 808 bass drum sample is based on the following premise: the transposition of the elements of the music that are not the 808 bass drum is less problematic than the transposition of the 808 bass drum. The change in pitch values doesn’t affect the musical intervals, but the shift in register affects the perceived spectral profile. The change in perceived spectral profile is more important in the case of the TR-808 bass drum due to its low-frequency content.

Following Frisius [48, p. 81], “a [music] theory [that] posits a principle of neutral transposition, according to which groups of pitches essentially do not change their character if one transposes them”, doesn’t take into account the transposition of the sounds themselves. Frisius

remarks that such a theory may not be suited to music from the 20th century. He mentions the composer Luigi Rus-solo, who found it difficult to “transpos[e] sonic gestures into other registers without losing their identity”. According to Frisius, this difficulty is “felt above all when pitch is not clearly definable”. One way to understand the phenomenon is that transposed melodies are only the “same as” each other because they are constructed using a set of pitches whose chromas repeat at the octave. The listener encodes them in terms of pitch sequences. If two sounds are “transpositions” of each other but are not perceived in terms of pitch, then they are just different sounds.

Pitch intervals may be robust to transposition, but the register will change, and so will timbre. The phenomenon has been described in orchestration treatises [49–51]. According to Hector Berlioz, as far as the violin is concerned, C major may be “*grave, mais sourd et terne*” (rich in low frequencies, but dull and muted), and F# minor “*tragique, sonore, incisif*” (tragic, resonant, incisive) [49]. More recently, Reymore et al. [52] have studied the relation between pitch height and timbre in acoustic instruments.

Personal interviews with music producers from the production company Hyper-Music (<https://www.hyper-music.com/>) suggest that in recent popular music, the simultaneous consideration for pitch values and register when considering transposition is paramount. According to one of Hyper-Music producers, Storch prioritizing the register of an instrument over particular pitch values is a “basic rule” of modern music production. Pitch is subservient to spectral formants. Priority is given to the absolute position of the formants in the spectrum. If pitch has to be changed so that the formants of the sound carrying the pitch reach the desired positions, it will be changed. Another producer from the same company claims to be always cautious with transposition, as it may affect timbre. In accordance with Frisius’ point of view, if the pitch content of the part is not too strong, the same producer may simply forego transposition, even if the pitched content of the sample conflicts with other tonal elements.

In productions involving a TR-808 bass drum, Hyper-Music’s producers often set the tonality to D or Eb to take full advantage of the bass drum’s character. Such tonalities neighbor that of the example shown in Figure 4.

7. CONCLUSION

According to Section 2, some authors have previously divided the musical signal into two categories: percussion (rich in noise, short duration), and harmonic elements (long duration, most of the energy concentrated in spikes in the spectrum) [4, 5, 7]. However, other authors have studied the existence of pitch in percussion [8–10]. In music production, drum tuning has been seen as essential [1] but sometimes difficult [8]. Scott Storch, a renowned music producer, has emphasized the importance of fine-tuning drums to match the music’s key despite the inherent challenges in doing so.

In Section 3, we showed that the Roland TR-808 Rhythm Composer has been deemed an influential analog

drum machine [19–21], primarily known for its distinctive and deep bass drum sound [25]. Producers and musicians from various music genres have testified to its efficiency in providing low-end foundation. They use the 808 bass drum not only as a kick drum but also as a tonal instrument that plays basslines, thus emphasizing the importance of its tuning.

Signal analyses of 808 bass drum samples reported in Section 4 show that its fundamental frequency can be found ca. 50Hz and may or may not have lasting harmonics. The measured evolution of the power spectrum in popular music suggests that digital audio technology enabled the faithful reproduction of the 808 bass drum’s extended bandwidth, which played a crucial role in the rise of trap music’s popularity and its subsequent influence on mainstream music [37].

In Section 5, we discussed tuning the song’s key to the 808 bass drum. Producers often try to tune the bass drum to match the song’s key. However, Scott Storch suggests an alternative approach: adjusting the song’s key to fit the 808 bass drum sample. Storch explains that some songs might have a key that is too low for the bass to be correctly reproduced by speakers. Instead, he recommends transposing the music up to achieve a more balanced and powerful bass response. If, for instance, the bass drum is transposed down one perfect fourth to match the song’s key, its fundamental frequency loses ca. 11.8 dB in overall gain, considering the response of near field loudspeakers of the type that producers customarily use [39,40] and the human ear’s sensitivity to frequency [43]. The loss may affect the instrument’s timbre and invalidate the specific bodily sensations the sub-bass range may evoke [30, 31, 47]. The analysis also suggests that the gain loss primarily affects the fundamental frequency and lower harmonics. The discussion emphasizes the importance of controlling the level of bass in music production. It suggests that adjusting the song’s key to the 808 bass drum can indeed be a helpful technique to achieve this goal.

In Section 6, we briefly discussed the relationship between pitch and register in music and how transposition may affect these elements. While classical Western music theory emphasizes the robustness of pitch intervals to transposition, other perspectives [48,52] suggest that transposition has significant consequences on timbre. Orchestration treatises have long associated specific timbral characteristics with different keys, highlighting the importance of considering both pitch and register [49–51]. Recent interviews with popular music producers suggest the approach is significant in modern music production.

An intriguing research direction may stem from the assessment of one of the interviewees, according to which spectral formants have precedence over pitch values in modern popular music. Storch’s handling of the 808 bass drum is an example of this principle. If such a claim proves to have merit, it may have consequences on music analysis and user interaction in generative systems applied to popular music.

8. ACKNOWLEDGMENTS

Many thanks to Yann Macé and Luc Leroy from the music production company Hyper Music for their insights into Scott Storch's work and the subsequent discussions. Special thanks to David Meredith (Aalborg University) for his valuable comments.

9. REFERENCES

- [1] R. Toulson, C. C. Crigny, P. Robinson, and P. Richardson, "The perception and importance of drum tuning in live performance and music production," *The Journal on the Art of Record Production*, vol. 4, 2009.
- [2] B. Zisook, "Remix Hotel Atlanta announces keynote speaker Scott Storch," <https://djbooth.net/features/remix-hotel-atlanta-announces-keynote-speaker-scott-storch-0830072>, 2007, accessed: 2023-11-09.
- [3] Penton Media, "Scott storch on kick drums," <https://youtu.be/R5PUcMAFZOI>, 2007, accessed: 2023-11-09.
- [4] H. Rump, S. Miyabe, E. Tsunoo, N. Ono, and S. Sagayama, "Autoregressive MFCC models for genre classification improved by harmonic-percussion separation." in *ISMIR*, 2010, pp. 87–92.
- [5] N. Ono, K. Miyamoto, H. Kameoka, and S. Sagayama, "A real-time equalizer of harmonic and percussive components in music signals." in *ISMIR*, 2008, pp. 139–144.
- [6] D. Fitzgerald, "Harmonic/percussive separation using median filtering," in *13th International Conference on Digital Audio Effects (DAFX10)*, Graz, Austria, 2010.
- [7] J. Yoo, M. Kim, K. Kang, and S. Choi, "Nonnegative matrix partial co-factorization for drum source separation," in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2010, pp. 1942–1945.
- [8] P. G. Richardson, "Acoustic analysis and tuning of cylindrical membranophones," Ph.D. dissertation, Anglia Ruskin University, 2010.
- [9] P. R. Antunes, "Is it possible to tune a drum?" *Journal of Computational Physics*, vol. 338, pp. 91–106, 2017.
- [10] C.-W. Wu, C. Dittmar, C. Southall, R. Vogl, G. Widmer, J. Hockman, M. Müller, and A. Lerch, "A review of automatic drum transcription," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 9, pp. 1457–1483, 2018.
- [11] J. Roberts, "About drums," <https://circularscience.com/about-drums/>, accessed: 2023-11-09.
- [12] Attack Magazine, "Four of the best stem separation tools," <https://youtu.be/9oNHoE4wHc8&t=870> and [t=1181](https://youtu.be/9oNHoE4wHc8&t=1181), 2022, accessed: 2023-11-09.
- [13] S. Loose, "iZotope RX 9 - fixing drums in the master??" <https://youtu.be/LCH23ZiTXXCA&t=177>, 2022, accessed: 2023-11-09.
- [14] Vervysickbeats, "This Might be the CLEANEST Stem Remover from Songs | StemRoller," <https://youtu.be/-G76oQ3uL90&t=364>, 2023, accessed: 2023-11-09.
- [15] Drum Magazine, "How to tune drums in four steps," <https://drummagazine.com/how-to-tune-drums-in-four-steps/>, accessed: 2023-11-09.
- [16] M. Singleton, "20 producers that the R&B game," https://www.yardbarker.com/entertainment/articles/20_producers_that_the_r_b_game/s1_37845173, Oct. 2022, accessed: 2023-11-09.
- [17] D. Chapman, "'That ill, tight sound': telepresence and biopolitics in post-Timbaland rap production," *Journal of the Society for American Music*, vol. 2, no. 2, pp. 155–175, 2008.
- [18] R. Levine and B. Werde, "Superproducers," <https://www.wired.com/2003/10/producers/>, Oct. 2003, accessed: 2023-11-09.
- [19] Z. Hasnain, "How the Roland TR-808 revolutionized music," *The Verge*, Apr. 2017.
- [20] O. Meyers, "Roland TR-808 Rhythm Composer," McGill University, Tech. Rep., 2003.
- [21] K. J. Werner, J. S. Abel, and J. O. Smith III, "A physically-informed, circuit-bendable, digital model of the Roland TR-808 bass drum circuit." in *DAFx*, 2014, pp. 159–166.
- [22] G. Dayal, "Roland TR-808," *Grove Music Online*, Jan. 2014. [Online]. Available: <https://doi.org/10.1093/gmo/9781561592630.article.A2257229>
- [23] A. Dunn, "808," <https://youtu.be/KClqn0oN11Y>, 2015, accessed: 2023-11-09.
- [24] S. Storch, "Masterclass: becoming a hitmaker with Scott Storch. Chapter 8, 'Defining a bass line,'" <https://www.aular.com/masterclass/scott-storch-becoming-a-hitmaker/>, 2022, accessed: 2023-11-09.
- [25] C. Carter, "Roland TR808 Rhythm Composer (Retro)," *Sound on Sound*, May 1997.
- [26] A. Lavoie, "What is an 808? 7 ways to make huge 808 kicks," <https://blog.landr.com/what-is-an-808/>, Sep. 2020, accessed: 2023-11-09.
- [27] C. Burchell, "Production hacks: Creating 808 basslines," <https://articles.roland.com/production-hacks-creating-808-basslines/>, Nov. 2022, accessed: 2023-11-09.
- [28] Trisamples, "TR-808," <https://trisamples.com/roland-tr808-free-download/>, 2020, accessed: 2023-11-09.

- [29] G. Reid, “Practical bass drum synthesis,” *Sound on Sound*, Feb. 2002.
- [30] R. Fink, M. Latour, and Z. Wallmark, *The relentless pursuit of tone: Timbre in popular music*. Oxford University Press, 2018.
- [31] R. Fink, “The boom in the box: Bass and sub-bass in desktop production,” *The Bloomsbury Handbook of Music Production*, 2020.
- [32] O. Read, *The Recording and Reproduction of Sound. A Complete Reference Manual for the Professional and the Amateur, 2nd edition*. Howard W. Sams Co., 1952.
- [33] A. Millard, *America on record: a history of recorded sound*. Cambridge University Press, 2005.
- [34] Best Ever Albums, “Best ever albums,” <https://www.besteveralbums.com/>, 2023, accessed: 2023-11-09.
- [35] P. D. Pestana, Z. Ma, J. D. Reiss, A. Barbosa, and D. A. Black, “Spectral characteristics of popular commercial recordings 1950-2010,” in *Audio Engineering Society Convention 135*. Audio Engineering Society, 2013.
- [36] M. J. Hove, P. Vuust, and J. Stupacher, “Increased levels of bass in popular music recordings 1955–2016 and their relation to loudness,” *The Journal of the Acoustical Society of America*, vol. 145, no. 4, pp. 2247–2253, 2019.
- [37] J. Kaluža, “Reality of trap: Trap music and its emancipatory potential.” *IAFOR Journal of Media, Communication & Film*, vol. 5, no. 1, 2018.
- [38] C. Lee, “2 Chainz explains why ‘pretty girls like trap music’,” <https://www.rollingstone.com/music/music-features/2-chainz-explains-why-pretty-girls-like-trap-music-talks-his-bucket-list-and-benihana-193850>, 2017, accessed: 2023-11-09.
- [39] M. Senior, *Mixing secrets for the small studio*. Taylor & Francis, 2011.
- [40] P. R. Newell, K. R. Holland, and J. P. Newell, “The Yamaha NS10M: twenty years a reference monitor. Why?” *Proceedings of the Institute of Acoustics*, vol. 23, no. 8, pp. 29–40, 2001.
- [41] A. Robinson, “Radio days,” *The Mix*, Aug. 1997.
- [42] H. Fletcher and W. A. Munson, “Loudness, its definition, measurement and calculation,” *Bell System Technical Journal*, vol. 12, no. 4, pp. 377–430, 1933.
- [43] ISO, “Normal equal-loudness level contours-ISO 226: 2003,” 2003. [Online]. Available: <https://www.iso.org/standard/34222.html>
- [44] S. Storch, “Masterclass: becoming a hitmaker with Scott Storch. Chapter 6, ‘Selecting drum sounds’,” <https://www.aulart.com/masterclass/scott-storch-becoming-a-hitmaker/>, 2022, accessed: 2023-11-09.
- [45] J. C. R. Licklider, “A duplex theory of pitch perception,” *The Journal of the Acoustical Society of America*, vol. 23, no. 1, Supplement, pp. 147–147, 1951.
- [46] Y. Takahashi, K. Kanada, and Y. Yonekawa, “The relationship between vibratory sensation and body surface vibration induced by low-frequency noise,” *Journal of low frequency noise, vibration and active control*, vol. 21, no. 2, pp. 87–100, 2002.
- [47] M. J. Hove, S. A. Martinez, and J. Stupacher, “Feel the bass: Music presented to tactile and auditory modalities increases aesthetic appreciation and body movement.” *Journal of Experimental Psychology: General*, vol. 149, no. 6, p. 1137, 2020.
- [48] R. Frisius, “In search of lost harmony,” *Contemporary music: theoretical and philosophical perspectives*, pp. 77–87, 2010.
- [49] H. Berlioz, *Grand traité d’instrumentation et d’orchestration modernes, dédié à sa majesté Frédéric Guillaume IV roi de Prusse*. Schonenberger, 1844.
- [50] F. A. Gevaert, *Nouveau traité d’instrumentation*. Lemoine & fils, 1885.
- [51] C. Koechlin, *Traité de l’orchestration en 4 volumes*. Eschig, 1954.
- [52] L. Reymore, J. Noble, C. Saitis, C. Traube, and Z. Wallmark, “Timbre semantic associations vary both between and within instruments: An empirical study incorporating register and pitch height,” *Music Perception: An Interdisciplinary Journal*, vol. 40, no. 3, pp. 253–274, 2023.

FRUITSMUSIC: A REAL-WORLD CORPUS OF JAPANESE IDOL-GROUP SONGS

Hitoshi Suda¹ Shunsuke Yoshida² Tomohiko Nakamura¹ Satoru Fukayama¹ Jun Ogata¹

¹ National Institute of Advanced Industrial Science and Technology (AIST), Tokyo, Japan

² The University of Tokyo, Tokyo, Japan

{suda.h, shunsuke.yoshida, tomohiko-nakamura, s.fukayama, jun.ogata}@aist.go.jp

ABSTRACT

This study presents FruitsMusic, a metadata corpus of Japanese idol-group songs in the real world, precisely annotated with who sings what and when. Japanese idol-group songs, vital to Japanese pop culture, feature a unique vocal arrangement style, where songs are divided into several segments, and a specific individual or multiple singers are assigned to each segment. To enhance singer diarization methods for recognizing such structures, we constructed FruitsMusic as a resource using 40 music videos of Japanese idol groups from YouTube. The corpus includes detailed annotations, covering songs across various genres, division and assignment styles, and groups ranging from 4 to 9 members. FruitsMusic also facilitates the development of various music information retrieval techniques, such as lyrics transcription and singer identification, benefiting not only Japanese idol-group songs but also a wide range of songs featuring single or multiple singers from various cultures. This paper offers a comprehensive overview of FruitsMusic, including its creation methodology and unique characteristics compared to conversational speech. Additionally, this paper evaluates the efficacy of current methods for singer embedding extraction and diarization in challenging real-world conditions using FruitsMusic. Furthermore, this paper examines potential improvements in automatic diarization performance through evaluating human performance.

1. INTRODUCTION

In Japanese pop culture, an *idol* is a performer who engages in dancing, singing, and entertaining fans [1]. In the culture, idols frequently participate in activities, such as concerts and television programs, as members of idol groups. One of the most renowned contemporary idol groups is AKB48, which has 40 single compact discs (CDs) that are million-sellers, as certified by The Recording Industry Association of Japan¹. FRUITS ZIPPER has

¹ https://www.riaj.or.jp/f/data/cert/gd_search.html

emerged as another notable group comprising seven girls and being awarded the Best New Artist at the Japan Record Awards 2023, the most prestigious accolade in Japanese music culture [2]. Not only can fans attend concerts, but they can also interact with the idols at handshaking events (*Akushukai*) or bonus events (*Tokutenkai*), where the fans can forge deep connections with the idols [3].

Idol-group songs feature several unique characteristics. One notable characteristic is *song division*, also called *utawari* in Japanese [4, 5]. This approach involves a dynamic vocal arrangement where the singing roles shift throughout the song; individual members may take turns singing solo lines, or multiple members may sing together in unison. In particular, the entire group often sings together in the chorus sections, known as *sabi*. Song division is chosen intentionally to maximize the charm and attractiveness of each idol and song. Therefore, the analysis of song division is crucial for understanding the structure and expression of songs, as well as the creators' intentions.

Song division plays a crucial role also in shaping audience participation through chants and shouts, known as *calls* and *mixes*, which are indispensable elements of idol-group concerts [6]. Fans spontaneously create these chants and shouts, reflecting the song's structure, musical intensity, and song division, specifically which member is assigned to sing at any given moment. Furthermore, song division significantly influences music videos and concert recordings produced by idol groups, demonstrating its pivotal role in producing and appreciating idol music content.

As previously described, song division is crucial for understanding and enjoying the musical compositions of idol groups. To aid fans' comprehension, some idol groups release official charts showing how songs are divided among members. For Korean pop groups with similar features to Japanese idol groups, several fans create *line distribution* videos. These videos, widely viewed on platforms like YouTube and TikTok, visualize the structures of song division, facilitating a deeper understanding. Therefore, developing techniques for recognizing song division will help fans enjoy the music compositions and enhance their interaction with idols. In addition, such advancements will support creators in promoting idol groups.

The task to estimate song division, i.e., *who sings when*, within a music signal is known as *singer diarization*. This technique has been inspired by speaker diarization, which identifies *who speaks when* in conversational speech [7–9]. The singer diarization technique was initially introduced



to analyze folk music and has been adapted for Japanese idol-group songs [4]. However, existing research has not examined songs from real-world idol groups but from idol-themed games and anime. These game and anime songs generally belong to narrower genres, feature simpler song division structures, and have vocals that are easier to distinguish, thanks to the distinctive voice qualities of the voice actors. Further research indicates that in-the-wild audio signals can improve diarization performance in real-world settings, even with small datasets [4, 10]. Consequently, compiling a dataset featuring songs from real-world idol groups is critical for developing practical applications targeting pop culture.

This study addresses the demand for a practical dataset in music information retrieval (MIR) by constructing a new corpus, FruitsMusic. This corpus consists of detailed annotations about *who sings what and when* in real-world songs performed by Japanese idol groups from YouTube, enabling the advancement of singer diarization techniques and their assessments. Beyond singer diarization, FruitsMusic also advances various MIR techniques such as lyrics transcription [11, 12], emotion classification [13, 14], singer identification [15, 16], and singer-based music search [17], for not only Japanese idol-group songs but also a wide array of musical pieces featuring single or multiple singers from different cultures. A significant advantage of FruitsMusic is its focus on real idol groups, allowing for evaluations in challenging scenarios and enhancing the applicability of MIR techniques in the real world. This paper details the structure, development methodology, and unique characteristics of FruitsMusic. The paper also demonstrates the applications of evaluating existing methods in two MIR tasks, singer embedding extraction and diarization, in real-world scenarios.

2. STRUCTURE AND CONSTRUCTION METHODOLOGY OF FRUITSMUSIC

In this study, we constructed FruitsMusic (Corpus of Fully Real-World Popular Idol-group Songs from YouTube Videos for Music Information Processing) aimed at developing and evaluating various MIR techniques. This corpus is a collection of annotations for 163 minutes of music video content on YouTube, detailing *who sings what and when*. The corpus includes annotations for 40 songs performed by 18 different groups, featuring a total of 122 unique female singers, all approximately 20 years of age. The corpus is available at <https://huggingface.co/datasets/fruits-music/fruits-music>².

2.1 Related Works

Several corpora derived from YouTube have been constructed across various research fields. The key advantage of this approach is the utilization of a wide range of real-world video and audio content.

For example, ActivityNet and YouTube-8M are benchmark datasets widely used in video processing [18, 19].

² This paper has been written based on FruitsMusic version 1.2.0.

```
{
  "id": "XXm01",
  "youtubeId": "YouTube ID",
  "type": "music_video",
  "singerIds": ["XXs01", "XXs02", "XXs04", "XXs05", "XXs06"],
  "title": "Song Title",
  "songStartsAt": 0,
  "duration": 216128,
  "states": [
    {
      "start": 1869,
      "end": 17233,
      "singers": [0, 1, 2, 3, 4],
      "lyrics": "Lyrics 1",
      "realLyrics": null
    },
    {
      "start": 22543,
      "end": 26930,
      "singers": [1],
      "lyrics": "Lyrics 2",
      "realLyrics": null
    }
  ]
}
```

(a) JSON file

```
SPEAKER XXm01 1 1.869 15.364 <NA> <NA> XXs01 <NA> <NA>
SPEAKER XXm01 1 1.869 15.364 <NA> <NA> XXs02 <NA> <NA>
SPEAKER XXm01 1 1.869 15.364 <NA> <NA> XXs04 <NA> <NA>
SPEAKER XXm01 1 1.869 15.364 <NA> <NA> XXs05 <NA> <NA>
SPEAKER XXm01 1 1.869 15.364 <NA> <NA> XXs06 <NA> <NA>
SPEAKER XXm01 1 22.543 4.387 <NA> <NA> XXs02 <NA> <NA>
```

(b) RTTM file

Table 1. An example of JSON and RTTM files.

Similarly, YouTube-ASL, a large-scale American Sign Language corpus, originates from YouTube [20].

In the field of audio processing, several corpora have utilized YouTube videos. AudioSet, for instance, is widely adopted for recognizing and detecting audio events [21]. VoxLingua107 covers 6,628 hours of speech across 107 languages and is helpful to language detection [22]. Further, JTubeSpeech consists of extensive Japanese speech data from YouTube and helps the development of diverse speech processing techniques [23]. Similarly, YODAS consists of 500,000 hours of speech in over 100 languages and makes multilingual speech processing techniques applicable in the wild [23, 24]. Coco-Nut is another corpus with subjective descriptions of voices, designed for controlling speaker identity based on text prompts [25].

These prior works underscore the effectiveness of YouTube-based corpora, which we also adopted in this study. Our corpus focuses especially on accuracy and reliability, which are less emphasized in these prior corpora. In addition, the video-based nature of FruitsMusic facilitates multimodal processing, such as multimodal diarization [26]. Note that these prior corpora have been curated to protect individual privacy rights by excluding personal information, and FruitsMusic also maintains these ethical standards.

2.2 Structure of the Corpus

FruitsMusic includes annotations in JavaScript Object Notation (JSON) format, Rich Transcription Time Marked (RTTM) files for diarization, and text files of lyrics. Table 1 presents an example of JSON and RTTM files.

2.2.1 JSON Files

The JSON files include the following information:

- **Song ID.** This field is formed by combining a two-character idol-group ID, the letter “m”, and a two-digit ID.
- **Video ID on YouTube.**
- **Type of the video.** This field is either of `music_video`, `middle_music_video`, or `dance_practice`. The names of these types are derived from traditions in Japanese idol culture.
- **List of singer IDs.** Each ID is formed by combining a two-character idol-group ID, the letter “s”, and a two-digit ID.
- **Song title.** This field aims at natural language processing (NLP) tasks.
- **Start time and duration of the song.** The videos may contain content beyond songs, such as comments from idols. This information is provided to help filter out such content.
- **Singing states.** This is a list of the start and end times of the segment, the singers assigned to the segment, and the lyrics. The `lyrics` field contains the official lyrics, which may differ from the actual lyrics sung. In such cases, the `realLyrics` field is used.

The time and duration fields are annotated in milliseconds.

2.2.2 RTTM Files

The RTTM format is specially designed for speaker diarization tasks, identifying *who speaks when* [27]. Table 1b presents an example of an RTTM file. Within this format, each line details the start time and duration of the segment, as well as the singer’s ID. For simultaneous singing, the format allocates a separate line to each singer, resulting in multiple lines corresponding to the number of singers.

2.2.3 Text Files of Lyrics

Lyrics lines may be duplicated in the JSON files to precisely represent *who sings what and when* (e.g., DRm03). As a result, extracting lyrics from JSON files is not straightforward. To support the development and assessment of techniques involving lyrics, such as lyrics transcription, FruitsMusic provides separate text files of lyrics.

2.3 Subsets

FruitsMusic is split into Subset A and Subset B. Subset A is designed mainly for training, and Subset B is for evaluation. However, both subsets can be arbitrarily used for various purposes. Subset A contains 32 songs, while Subset B has 8 songs. To ensure unbiased evaluation, Subset B does not contain any singers from Subset A, and each group in Subset B contributes only one song. The songs in Subset B were chosen to cover various genres (dance, rock, synthpop, etc.) and division styles. Also, groups in Subset B are generally less famous than those in Subset A, which helps ensure fairer and less biased human evaluation.

	CHiME-5	FruitsMusic
Average audio length	9031 s	244 s
# Speakers	4	4–9
Average segment length	2.11 s	4.44 s
Total length per speaker	1159.6 s	15.9 s
Segments without speakers	22.3%	23.9%
Solo segments	51.4%	42.6%
Multiple-speaker segments	26.4%	33.5%
Segments with 3+ speakers	6.4%	26.5%

Table 2. Comparison of FruitsMusic with the CHiME-5 dataset [28], a conversational speech dataset. The “Total length per speaker” row indicates the average total duration per speaker in each audio.

2.4 Song Selection

We meticulously selected the songs for FruitsMusic to ensure the corpus’s reliability and usefulness. To achieve accurate annotations, we initially gathered extensive knowledge about the idol groups. We then used reliable sources, including concert recordings and official announcements, for information. Additionally, to support applications like singer diarization, each singer has at least one solo section within the database. Moreover, we assign each singer to only one group in FruitsMusic. While idols may participate in multiple groups or move between groups in reality, we avoid such complexities in this database. FruitsMusic focuses solely on contemporary songs released from 2022 onwards to reflect the latest music trends.

2.5 Rules

This corpus has been constructed using copyrighted materials. Users are required to follow the licensing agreement specified in the corpus documentation to protect the rights of creators and idols. The agreement sets three major rules. First, the copyrighted content of this corpus, such as lyrics texts, is not intended for appreciation or entertainment. Second, the corpus cannot be used to develop or enhance generative artificial intelligence (AI) techniques, such as singing voice synthesis, voice conversion, lyrics generation, and music creation. However, users can utilize the corpus for recognition or information extraction tasks, including lyrics recognition, singer embedding extraction, and assessing the naturalness of lyrics or music. Third, when citing this corpus in any media, including academic works and presentations, users are required to identify both the groups and the singers using the provided IDs and refrain from using their real names. If the mention of song names is not essential for the discussion, users are also required to refer to them by their respective IDs.

3. COMPARISON WITH CONVERSATIONAL DATASET

This section compares FruitsMusic with the CHiME-5 dataset, a conversational speech dataset designed for The

5th CHiME Speech Separation and Recognition Challenge [28], to explore the differences between conversational speech and songs with song division. The CHiME-5 dataset contains 20 conversational speech instances, each from four speakers. Table 2 shows the comparison results, considering all subsets of both CHiME-5 and FruitsMusic.

Initially, the average audio length in FruitsMusic is significantly shorter than in CHiME-5. Unlike conversational speech, which is not limited by specific length constraints, the duration of songs is tightly controlled by the structure of the musical compositions. Furthermore, the average total duration of speech segments per speaker in FruitsMusic is extremely shorter than in CHiME-5. This difference arises from the shorter overall audio length and the larger number of singers in FruitsMusic. Since solo segments play a key role in capturing singer characteristics, developing singer identification techniques under these challenging conditions, different from conversational speech, is essential for improving singer diarization and other MIR systems for songs featuring multiple singers.

The comparison reveals a noteworthy difference in the frequency of simultaneous speakers between CHiME-5 and FruitsMusic. In particular, sections featuring 3 or more singers in FruitsMusic are significantly longer than in CHiME-5. This indicates that the methods that treat overlapping speech as segments with two speakers, often adopted in speaker diarization [29], cannot be directly applied to singer diarization. Furthermore, about 60% of segments with multiple singers feature vocals from only a subset of the entire group. Hence, the assumption that all singers are present in overlapped segments proves ineffective for singer diarization; it is crucial to accurately and independently determine the vocal activity of each singer.

4. APPLICATION 1: SINGER EMBEDDINGS

Singer embeddings are multidimensional vectors that capture each singer’s unique vocal traits. In singing information processing, high-quality singer embeddings are crucial for enhancing the performance of tasks involving singers, such as singer identification, voice matching, and singer diarization. This section evaluates two types of embeddings extracted from song segments by a specific group and discusses the effectiveness of each extraction technique in real-world scenarios. This section visualizes these embeddings to understand their effectiveness in distinguishing singers and provides a numerical analysis of the clustering performance based on singers.

In our evaluation, we compare two types of singer embeddings. The first type involves x-vectors, traditional yet effective speaker embeddings derived from deep neural networks (DNNs) for speaker identification [30]. Specifically, we utilize an x-vector extractor `microsoft/wavlm-base-plus-sv`³, which incorporates WavLM, a large-scale pre-trained model based on self-supervised learning [31]. Second, we evaluated embeddings based on ECAPA-TDNN, an enhanced time de-

lay neural network (TDNN) in x-vector extractors [32]. ECAPA-TDNN-based embeddings have been proven to show remarkable performance in speaker recognition and diarization [32–34]. We used an ECAPA-TDNN model provided by SeechBrain⁴ [35]. In addition, this evaluation considers both mixed and vocal signals, with the latter extracted using Demucs, an open-source music source separation tool [36, 37]. We utilized the `htdemucs_ft` model, a fine-tuned version of the Hybrid Transformer Demucs, renowned for its state-of-the-art performance in music source separation.

We focus on the group KF, which comprises seven members and has eight songs, the most available on FruitsMusic. For this study, we selected segments where a singer performs solo for over 2 seconds. On average, each singer has 20 segments, totaling approximately 101 seconds of solo performance.

As objective evaluation metrics, F values are calculated to benchmark clustering efficacy [38]. Here, the F value is the harmonic mean of two metrics: purity P and inverse purity I . The P and I are defined as follows:

$$P = \frac{1}{N} \sum_i \max_j |C_i \cap S_j|, \text{ and} \quad (1)$$

$$I = \frac{1}{N} \sum_j \max_i |C_i \cap S_j|. \quad (2)$$

In these equations, C_i is the i -th cluster, and S_j is the set of the j -th singer’s samples. High F values indicate superior performance, with a theoretical maximum of 1. For this evaluation, spectral clustering [39] was performed to create 7 clusters, matching the number of singers. All the embeddings were L_2 -normalized in advance.

Figure 1 shows the visualizations of the acquired embeddings by reducing their dimensions into two using t -SNE. The effectiveness of Demucs is confirmed across both extraction methods. Compared to x-vectors, embeddings derived with ECAPA-TDNN provide more expressive singer representations. Specifically, Figure 1d reveals that samples from certain singers, specifically KFs01 (represented in red circles) and KFs06 (in pink stars), tend to gather by singer identity. Hence, ECAPA-TDNN-based embeddings are proved to effectively capture singers’ unique identities even from short singing segments. This shows the advantages of the ECAPA-TDNN methodology over conventional TDNN in x-vector extractors. However, none of the plots show distinct clusters visibly forming, and the highest F value was only 0.64. This indicates that tasks like singer diarization and number estimation remain challenging using any embedding extractor evaluated. Since the ECAPA-TDNN model is trained with speech datasets, fine-tuning it with singing voice datasets will enhance its performance. Note that the separated vocal signals are distorted; therefore, using datasets with both clean and mixed or separated signals from real-world conditions will be effective.

³ <https://huggingface.co/microsoft/wavlm-base-plus-sv>

⁴ <https://huggingface.co/speechbrain/spkrec-ecapa-voxceleb>

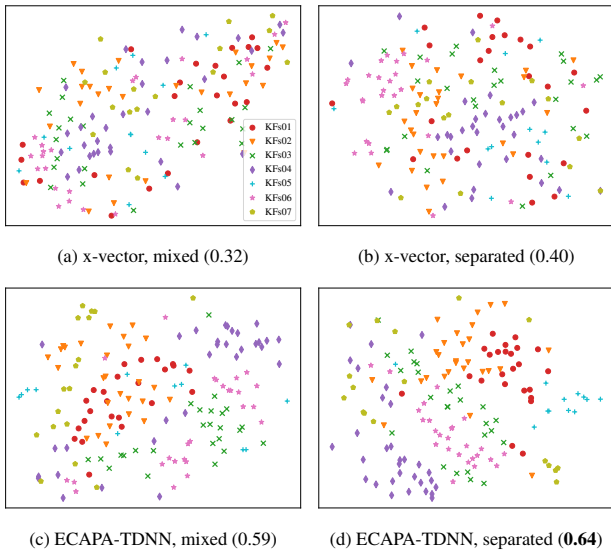


Figure 1. *t*-SNE visualizations of singer embeddings from the idol group KF’s songs, where each color and shape represents a different singer. Captions detail the extraction methods and whether Demucs was applied. Values in parentheses represent F values, measuring the clustering performance.

5. APPLICATION 2: SINGER DIARIZATION

To evaluate the efficacy of FruitsMusic in training singer diarization models, we trained several models with Subset A of FruitsMusic and assessed their performance using Subset B. In this comparison, the number of singers was not given to the systems. Furthermore, we engaged a human evaluator to perform the manual diarization of songs in Subset B and discuss the potential advancements in automatic diarization performance.

5.1 Construction of a Synthesized Dataset

To improve the diarization performance, we utilized songs from commercial CDs in addition to FruitsMusic. This dataset consists of 272 songs performed by multiple singers, with a separate recording for each song and singer combination. For example, if three singers perform song A, each of the three singers has solo recordings: one by singer 1, another by singer 2, and a third by singer 3. On average, each song features 4.1 singers, resulting in a total of 1126 recordings. All the songs were sourced from idol-themed games and anime, and the singers were 129 unique female voice actresses. We executed source separation on all 1126 recordings using Demucs to generate isolated vocal and accompaniment signals.

We generated five song division patterns for each song, capping the number of singers to a maximum of seven. We applied voice activity detection (VAD) first and randomly assigned singers to each segment. During the assignment, a single singer was allocated to 60% of the segments, all singers to 23%, and random singers to the remaining 17%. We mixed the vocal signals based on the generated song division and combined the sepa-

rated accompaniment with the mixed vocal tracks to create the final mixture. In these generated songs with song division, singers perform in unison during segments with multiple singers. The VAD process used `pyannote/voice-activity-detection`⁵.

5.2 Evaluated Systems

In this experiment, the following systems are compared.

5.2.1 SA-EEND with EDA

The first approach adopted Self-Attentive End-to-End Neural Diarization (SA-EEND) [10]. Since the number of singers for the evaluated signals was unknown, we used enhanced SA-EEND with Encoder-Decoder-based Attractors (EDA) [40]. The hyperparameters matched those of the CALLHOME dataset, as specified in the original publication [40]. The input signals were downsampled to 8000 Hz and were converted to monaural signals.

5.2.2 `pyannote.audio`

The second method used `pyannote.audio`⁶, an open-source toolkit for speech processing tasks [29, 41]. The diarization workflow is structured as a pipeline process, incorporating PyanNet-based modules. To conduct this experiment, we fine-tuned the publicly available pre-trained model `pyannote/speaker-diarization-3.1`⁷ using the prepared song datasets. This fine-tuning process adapted the segmentation models and optimized the thresholds for both segmentation and clustering. The input signals were downsampled to 16 000 Hz and converted to monaural signals.

5.2.3 Human Evaluator

In addition to the automatic diarization approaches, we also engaged a human evaluator to perform manual singer diarization to gauge the achievable performance. The individual understands Japanese and often listens to Japanese pop music (about 60 hours a month), yet was completely unfamiliar with any of the songs in Subset B of FruitsMusic. To maintain the experiment’s integrity, we presented only the audio signals of the songs without any corresponding videos. The participant was allowed to use any external tool to aid in the diarization process but was explicitly restricted from searching for the songs on the internet.

5.3 Experimental Setup

As a training dataset, Subset A from FruitsMusic was used. The songs DRm01, Kfm01, Rgm01, SBm01, and SYm01 were designated for validation. The remaining songs, excluding three songs featuring nine singers, were allocated for training. Due to the highly extended training time required for SA-EEND with EDA for songs featuring more than seven singers, three songs with nine singers, VYm02,

⁵ <https://huggingface.co/pyannote/voice-activity-detection>

⁶ <https://github.com/pyannote/pyannote-audio>

⁷ <https://huggingface.co/pyannote/speaker-diarization-3.1>

System	Mixed	Separated
SA-EEND with EDA		
Synthesized only	99.5%	101.3%
Synthesized + FruitsMusic	103.2%	83.8%
pyannotate.audio		
Synthesized only	92.9%	69.9%
Synthesized + FruitsMusic	91.3%	50.3%
Human	22.7%	—

Table 3. DER for Subset B in FruitsMusic with the several diarization systems.

VYm03, and XSm02, were excluded from the dataset. The loudness of all songs was normalized to -14 LUFS.

The evaluation metric used was the diarization error rate (DER) [27], defined as:

$$\text{DER} = \frac{\sum_{s=1}^S d_s \left[\max \left(N_s^{(\text{ref})}, N_s^{(\text{hyp})} \right) - N_s^{(\text{correct})} \right]}{\sum_{s=1}^S d_s N_s^{(\text{ref})}}. \quad (3)$$

Here, S is the total number of segments, d_s represents the duration of the s -th segment, and $N_s^{(\text{ref})}$, $N_s^{(\text{hyp})}$, and $N_s^{(\text{correct})}$ correspond to the number of ground-truth singers, estimated singers, and accurately identified singers in the s -th segment, respectively. According to this definition, DER can exceed 100%. The calculation of DER was performed with `dscore`⁸, an open-source tool. Due to the implementation of `dscore`, self-overlapped segments, which contain multiple recordings of the same singer, were normalized in the calculation process. The collar size, the time ignored in DER calculation around segment boundaries, was set to zero.

The model selection criterion was achieving the minimum DER on the validation set. For each condition, we developed two versions of the system: one trained on mixed signals and another trained on extracted vocal signals. The vocal signal extraction was performed using the `htdemucs_ft` model of `Demucs` [36, 37].

5.4 Results

Table 3 shows the DER of all the systems. The performance of the mixed signal systems is significantly inferior to that of the separated signal systems. In other words, across evaluated systems, `Demucs` effectively improved diarization performance; hence, a pipeline system combining source separation and diarization proved more effective than using a single system on mixed signals in the case of this evaluation. In both approaches, `SA-EEND` and `pyannotate.audio`, training with `FruitsMusic` significantly improved the overall performance, particularly for the separated signal systems. The results suggest that `FruitsMusic`, despite its smaller size, can significantly enhance diarization performance rather than relying solely on large-scale synthesized datasets.

⁸ <https://github.com/nryant/dscore>

System	BD	BI	JA	JY	MG	QD	SL	TJ
SA-EEND with EDA								
w/o FruitsMusic	1	3	2	2	4	0	6	2
w/ FruitsMusic	2	2	2	2	2	2	2	2
pyannotate.audio								
w/o FruitsMusic	3	5	3	3	3	3	3	3
w/ FruitsMusic	7	7	7	7	7	6	7	7
Human	8	6	6	5	7	5	6	4
Ground truth	9	4	7	5	7	5	6	4

Table 4. Estimated total number of singers derived from diarization results. All the systems used separated vocal signals using `Demucs`. Each column shows a song in Subset B. The suffixes “m01” of song IDs are omitted.

Table 4 shows the estimated number of singers included in the diarization results. The `SA-EEND`-based systems struggled to distinguish singers accurately. This seems due to the difficulties of naive DNN-based methods in distinguishing singer identities, as discussed in Section 4. On the other hand, `pyannotate.audio` demonstrated an almost invariant estimation of the number of singers. This indicates a potential overfitting to the training datasets, with the most common number of singers in the training set tending to dominate the predictions.

Among the evaluated systems, human performance was remarkably superior to the automatic diarization systems in terms of DER. Notably, a human evaluator accurately estimated the number of singers in 5 out of 8 songs. This demonstrates that humans can effectively distinguish individual singers’ voices even within mixed music signals. Therefore, these results proved a significant potential for improving both automatic singer identification and diarization performance.

6. CONCLUSION

This paper presents `FruitsMusic`, a novel corpus of precise annotations on *who sings what and when* in Japanese idol-group songs. The song selection and subset creation were meticulously conducted to facilitate unbiased evaluation and ensure usefulness across a wide range of genres, song division styles, and idol groups. The corpus can be applied to various MIR tasks, such as singer diarization, singer identification, and lyrics transcription. This paper showcases its applications in evaluating singer embedding extraction and diarization techniques. The results showed that distinguishing singers from short singing segments remains challenging, despite effective methods in speech processing. The paper also suggests potential advancements in automatic diarization performance by assessing human performance. We acknowledge significant existing areas for performance improvement in diverse MIR tasks, and we are confident that `FruitsMusic` has the potential to advance various techniques among them.

7. ACKNOWLEDGMENTS

This work was supported by JSPS KAKENHI Grant Number JP23K20017. This paper is based on results obtained from a project, JPNP20006, commissioned by the New Energy and Industrial Technology Development Organization (NEDO). This study was supported by the BRIDGE program of the Cabinet Office, Government of Japan.

8. ETHICS STATEMENT

The FruitsMusic corpus is derived from YouTube videos of Japanese idol groups and consists of annotations that detail *who sings what and when*. Hence, FruitsMusic has the potential to help develop singing voice synthesis (SVS) or voice conversion (VC) systems that replicate the voices of actual idols. Moreover, analysis of the videos within FruitsMusic enables associating the real names of groups and singers with their IDs. These scenarios raise potential concerns about infringing on the personality rights of the idols. Additionally, using FruitsMusic to construct or trigger other generative AI techniques, such as lyrics and music generation, may violate the rights of composers and creators. In light of these considerations, as described in Section 2.5, FruitsMusic enforces the stringent licensing agreement and requires all users to adhere to it when downloading and using the corpus.

9. REFERENCES

- [1] P. W. Galbraith and J. G. Karlin, "Introduction: The mirror of idols and celebrity," in *Idols and Celebrity in Japanese Media Culture*, P. W. Galbraith and J. G. Karlin, Eds., Aug. 2012, pp. 1–32.
- [2] "FRUITS ZIPPER was awarded the Best New Artist at the Japan Record Awards 2023 (in Japanese)," *Sankei Shimbun*, Dec. 2023.
- [3] S. Nozawa, "The concentration booth and the handshaking lane: Ideologies of the phatic," *International Journal of the Sociology of Language*, vol. 2023, no. 284, pp. 15–36, Nov. 2023.
- [4] H. Suda, D. Saito, S. Fukayama, T. Nakano, and M. Goto, "Singer diarization for polyphonic music with unison singing," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 1531–1545, May 2022.
- [5] Y. Okada, Ed., *Living with Idols (in Japanese)*. Pot Publishing, Jul. 2013.
- [6] W. Xie, "Japanese "idols" in trans-cultural reception: the case of AKB48," in *The Art of Reception*, J. Bracker and A.-K. Hubrich, Eds., Mar. 2021, pp. 371–399.
- [7] M. Thlithi, C. Barras, J. Pinquier, and T. Pellegrini, "Singer diarization: application to ethnomusicological recordings," in *Proc. of the 5th International Workshop on Folk Music Analysis (FMA 2015)*, Jun. 2015, pp. 124–125.
- [8] X. Anguera, S. Bozonnet, N. Evans, C. Fredouille, G. Friedland, and O. Vinyals, "Speaker diarization: A review of recent research," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 2, pp. 356–370, Feb. 2012.
- [9] T. J. Park, N. Kanda, D. Dimitriadis, K. J. Han, S. Watanabe, and S. Narayanan, "A review of speaker diarization: Recent advances with deep learning," *Computer Speech & Language*, vol. 72, no. C, pp. 1–34, Mar. 2022.
- [10] Y. Fujita, N. Kanda, S. Horiguchi, Y. Xue, K. Nagamatsu, and S. Watanabe, "End-to-end neural speaker diarization with self-attention," in *Proc. of the 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Dec. 2019, pp. 296–303.
- [11] A. Mesaros and T. Virtanen, "Automatic recognition of lyrics in singing," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2010, no. 1, pp. 1–11, Feb. 2010.
- [12] X. Gao, C. Gupta, and H. Li, "Automatic lyrics transcription of polyphonic music with lyrics-chord multi-task learning," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 2280–2294, Jul. 2022.
- [13] D. Yang and W.-S. Lee, "Music emotion identification from lyrics," in *Proc. of the 11th IEEE International Symposium on Multimedia*, Dec. 2009, pp. 624–629.
- [14] Y. E. Kim, E. M. Schmidt, R. Migneco, B. G. Morton, P. Richardson, J. Scott, J. A. Speck, and D. Turnbull, "Music emotion recognition: A state of the art review," in *Proc. of the 11th International Society for Music Information Retrieval Conference (ISMIR 2010)*, vol. 86, Aug. 2010, pp. 937–952.
- [15] C. Nithin and J. Cheriyan, "A novel approach to automatic singer identification in duet recordings with background accompaniments," in *Proc. of the 2014 Annual International Conference on Emerging Research Areas: Magnetics, Machines and Drives (AICERA/iCMMMD)*, Jul. 2014, pp. 1–6.
- [16] H. Fujihara, M. Goto, T. Kitahara, and H. G. Okuno, "A modeling of singing voice robust to accompaniment sounds and its application to singer identification and vocal-timbre-similarity-based music information retrieval," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 3, pp. 638–648, Mar. 2010.
- [17] M. Goto, T. Saitou, T. Nakano, and H. Fujihara, "Singing information processing based on singing voice modeling," in *Proc. of the 2010 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2010, pp. 5506–5509.

- [18] F. C. Heilbron, V. Escorcía, B. Ghanem, and J. C. Niebles, "ActivityNet: A large-scale video benchmark for human activity understanding," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2015.
- [19] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan, "YouTube-8M: A large-scale video classification benchmark," *arXiv [cs.CV] 1609.08675*, Sep. 2016.
- [20] D. Uthus, G. Tanzer, and M. Georg, "YouTube-ASL: A large-scale, open-domain american sign language-english parallel corpus," in *37th Conference on Neural Information Processing Systems (NeurIPS 2023) Track on Datasets and Benchmarks*, Jun. 2023.
- [21] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. Channing Moore, M. Plakal, and M. Ritter, "Audio Set: An ontology and human-labeled dataset for audio events," in *Proc. of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2017, pp. 776–780.
- [22] J. Valk and T. Alumäe, "VoxLingua107: A dataset for spoken language recognition," in *Proc. of the 2021 IEEE Spoken Language Technology Workshop (SLT)*, Jan. 2021, pp. 652–658.
- [23] S. Takamichi, L. Kürzinger, T. Saeki, S. Shiota, and S. Watanabe, "JTubeSpeech: corpus of japanese speech collected from YouTube for speech recognition and speaker verification," *arXiv [cs.SD] 2112.09323*, Dec. 2021.
- [24] X. Li, S. Takamichi, T. Saeki, W. Chen, S. Shiota, and S. Watanabe, "Yodas: Youtube-Oriented dataset for audio and speech," in *Proc. of the 2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Dec. 2023, pp. 1–8.
- [25] A. Watanabe, S. Takamichi, Y. Saito, W. Nakata, D. Xin, and H. Saruwatari, "Coco-Nut: Corpus of Japanese utterance and voice characteristics description for prompt-based control," in *Proc. of the 2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Sep. 2023.
- [26] A. Noulas, G. Englebienne, and B. J. A. Krose, "Multimodal speaker diarization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 1, pp. 79–93, Mar. 2011.
- [27] National Institute of Standards and Technology (NIST), "The 2009 (RT-09) rich transcription meeting recognition evaluation plan," Tech. Rep., 2009.
- [28] J. Barker, S. Watanabe, E. Vincent, and J. Trmal, "The fifth 'CHiME' speech separation and recognition challenge: Dataset, task and baselines," in *Proc. of the Interspeech 2018*, Mar. 2018, pp. 1561–1565.
- [29] A. Plaquet and H. Bredin, "Powerset multi-class cross entropy loss for neural speaker diarization," in *Proc. of the Interspeech 2023*, 2023, pp. 3222–3226.
- [30] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," in *Proc. of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2018, pp. 5329–5333.
- [31] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao, J. Wu, L. Zhou, S. Ren, Y. Qian, Y. Qian, J. Wu, M. Zeng, X. Yu, and F. Wei, "WavLM: Large-scale self-supervised pre-training for full stack speech processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1505–1518, Oct. 2022.
- [32] B. Desplanques, J. Thienpondt, and K. Demuynck, "ECAPA-TDNN: Emphasized channel attention, propagation and aggregation in TDNN based speaker verification," in *Proc. of the Interspeech 2020*, ISCA, Oct. 2020.
- [33] N. Dawalatabad, M. Ravanelli, F. Grondin, J. Thienpondt, B. Desplanques, and H. Na, "ECAPA-TDNN embeddings for speaker diarization," *arXiv [eess.AS] 2104.01466*, Apr. 2021.
- [34] T. L. Nguyen, B. T. Ta, D. Van Hai, T. A. X. Tran, and N. M. Le, "Speaker diarization for Vietnamese conversations using deep neural network embeddings," in *Proc. of the 2022 IEEE Ninth International Conference on Communications and Electronics (ICCE)*, Jul. 2022, pp. 300–305.
- [35] M. Ravanelli, T. Parcollet, P. Plantinga, A. Rouhe, S. Cornell, L. Lugosch, C. Subakan, N. Dawalatabad, A. Heba, J. Zhong, J.-C. Chou, S.-L. Yeh, S.-W. Fu, C.-F. Liao, E. Rastorgueva, F. Grondin, W. Aris, H. Na, Y. Gao, R. D. Mori, and Y. Bengio, "SpeechBrain: A general-purpose speech toolkit," *arXiv [eess.AS] 2106.04624*, Jun. 2021.
- [36] A. Défossez, "Hybrid spectrogram and waveform source separation," in *Proc. of the ISMIR 2021 Music Demixing Workshop*, Nov. 2021, pp. 1–11.
- [37] S. Rouard, F. Massa, and A. Défossez, "Hybrid transformers for music source separation," in *Proc. of the 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Jun. 2023, pp. 1–5.
- [38] A. Hotho, A. Nürnberger, and G. Paaß, "A brief survey of text mining," *Journal for Language Technology and Computational Linguistics*, vol. 20, no. 1, pp. 19–62, Jul. 2005.
- [39] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

- [40] S. Horiguchi, Y. Fujita, S. Watanabe, Y. Xue, and K. Nagamatsu, “End-to-end speaker diarization for an unknown number of speakers with encoder-decoder based attractors,” in *Proc. of the Interspeech 2020*, May 2020, pp. 269–273.
- [41] H. Bredin, “Pyannote.Audio 2.1 speaker diarization pipeline: Principle, benchmark, and recipe,” in *Proc. of the Interspeech 2023*, Aug. 2023.

CLASSICAL GUITAR DUET SEPARATION USING GUITAR DUETS - A DATASET OF REAL AND SYNTHESIZED GUITAR RECORDINGS

Marios Glytsos^{1,3} Christos Garoufis^{1,2,3} Athanasia Zlatintsi^{1,2,3} Petros Maragos^{1,3}

¹Robotics Institute, Athena Research Center, Athens, Greece

²Institute of Language and Speech Proc., Athena Research Center, Athens, Greece

³School of ECE, National Technical University of Athens, Athens, Greece

mariosgly@gmail.com, {christos.garoufis,athanasia.zlatintsi}@athenarc.gr, maragos@cs.ntua.gr

ABSTRACT

Recent advancements in music source separation (MSS) have focused in the multi-timbral case, with existing architectures tailored for the separation of distinct instruments, overlooking thus the challenge of separating instruments with similar timbral characteristics. Addressing this gap, our work focuses on monotimbral MSS, specifically within the context of classical guitar duets. To this end, we introduce the GuitarDuets dataset, featuring a combined total of approximately three hours of real and synthesized classical guitar duet recordings, as well as note-level annotations of the synthesized duets. We perform an extensive cross-dataset evaluation by adapting Demucs, a state-of-the-art MSS architecture, to monotimbral source separation. Furthermore, we develop a joint permutation-invariant transcription and separation framework, to exploit note event predictions as auxiliary information. Our results indicate that utilizing both the real and synthesized subsets of GuitarDuets leads to improved separation performance in an independently recorded test set compared to utilizing solely one subset. We also find that while the availability of ground-truth note labels greatly helps the performance of the separation network, the predicted note estimates result only in marginal improvement. Finally, we discuss the behavior of commonly utilized metrics, such as SDR and SI-SDR, in the context of monotimbral MSS.

1. INTRODUCTION

The task of music source separation (MSS) involves dissecting a musical composition into its constituent sources, typically segregating individual instruments or vocal tracks from a composite audio mixture [1, 2, 3]. Due to the multitude of the co-playing sources, as well as its utility in a variety of applications [1], MSS stands as a significant challenge in the field of Music Information Re-

trieval (MIR) [4]. The majority of research efforts have focused on multi-timbral music source separation [5, 6, 7]. In this case, the goal is the separation of distinct instrumental sources from a mixture, where the sources belong to different instrument families or types such as vocals, bass, drums and others, and as such can be framed as an extension of the task of speech denoising into the music domain [8].

Through the advancement of digital signal processing [9] and deep learning [5, 10], considerable progress has been made in extracting distinct instrumental tracks from complex musical compositions. Most recent, deep-learning based approaches for MSS are divided into spectrogram-domain approaches [11], waveform-domain methodologies [10, 12, 13, 14, 15] and hybrid ones, working simultaneously in both domains [5]. Spectrogram-domain methods typically isolate sources via mask prediction [16], waveform-domain approaches enhance source separation by applying spectrogram techniques in a learned latent space [14, 17], or directly predicting the isolated waveforms [12], with the additional advantage of incorporating phase information, while hybrid architectures [5] leverage the strengths of both. Moreover, recent findings have highlighted the benefits of using static or dynamic activity labels [18, 19, 20, 21, 22], as well as jointly training transcription and source separation modules [23, 24], which enhances task performance, paralleling efforts in simultaneous speech recognition and separation training [25].

However, an area that remains relatively underexplored is monotimbral music source separation. This subfield of MSS focuses on extracting audio components that belong to the same instrument family, or different builds of the same instrument. It can be viewed as the counterpart of the speaker separation problem [31] in the music domain. While this similarity has led to the development of similar methodologies for network training [32], speaker separation, especially within the context of, rarely available in MSS datasets, multi-microphone recordings [33], can also rely on spatial cues. The limited exploration in this area can largely be attributed to the historical focus on isolating the most prominent instruments in popular music, while the demand for separation of instruments with close timbral characteristics is less pronounced. Indeed, there are very few datasets suitable for training algorithms on the task of separating instrumental tracks from the same



© M. Glytsos, C. Garoufis, and A. Zlatintsi and P. Maragos. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** M. Glytsos, C. Garoufis, and A. Zlatintsi and P. Maragos, “Classical Guitar Duet Separation using GuitarDuets - a Dataset of Real and Synthesized Guitar Recordings”, in *Proc. of the 25th Int. Society for Music Information Retrieval Conf.*, San Francisco, United States, 2024.

Datasets	Real Data Incl.	Monotimbral	Polyphonic	Note Annotations	Duration
musdb18 [7]	✓	✗	✓	✗	ca. 10h
MoisesDB [26]	✓	✗	✓	✗	ca. 14.5h
URMP [27]	✓	✗	✓	✓	1h 6min
SLAKH [28]	✗	✗	✓	✓	ca. 145h
EnsembleSet [29]	✗	✓	✗	✓	6h 9min
GuitarSet [30]	✓	✓	✓	✓	3h 3min
GuitarDuets	✓	✓	✓	Partial	2h 44min

Table 1: Comparison of the GuitarDuets dataset with existing datasets in the literature for music source separation; we note that GuitarSet is strictly monotimbral, since it was entirely recorded using one guitar.

	GuitarDuets(R)	GuitarDuets(S)
# Tracks	34	35
Dur./Track (mins)	1.72 ±1.35	3.03 ±2.86
Total Dur. (mins)	58	106
Notes/sec.	-	7

Table 2: Detailed statistics of the real and synthesized subsets of the GuitarDuets dataset; note statistics are included for the synthetic subset only.

instrument family in a polyphonic context [30], with the majority of publicly available datasets covering the case of separating mixtures of multiple singing voices [34, 35, 36].

In this paper, we attempt to bridge this gap by introducing GuitarDuets¹, a dataset consisting of a total of ca. 3 hours of real and synthesized guitar duet recordings, along with partial note-level annotations, which can be leveraged as auxiliary score information. We benchmark GuitarDuets in the tasks of i) unconditional guitar duet separation and ii) score-informed duet separation, using the hybrid Demucs [5] as our separation model. We also examine the possibility of integrating note-level predictions into a joint transcription and separation framework. In more detail, the main contributions of this work are:

- Introduction of GuitarDuets, a dataset for monotimbral music source separation, featuring both real classical guitar duet recordings and synthetic recordings generated from online transcriptions and virtual instruments. The synthetic portion includes MIDI representations for each guitar part, enriching the dataset for algorithm training and detailed analysis.
- Extensive cross-dataset evaluation across various conditions, including real and generated synthetic data, as well as the existence or absence of auxiliary score information in specific Demucs branches.
- Development of a joint transcription and separation framework, which incorporates transcription predictions, by adapting existing architectures [5, 37] to the task of monotimbral source separation with the introduction of a permutation-invariant [32] loss. We show that incorporation of these note-level predictions can improve the separation of real guitar duets.
- Finally, we analyze the behavior of commonly-utilized source separation metrics in the context of classical guitar duets to understand their effectiveness when applied in sources with similar timbres.

¹ The dataset is available at: <https://zenodo.org/records/12802440>

2. DATASETS

2.1 Existing Datasets

Datasets available for music source separation or transcription are primarily divided into multitimbral and monotimbral ones, each offering instrument-specific tracks or stems, often accompanied by transcriptions. Multitimbral datasets such as musdb18 [7], URMP [27], MedleyDB [38], MoisesDB [26] and SLAKH [28] are most prominent, featuring both real and synthesized data from a broad spectrum of instruments; some extend to multimodal forms, including for instance audiovisual elements [27].

In contrast, monotimbral instrumental datasets, notably fewer in number, include focused collections such as GuitarSet [30] and EnsembleSet [29]. GuitarSet provides detailed annotations for acoustic guitar recordings, consisting of pairs of comping and soloing performances, while EnsembleSet targets chamber ensembles with high-quality synthetic reproductions of classical music. Despite their utility, these monotimbral datasets face some limitations, namely: GuitarSet’s structure, with distinct solo and accompaniment parts, oversimplifies the separation task due to the distinct role of each guitar. Also, the lack of timbral differences between the two guitars prevents the networks from focusing on timbral cues for the separation task. On the other hand, EnsembleSet’s reliance on synthetic data introduces a realism gap, underscoring the need for datasets that more accurately capture the dynamics of live musical performances. Moreover, the instruments it contains are largely monophonic, which hinders its use for scenarios with polyphonic co-playing instruments.

2.2 The GuitarDuets Dataset

In this section, we will describe the GuitarDuets dataset, comprising both real recordings of classical guitar duets and synthesized recordings, leveraging virtual instruments and MIDI scores. This approach aimed to provide an original and realistic set of guitar duet recordings for training and evaluating deep learning algorithms on monotimbral MSS, while simultaneously overcoming their limited duration, granting ample training data and enabling analysis between real and synthetic datasets. In total, our dataset comprises 58.6 minutes of real data and 106 minutes of synthesized data, amounting to 164.6 minutes overall. A comparison of the GuitarDuets with the most prominent datasets for MSS in the literature is outlined in Table 1,

whereas detailed statistics about both the real and synthesized subsets of GuitarDuets are given in Table 2.

Real Recordings: For the recordings featuring real classical guitars, we utilized a quiet, acoustically treated room and high-quality condenser microphones (Presonus PM-2), one for each guitar. During the recording process, four different classical guitars were used, with some tracks (16 min.) replayed using different guitars to further enhance timbral diversity. Simultaneous play was crucial for capturing the musical interplay between the two guitarists. The whole recording process resulted in the recording of 27 guitar duets (per-track duration: 123 ± 82 sec.), mostly from the Modern Classical and Nuevo Tango genres and from the Romantic Period. This approach, while essential for the integrity and realism of the dataset, introduced a challenge with cross-microphone sound bleeding. This crossover of sound presented a significant concern, as it compromises the isolation of the individual guitar tracks, impacting the quality of the dataset. In addressing the issue of source bleeding in microphones, we recorded a specialized test set that is free from such leakage. This set, consisting of 7 tracks (per-track duration: 39 ± 13 sec.), was created to ensure the absence of cross-feed between microphones. Each guitar track was exported as a 44,100 Hz, 16-bit WAV file in stereo format, with mixed audio files created by averaging individual guitar performances.

Synthetic Recordings: Despite the inherent realism of the real recordings, their small duration could prove problematic for network training, whereas the single recording setup utilized could import biases. A commonly utilized shortcut to increase the duration of real recordings is to virtually augment them, by synthesizing additional pieces based on note-level transcriptions and virtual music instruments, which has proven effective not only in generating multitrack datasets [28, 29], but also in tasks such as tablature generation [39, 40]. In our case, “Session Guitarist - Picked Nylon”², a sample-based virtual instrument, was utilized to generate classical guitar sounds. It offers a wide range of playing styles, capturing the nuances of nylon-stringed guitars. We selected guitar duet MIDI scores from the MuseScore community³, representing a broad spectrum of pieces. Logic Pro X⁴ served as the digital audio workstation (DAW) for transforming MIDI scores into realistic guitar performances. By configuring multiple instances of the PICKED NYLON plugin with distinct timbral settings, we produced different guitar sounds. The final dataset was exported as 44,100 Hz, 16-bit WAV files in both stereo and mono formats for mixed audio file creation.

3. METHODOLOGY

3.1 Separation Architecture

In this work the Hybrid Transformer Demucs [5] was used as the separation backbone, consisting of dual U-Nets [16], operating in both time and spectrogram domains, each

² <https://www.native-instruments.com/en/products/komplete/guitar/session-guitarist-picked-nylon/>

³ <https://musescore.com/>

⁴ <https://www.apple.com/logic-pro/>

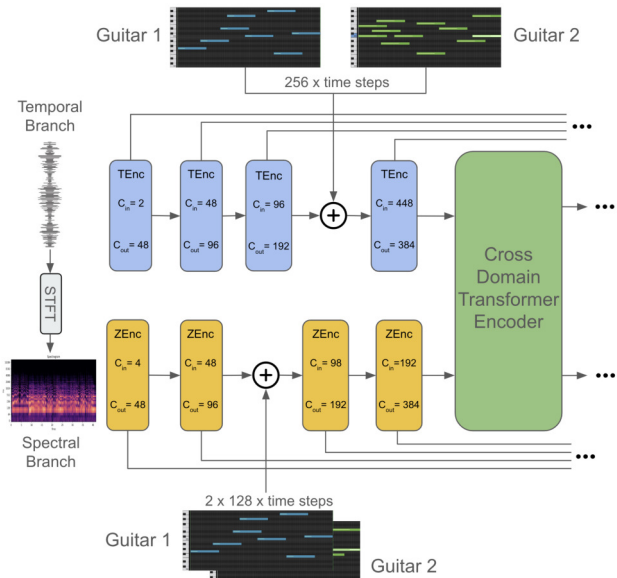


Figure 1: Overview of the incorporation of note-level annotations into the Demucs Architecture.

featuring four encoder and decoder layers. The temporal encoder (TEncoder) downsamples the input waveform through a series of 1D convolutions, whereas the spectral encoder (ZEncoder) gradually compresses the STFT magnitude of the input by applying convolutions across the spectral axis. The traditional convolutional layers, positioned between the encoder and decoder in previous iterations of the Demucs architecture [41] are replaced with a cross-domain Transformer Encoder, composed of interleaved self-attention and cross-attention Encoder layers, each equipped with Layer Scale [42]. The attention mechanism operates with eight heads, and the hidden state size of the feedforward network is four times the dimension of the transformer. The primary decoder layer is shared, branching into both temporal and spectral domains, with the respective decoders built symmetrically to the encoders. The spectral output, post an inverse Short Time Fourier Transform (ISTFT), is merged with the temporal output, producing the model’s prediction. We note that in our experiments, the input length is set to 4 seconds.

3.2 Score-Informed Separation

In the context of Score-Informed Separation, the separation network is conditioned on the note-level transcripts of the recordings. To this end, binary vectors indicating the presence or absence of each of the 128 MIDI notes during small temporal frames are concatenated with the intermediate feature maps in each branch, as indicated in Fig. 1. In particular, in the temporal branch, the activity labels of each guitar are inserted after the third TEncoder layer. The binary vector for each guitar has a dimensionality of $128 \times N_s$, where N_s corresponds to the number of samples for each 4-second segment, yielding a combined shape of $256 \times N_s$ for both guitars. Thus, the activity labels have to be downsampled across the temporal axis, to match the resolution of the encoder at this stage. In a parallel manner, within the frequency branch, these binary vectors are introduced following the second ZEncoder

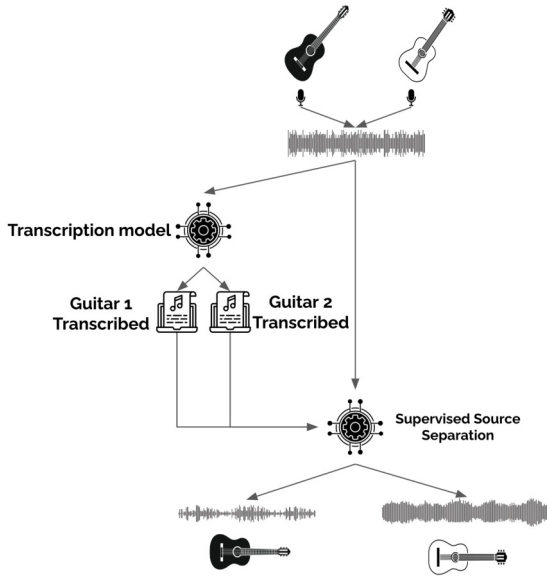


Figure 2: Overview of the proposed methodology for joint transcription and separation of guitar duets.

layer. The shape of the activity labels for concatenation is $2 \times 128 \times N_s$, aligning with the two guitars’ MIDI notes. In this case, the activity labels are concatenated with the feature maps across the channel dimension; since the frequency resolution of the ZEncoder, at this stage, matches the number of MIDI notes, resampling occurs only across the temporal axis, by downsampling the note activity labels to the respective temporal resolution of the ZEncoder.

If transcriptions are not available, we use a separate transcription network to generate them creating a joint transcription-separation framework. The first network would intake the combined sounds of the two guitars and generate a binarized piano roll representation for each individual guitar. Afterward, the second model combines the mixed audio and the generated piano rolls to create separate estimates for each guitar as depicted in Figure 2. From a musical endpoint the transcription network could potentially capture note interdependencies and guitar duet patterns through binarized vector features, aiding in note prediction. This transcription informs the separation model, which refines the output by focusing on timbre.

For the transcription architecture, we utilize the Residual Shuffle-Exchange Network (RSE) [37], which has achieved state-of-the-art results in MusicNet [43]. This network enhances the neural Shuffle-Exchange network [44] by employing both Switch and Shuffle layers to capture sequence dependencies effectively, as well as reducing its computational overhead by incorporating strided convolutions. For further details about the architecture we refer to [37, 44]. In our implementation, the RSE’s output layer is modified to produce a binarized 2×128 -dimensional representation, to assign activity labels for each of the 128 MIDI notes to the corresponding instrument.

4. EXPERIMENTAL EVALUATION

4.1 Experimental Setup

For the separation experiments, we used both the real and synthesized subsets of GuitarDuets, which we will further denote as GuitarDuets(R) and GuitarDuets(S), respectively, as well as the GuitarSet, for which mixtures were generated via addition of the available comping and solo excerpts. We adapted the backbone Demucs model for classical guitar duet separation, modifying it to output two stereo signals, one for each guitar. Data augmentation techniques including channel swapping, time cropping, amplitude scaling and remixing individual guitar parts from different performances were employed during network training to improve generalization. As our loss function we used the quantity:

$$\alpha \cdot \min(|\hat{g}_1 - g_1| + |\hat{g}_2 - g_2|, |\hat{g}_2 - g_1| + |\hat{g}_1 - g_2|) + \beta \cdot |(\hat{g}_1 + \hat{g}_2) - (g_1 + g_2)|, \quad (1)$$

where the first term corresponds to the traditional permutation-invariant L1 loss between the ground truth signals g_1, g_2 and the output sources \hat{g}_1, \hat{g}_2 , and the second term models the similarity between the sum of the guitar estimates and the input mixtures, encouraging the network to provide separate guitar tracks which neither discard nor duplicate note instances, whereas the weight values were set, after preliminary experiments, to $\alpha = 0.8, \beta = 0.2$.

For the transcription architecture experiments, we employed GuitarSet and the GuitarDuets(S) dataset, which contain note-level annotations for individual guitar parts. We transformed labels from GuitarSet (.jams files) and the MIDI files from our dataset to CSV format. All audio files, initially sampled at 44,100 Hz, were resampled to 11,000 Hz and converted to mono, to render them compatible with the RSE backbone [37]. Similar to the separation case, the loss function—in this case, the binary cross entropy—was employed within a permutation invariant framework.

4.2 Cross-Dataset Analysis

For the purposes of the cross-dataset analysis, we consider the GuitarDuets(R) and GuitarDuets(S) subsets as separate datasets, and train the Demucs backbone on various combinations of GuitarSet, GuitarDuets(R) and GuitarDuets(S), using the same experimental protocol and an 80-20 training-validation split; all networks are evaluated on the bleeding-free testing set of GuitarDuets(R). Upon inspection of the results, presented in Table 3, several key insights emerge. Namely, the complete GuitarDuets dataset yielded the highest SDR values for the first guitar. The inclusion of the synthesized subset likely provided additional information that enhanced the model’s performance with regards to the SDR. On the other hand, the inclusion of these synthetic parts made the model prone to auditory artifacts, since the highest SAR scores were achieved for the combination of GuitarDuets(R) with the GuitarSet. Finally, we observe that the combination of the complete GuitarDuets dataset with GuitarSet leads in diminished performance, probably due to the structural differences between the training subsets.

Source Datasets			Metrics			
GuitarDuets(R)	GuitarDuets(S)	GuitarSet	SDR	SI-SDR	SAR	SIR
✓	✓	✓	G1: 4.297 G2: 0.835	G1: 3.403 G2: -2.880	G1: 7.670 G2: 2.062	G1: 10.766 G2: 4.495
✓	✗	✓	G1: 4.522 G2: 1.359	G1: 4.280 G2: -2.238	G1: 9.483 G2: 10.898	G1: 6.273 G2: 7.631
✗	✓	✓	G1: 4.493 G2: 1.137	G1: 1.530 G2: -1.566	G1: 8.191 G2: 8.305	G1: 7.038 G2: 8.081
✗	✗	✓	G1: 4.632 G2: 1.378	G1: 3.871 G2: -1.198	G1: 7.971 G2: 6.332	G1: 7.321 G2: 8.968
✗	✓	✗	G1: 3.472 G2: 0.200	G1: 1.857 G2: -4.052	G1: 8.212 G2: 4.502	G1: 8.217 G2: 4.501
✓	✗	✗	G1: 4.952 G2: 1.014	G1: 3.573 G2: -3.536	G1: 7.628 G2: 1.424	G1: 10.413 G2: 4.873
✓	✓	✗	G1: 5.882 G2: 0.920	G1: 4.315 G2: -3.133	G1: 8.488 G2: 0.896	G1: 11.706 G2: 4.104

Table 3: Separation results on the testing set of GuitarDuets(R), according to the datasets utilized during training. G1 corresponds to the 1st guitar, G2 to the 2nd. Higher is better for all metrics.

Dataset	Note Labels	Branch Conditioning		Metrics			
		Time	Frequency	SDR	SI-SDR	SAR	SIR
GuitarDuets(S)	Ground Truth	✓	✗	G1: 4.453 G2: 4.355	G1: 3.117 G2: 0.072	G1: 4.972 G2: 3.197	G1: 12.411 G2: 8.292
		✗	✓	G1: 4.547 G2: 3.301	G1: 3.293 G2: -0.410	G1: 4.685 G2: 3.451	G1: 9.523 G2: 9.882
		✓	✓	G1: 4.717 G2: 4.863	G1: 3.378 G2: 0.154	G1: 4.362 G2: 4.316	G1: 12.081 G2: 10.537
GuitarDuets(S)	Estimated	✓	✓	G1: 3.414 G2: 1.977	G1: 1.398 G2: -1.511	G1: 3.455 G2: 3.087	G1: 10.655 G2: 7.035
	None	✗	✗	G1: 2.575 G2: 2.569	G1: 2.436 G2: -2.514	G1: 4.473 G2: 3.473	G1: 12.795 G2: 5.717
GuitarDuets(R)	Estimated	✓	✓	G1: 5.313 G2: 1.035	G1: 4.352 G2: -3.291	G1: 7.638 G2: 1.998	G1: 11.110 G2: 5.089
	None	✗	✗	G1: 4.952 G2: 1.014	G1: 3.573 G2: -3.536	G1: 7.628 G2: 1.424	G1: 10.413 G2: 4.873

Table 4: Separation results on the testing sets of GuitarDuets(S), GuitarDuets(R), when using solely the respective training sets for training, depending on the availability of note-level annotations and the Demucs branches conditioned on them. G1 corresponds to the 1st guitar, G2 to the 2nd. Higher is better for all metrics.

In our analysis, we observed a consistent discrepancy in the Signal-to-Distortion Ratio (SDR) between the two guitars, where the first guitar exhibited a decent SDR, while the second often fell below a threshold of 1 dB. This pattern suggests that the algorithm may be effectively separating the first guitar by identifying it as the primary source, whereas it perceives the second guitar as background noise, or merely an auditory artifact. It is important to note that the average amplitude of both guitars is on the same scale, so this observation is not attributed to amplitude differences. Notably, we observe that the most consistent SDR values for G2 were achieved when GuitarSet was included in the training set, which we attribute to its relatively noise-free structure.

4.3 Score-Informed Separation Approaches

For the experiments concerning score-informed separation, we investigated the integration of activity labels into our network, considering Demucs’ operation across frequency

and temporal domains, by using the GuitarDuets(S) as our dataset since it contains note-level annotations. We also investigate, using both GuitarDuets(R) and GuitarDuets(S), whether the joint transcription-separation architecture can aid in effective separation in scenarios where no ground truth data is available. In both cases, a part of the dataset (the bleeding-free subset of GuitarDuets(R), and 10% of GuitarDuets(S)) was used for performance evaluation; the rest were used for training and validation, at an 80:20 ratio.

The analysis, as detailed in Table 4, reveals that while using the temporal branch for note label integration leads to slightly improved results compared to the spectral branch, the hybrid approach achieves the most promising outcomes. This performance can be attributed to the inherent design of the Demucs architecture, which has historically shown improved efficiency when leveraging both domains concurrently [41]. It is also noteworthy that while the integration of ground-truth labels leads to higher SIR values, presumably due to the guidance that these labels

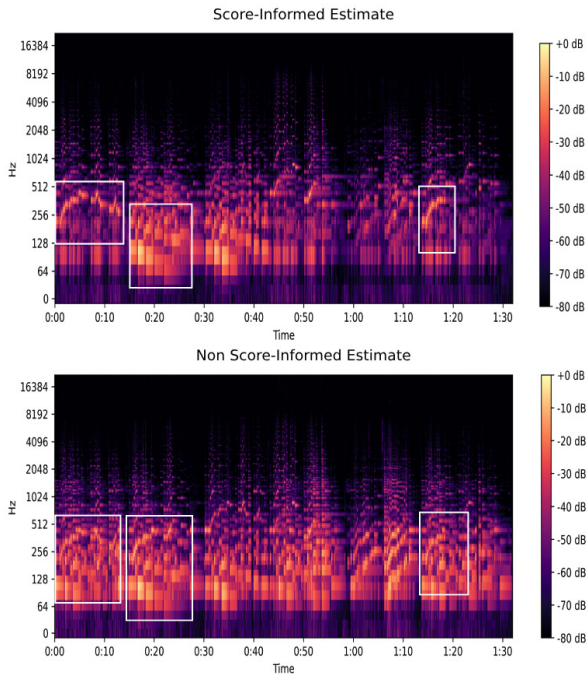


Figure 3: Comparison of spectrogram estimates with estimated (top) note-level annotations, and without those (bottom), for an instance from GuitarDuets (R).

provide to the separation network about the identity of each guitar, the improvement in SAR is marginal.

Regarding the joint transcription and separation framework, the performance does not reach the levels achieved when ground-truth note-level annotations are available, as measured in GuitarDuets(S). On the other hand, while in the case of GuitarDuets(R), the performance is slightly improved when these pseudo-annotations are available, GuitarDuets(S) achieves better results in their absence. A comparison of guitar estimates for the models trained with and without predicted note label information, for an instance of the GuitarDuets(R) test set, can be depicted in Figure 3; we assume that the incorporation of note activity labels enables the separation model to more accurately sustain notes, enhancing the quality of the isolated melodic and accompanying parts. On the other hand, we attribute the performance drop, when using GuitarDuets(S), to the reduced generalization of the transcription network. Since the training set of GuitarDuets(S) was used for its training, the separation network was trained using mostly correct labels, but evaluated with note-level annotations of pieces the transcription network did not use for training.

4.4 Comparative Metric Analysis

In the field of MSS, the evaluation of separation quality is often quantified using metrics such as SDR [45] and SI-SDR [46]. While they have been extensively used in studies focusing on separating different instruments, their behavior on sources with similar timbral characteristics remains less explored. Given that most prior work involves instruments with distinct timbres, direct comparison of our results with SDR values achieved across different datasets may not be appropriate for our study, which focuses on two classical guitars with similar timbral properties.

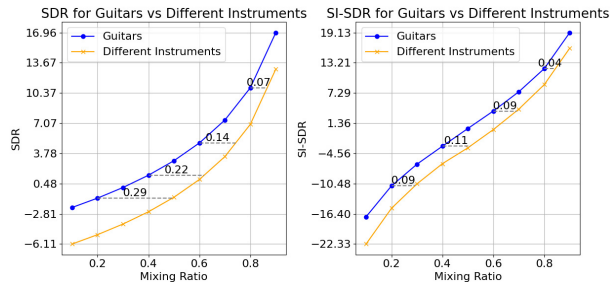


Figure 4: Comparison of the behavior of SDR (left) and SI-SDR (right) when assessing the separation of monotimbral (blue line) or multitimbral (orange line) duets.

In order to identify potential disparities in the behavior of the metrics that can be attributed to timbral similarities in the mixture components, we simulated imperfect estimates of a reference signal x_1 by creating additive synthetic mixtures of the signals x_1, x_2 as:

$$m = \alpha x_1 + (1 - \alpha)x_2, \quad (2)$$

with $\alpha \in (0, 1)$, and measured the values of the SDR, SI-SDR metrics between these mixtures and x_1 . We examined two cases using signals derived from Track 29 of the GuitarDuets(S): i) a monotimbral mixture, where both x_1 and x_2 constitute guitar signals, and ii) a multitimbral mixture, where x_2 was synthesized from the second guitar’s notes using a piano virtual instrument plugin. To guarantee a fair comparison across all tests, we performed amplitude normalization between the two tracks for each experiment.

The results, displayed in Figure 4, indicate that both metrics for the guitar mixtures are consistently higher than those obtained from mixtures of different instruments. For instance, the mixing ratio α required to reach an SDR value of 5 approaches 0.8 for the multi-timbral case, while 0.6 for the mono-timbral case. This suggests that the timbral similarity between the two guitars introduces a challenge for the metrics to accurately assess the quality of separation.

5. CONCLUSIONS

In this paper, we introduced GuitarDuets, a dataset consisting of both real and synthesized classical guitar duets. We exhibit that our dataset can be utilized for developing monotimbral source separation algorithms within both traditional and score-informed frameworks. We further developed a joint permutation-invariant framework for transcription and separation of monotimbral mixtures, which we show that can lead to improved performance in separation of real guitar duets. In the future, we plan to extend the recordings of both the real and synthesized subsets of GuitarDuets, and provide note-level annotations for its real subset. Furthermore, regarding the joint transcription-separation architecture, we intend to explore more sophisticated ways for integrating the predicted guitar transcripts into the separator [47, 48]. Finally, we aim to conduct extensive listening tests, which will help in further shedding light into both the performance of the various approaches we compare, and the significance of objective metrics within the context of monotimbral audio source separation.

Acknowledgments: This research was supported by the Hellenic Foundation for Research and Innovation (H.F.R.I.) under the “3rd Call for H.F.R.I. Research Projects to support Post-Doctoral Researchers” (Project Acronym: i-MRePlay, Project Number: 7773).

6. REFERENCES

- [1] R. Liu and S. Li, “A review on music source separation,” in *Proc. IEEE Youth Conf. on Information, Computing and Telecommunication*, 2009.
- [2] Y. Mitsufuji, G. Fabbro, S. Uhlich, F.-R. Stöter, A. Défossez, M. Kim, W. Choi, C. Y. Yu, and K. W. Cheuk, “Music demixing challenge 2021,” *Frontiers in Signal Processing*, vol. 1, Jan., 2022.
- [3] E. Cano, D. FitzGerald, A. Liutkus, M. D. Plumbley, and F.-R. Stöter, “Musical source separation: An introduction,” *IEEE Signal Processing Magazine*, vol. 36, no. 1, pp. 31–40, 2019.
- [4] Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimilakis, D. FitzGerald, and B. Pardo, “An overview of lead and accompaniment separation in music,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 8, pp. 1307–1335, 2018.
- [5] S. Rouard, F. Massa, and A. Défossez, “Hybrid transformers for music source separation,” in *Proc. Int’l Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2023.
- [6] F.-R. Stöter, S. Uhlich, A. Liutkus, and Y. Mitsufuji, “Open-unmix-a reference implementation for music source separation,” *Journal of Open Source Software*, vol. 4, no. 41, p. 1667, 2019.
- [7] Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimilakis, and R. Bittner, “Musdb18-a corpus for music separation,” 2017.
- [8] E. Vincent, T. Virtanen, and S. Gannot, *Audio Source Separation and Speech Enhancement*. John Wiley & Sons, 2018.
- [9] J.-L. Durrieu, A. Ozerov, C. Févotte, G. Richard, and B. David, “Main instrument separation from stereophonic audio signals using a source/filter model,” in *Proc. 17th European Signal Processing Conf.*, 2009.
- [10] A. Défossez, N. Usunier, L. Bottou, and F. R. Bach, “Music source separation in the waveform domain,” 2019. [Online]. Available: <http://arxiv.org/abs/1911.13254>
- [11] Y. Luo and J. Yu, “Music source separation with band-split RNN,” *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, vol. 31, pp. 1893–1901, 2023.
- [12] D. Stoller, S. Ewert, and S. Dixon, “Wave-u-net: A multi-scale neural network for end-to-end audio source separation,” in *Proc. Int’l Society for Music Information Retrieval Conf. (ISMIR)*, 2018.
- [13] W. Zai El Amri, O. Tautz, H. Ritter, and A. Melnik, “Transfer learning with jukebox for music source separation,” in *Proc. Int’l Conf. on Artificial Intelligence Applications and Innovation*, 2022.
- [14] Y. Luo and N. Mesgarani, “Conv-TasNet: Surpassing ideal time–frequency magnitude masking for speech separation,” *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, vol. 27, no. 8, p. 1256–1266, Aug. 2019.
- [15] C. Garoufis, A. Zlatintsi, and P. Maragos, “HTMD-Net: A Hybrid Masking-Denoising Approach to Time-Domain Monaural Singing Voice Separation,” in *Proc. 29th European Signal Processing Conf. (EUSIPCO)*, 2021.
- [16] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional networks for biomedical image segmentation,” in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015.
- [17] D. Samuel, A. Ganeshan, and J. Naradowsky, “Meta-learning extractors for music source separation,” in *Proc. Int’l Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.
- [18] O. Slizovskaia, L. Kim, G. Haro, and E. Gomez, “End-to-end sound source separation conditioned on instrument labels,” in *Proc. Int’l Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2019.
- [19] D. Petermann, P. Chandna, H. Cuesta, J. Bonada, and E. Gómez, “Deep learning based source separation applied to choir ensembles,” *Proc. Int’l Society for Music Information Retrieval Conf. (ISMIR)*, 2020.
- [20] M. Schwabe and M. Heizmann, “Improved separation of polyphonic chamber music signals by integrating instrument activity labels,” *IEEE Access*, vol. 11, pp. 42 999–43 007, 2023.
- [21] M. Miron, J. Janer, and E. Gómez, “Monaural score-informed source separation for classical music using convolutional neural networks,” in *Proc. Int’l Society for Music Information Retrieval Conf. (ISMIR)*, 2017.
- [22] M. Gover and P. Depalle, “Score-informed source separation of choral music,” in *Proc. Int’l Society for Music Information Retrieval Conf. (ISMIR)*, 2020.
- [23] L. Lin, Q. Kong, J. Jiang, and G. G. Xia, “A unified model for zero-shot music source separation, transcription and synthesis,” in *Proc. Int’l Society for Music Information Retrieval Conf. (ISMIR)*, 2021.
- [24] K. W. Cheuk *et al.*, “Jointist: Simultaneous improvement of multi-instrument transcription and music source separation via joint training,” 2023. [Online]. Available: <https://arxiv.org/pdf/2206.10805>
- [25] J. Shi, X. Chang, S. Watanabe, and B. Xu, “Train from scratch: Single-stage joint training of speech separation and recognition,” *Computer Speech & Language*, vol. 76, p. 101387, Apr., 2022.

- [26] I. Pereira, F. Araújo, F. Korzeniowski, and R. Vogl, “Moisesdb: A dataset for source separation beyond 4-stems,” in *Proc. Int’l Conf. of International Society for Music Information Retrieval (ISMIR)*, 2023.
- [27] B. Li *et al.*, “Creating a multitrack classical music performance dataset for multimodal music analysis: Challenges, insights, and applications,” *IEEE Trans. on Multimedia*, vol. 21, pp. 522–535, 2018.
- [28] E. Manilow *et al.*, “Cutting music source separation some slakh: A dataset to study the impact of training data quality and quantity,” in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2019.
- [29] S. Sarkar, E. Benetos, and M. Sandler, “EnsembleSet: A new high-quality synthesised dataset for chamber ensemble separation,” in *Proc. Int’l Society for Music Information Retrieval Conf. (ISMIR)*, 2022.
- [30] Q. Xi, R. M. Bittner, J. Pauwels, X. Ye, and J. P. Bello, “Guitarset: A dataset for guitar transcription,” in *Proc. Int’l Society for Music Information Retrieval Conf. (ISMIR)*, 2018.
- [31] E. C. Cherry, “Some experiments on the recognition of speech, with one and with two ears,” *Journal of the Acoustical Society of America*, vol. 25, pp. 975–979, 1953.
- [32] D. Yu, M. Kolbæk, Z.-H. Tan, and J. Jensen, “Permutation invariant training of deep models for speaker-independent multi-talker speech separation,” in *Proc. Int’l Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.
- [33] T. Yoshioka, H. Erdogan, Z. Chen, and F. Alleva, “Multi-microphone neural speech separation for far-field multi-talker speech recognition,” in *Proc. Int’l Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.
- [34] S. Sarkar, E. Benetos, and M. B. Sandler, “Vocal harmony separation using time-domain neural networks,” in *Proc. Interspeech Conf.*, 2021.
- [35] C.-B. Jeon, H. Moon, K. Choi, B. S. Chon, and K. Lee, “Medleyvox: An evaluation dataset for multiple singing voices separation,” in *Proc. Int’l Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2023.
- [36] R. Schramm and E. Benetos, “Automatic transcription of a cappella recordings from multiple singers,” in *AES International Conference Semantic Audio*, 2017.
- [37] A. Draguns, E. Ozolins, A. Sostaks, M. Apinis, and K. Freivalds, “Residual shuffle-exchange networks for fast processing of long sequences,” in *Proc. AAAI Conf. on Artificial Intelligence*, 2020.
- [38] R. Bittner, J. Salamon, M. Tierney, M. Mauch, C. Cannam, and J. Bello, “Medleydb: A multitrack dataset for annotation-intensive mir research,” in *Proc. Int’l Society for Music Information Retrieval Conf. (ISMIR)*, 2014.
- [39] Y. Zang, Y. Zhong, F. Cwitkowitz, and Z. Duan, “Synthtab: Leveraging synthesized data for guitar tablature transcription,” in *Proc. Int’l Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 1286–1290.
- [40] H. Pedroza, W. Abreu, R. Corey, and I. Roman, “Leveraging real electric guitar tones and effects to improve robustness in guitar tablature transcription modeling,” 2024. [Online]. Available: <https://arxiv.org/abs/2405.14679>
- [41] A. Défossez, “Hybrid spectrogram and waveform source separation,” in *Proc. Music Demixing Workshop (MDX)*, 2021.
- [42] H. Touvron, M. Cord, A. Sablayrolles, G. Synnaeve, and H. Jégou, “Going deeper with image Transformers,” in *Proc. Int’l Conf. on Computer Vision (ICCV)*, 2021.
- [43] J. Thickstun, Z. Harchaoui, and S. Kakade, “Learning features of music from scratch,” in *Proc. Int’l Conf. on Learning Representations (ICLR)*, 2017.
- [44] K. Freivalds, E. Ozolins, and A. Sostaks, “Neural Shuffle-Exchange Networks - Sequence processing in $O(n \log n)$ time,” in *Advances in Neural Information Proc. Systems (NeurIPS)*, 2019.
- [45] E. Vincent, R. Gribonval, and C. Févotte, “Performance measurement in blind audio source separation,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 4, pp. 1462–1469, 2006.
- [46] J. Le Roux, S. Wisdom, H. Erdogan, and J. R. Hershey, “SDR—half-baked or well done?” in *Proc. Int’l Conf. on Acoustics Speech and Signal Processing (ICASSP)*, 2019.
- [47] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville, “Film: Visual reasoning with a general conditioning layer,” in *Proc. AAAI Conf. on Artificial Intelligence*, 2018.
- [48] G. Meseguer-Brocal and G. Peeters, “Conditioned-unet: Introducing a control mechanism in the u-net for multiple source separations,” in *Proc. Int’l Society for Music Information Retrieval Conf. (ISMIR)*, 2019.

CAN LLMs "REASON" IN MUSIC? AN EVALUATION OF LLMs' CAPABILITY OF MUSIC UNDERSTANDING AND GENERATION

Ziya Zhou^{1,2} Yuhang Wu² Zhiyue Wu³
Xinyue Zhang² Ruibin Yuan^{1,2} Yinghao Ma^{2,4}
Lu Wang³ Emmanouil Benetos⁴ Wei Xue¹ Yike Guo¹
¹ AIS, The Hong Kong University of Science and Technology
² Multimodal Art Projection
³ Shenzhen University
⁴ C4DM, Queen Mary University of London

zzhoucp@connect.ust.hk, weixue@ust.hk, yikeguo@ust.hk

ABSTRACT

Symbolic Music, akin to language, can be encoded in discrete symbols. Recent research has extended the application of large language models (LLMs) such as GPT-4 and Llama2 to the symbolic music domain including understanding and generation. Yet scant research explores the details of how these LLMs perform on advanced music understanding and conditioned generation, especially from the multi-step reasoning perspective, which is a critical aspect in the conditioned, editable, and interactive human-computer co-creation process. This study conducts a thorough investigation of LLMs' capability and limitations in symbolic music processing. We identify that current LLMs exhibit poor performance in song-level multi-step music reasoning, and typically fail to leverage learned music knowledge when addressing complex musical tasks. An analysis of LLMs' responses highlights distinctly their pros and cons. Our findings suggest achieving advanced musical capability is not intrinsically obtained by LLMs, and future research should focus more on bridging the gap between music knowledge and reasoning, to improve the co-creation experience for musicians.

1. INTRODUCTION

Large language models (LLMs), such as GPT-4, harness the power of deep learning to produce human-like text. These models, trained on vast datasets of textual content, have notably propelled advancements in natural language processing (NLP). They excel in complex language understanding and generation tasks including translation, sentiment analysis, question answering, and summarization, showcasing their reasoning capability with sophistication.

Large language models (LLMs), initially pre-trained on

extensive textual corpora, can assimilate general linguistic patterns and structures. They are subsequently fine-tuned with domain-specific data, such as code and mathematical symbols, to enhance the adaptation to specific tasks. This refinement allows LLMs' proficiency to more accurately manage domain-specific terminology and complicated challenges like multi-step reasoning. Music Reasoning refers to the ability to estimate the varying harmonies, keys, rhythms, and other musical elements that are not explicitly annotated in a piece of music and are significant for music themes, progression, and styles [1]. The analogy between the reasoning process in music and mathematics suggests their structural similarities. Both disciplines fundamentally rely on patterns: music in rhythms, scales, and chord progressions, while mathematics involves sequences, symmetries, and geometric configuration. Moreover, music theory utilizes mathematical concepts to articulate intervals between pitches, chord structures, and the rhythmic temporal division [2,3], underscoring the intrinsic reasoning nature of the musical components.

Music can be represented as sequences of symbols such as MIDI or ABC notation, rendering it suitable for processing by LLMs, which excel in long-context understanding and multi-step reasoning. These models are capable to dissect and generate intricate musical patterns encompassing melodic, harmonic, and rhythmic structures. LLMs also play a pivotal role in enhancing interactive music generation systems, where user inputs tailor the model's output, enriching the composing experience. While previous studies [1,4,5] have investigated LLMs in music tasks, detailed interpretations of the process remains less explored. This paper conduct an evaluation of four LLMs, GPT-4 [6], Gemma-7B-it [7], Llama2-7B-chat [8], and Qwen-7B-chat [9], assessing their capabilities on tasks related to symbolic music understanding and generation:

- Music Understanding: 1) Music theory exercise; 2) Motif extraction; 3) Musical form extraction.
- Music Generation: 1) Chord-conditioned music generation; 2) Melody harmonization; 3) Musical-form-and-motif-conditioned music generation



© Z. Zhou, Y. Wu, and Z. Wu. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Z. Zhou, Y. Wu, and Z. Wu, "Can LLMs "Reason" in Music? An Evaluation of LLMs' Capability of Music Understanding and Generation", in *Proc. of the 25th Int. Society for Music Information Retrieval Conf.*, San Francisco, United States, 2024.

The task of "Musical Form & Motif Conditioned Music Generation" as described in Chatmusician [1] involves generating music that adheres to detailed prescribed conditions like form and motif. Figure 1 illustrates this process: The prompt's green text specifies conditional constraints including the musical form, motif, and some musical elements (key, time signature, etc.). Under the prompt, the left sheet presents the human composer's work. The right sheets show ABC notations from different models alongside the reference. The Gemma-7B-it model merely replicates the provided motif, adding no new elements. Similarly, GPT-4 simply repeats the given condition. Qwen-7B-chat and Llama-7B-chat include correct musical elements and the motif but fail to capture the musical form "AB" and maintain the duration of a measure.

The main contributions of our paper are as follows: (1) we provide multi-step prompt engineering and explore how LLMs exhibit their reasoning capabilities with multi-step instructions in music understanding and generation tasks. (2) we assess four major LLMs on various symbolic music tasks, analyzing their reasoning in ABC sequences through quantitative statistical results and qualitative human assessment, including error analysis. The examples, hand-crafted prompts, and codes of data preprocessing are available at [github](#).

2. RELATED WORK

In this section, we summarize related works from two perspectives. First, we introduce previous studies on LLMs in the symbolic music domain, explaining their performance and evaluation methods in music understanding and generation tasks. Then, we discuss the application of LLMs in reasoning math problems and controllable creative text generation, highlighting similarities between the reasoning processes in music and math and the conditioned, open-ended nature of both music and text generation.

2.1 LLMs in Symbolic Music Domain

This subsection reviews the application of LLMs in the symbolic music domain. Previous studies have focused on adapting LLMs for music understanding and generation. Chatmusician [1] uses continual pre-training and fine-tuning on LLaMA2 to understand and generate ABC notation music, without specialized music structures or tokenizers. SongComposer [4] collects a song pretraining dataset including lyrics, melodies and paired lyrics-melodies, employing 10K crafted QA pairs to enable LLMs to perform multiple music-related tasks such as lyric-to-melody conversion and song continuation. MusicAgent [10] integrates various music tools into a single system, though it lacks interaction among these tools. Most approaches view music creation as a linear process, which diverges from the multi-step approach humans use, limiting their applicability for generating creative works. To mimic human creative processes, ByteComposer [5] employs a four-step method to replicate the creative workflow of human composers: conception analysis, draft Com-

position, self-evaluation and modification, and human aesthetic selection. And designs an interactive agent system consisting of expert, generator, voter, and memory modules. What's more, they construct supervised fine-tuning data covering tasks of basic music theory conception, control code generation, music score evaluation and next-step planning. Despite being a significant step towards multi-step music creation with LLMs, it lacks a detailed discussion on the limits of LLMs at each stage.

2.2 Reasoning and Controllable Generation with LLMs

"Reasoning" in NLP involves integrating various knowledge sources or contexts to generate new assertions, events, or actions [11]. This process often breaks complex questions into sequential steps [12]. Techniques such as Chain-of-Thoughts (CoT) [13, 14] have shown effectiveness in addressing complex reasoning tasks, particularly in mathematics. The Program-of-Thoughts approach improves upon CoT by using language models to generate text and code, enhancing math problem-solving performance [15]. Plan-and-Solve (PS) Prompting, a zero-shot technique, outperforms zero-shot CoT significantly, exceeds Zero-shot Program-of-Thoughts, and matches 8-shot CoT in math reasoning [16].

While music and mathematics share similarities, it is crucial to recognize that music is not as deterministic. In controllable music generation, despite given chords, motifs, and forms, unpredictable elements still significantly affect the quality of the music, similar to controllable text generation. Zhang et al. [17], identify three types of control conditions: semantic, structural, and lexical. Semantic controls refer to content control such as sentiment [18, 19] or topic [20, 21], resembling style and emotion in music. Structural control involves shaping the structure of the generated text, such as setting a story's framework or using data from tables or graphs as input, similar to specifying musical forms for generation [22, 23]. Lexical controls manage vocabulary usage, ensuring specific keywords appear, akin to using musical chords and motifs as guidelines. LLMs are extensively applied in diverse controllable and creative generation tasks [24–26]. These systems' abilities in long-context and multi-step generation under predefined conditions are examined, though such analyses are rarely applied in the music domain.

3. METHODOLOGY

3.1 Datasets

In this paper, we incorporate six tasks covering from music understanding to generation. The data is collected from *MusicPile* and *MusicBench* in ChatMusician [1]. The statistics of the dataset we use are shown in Table 1. Each model can support the maximum length of tokens of each task.

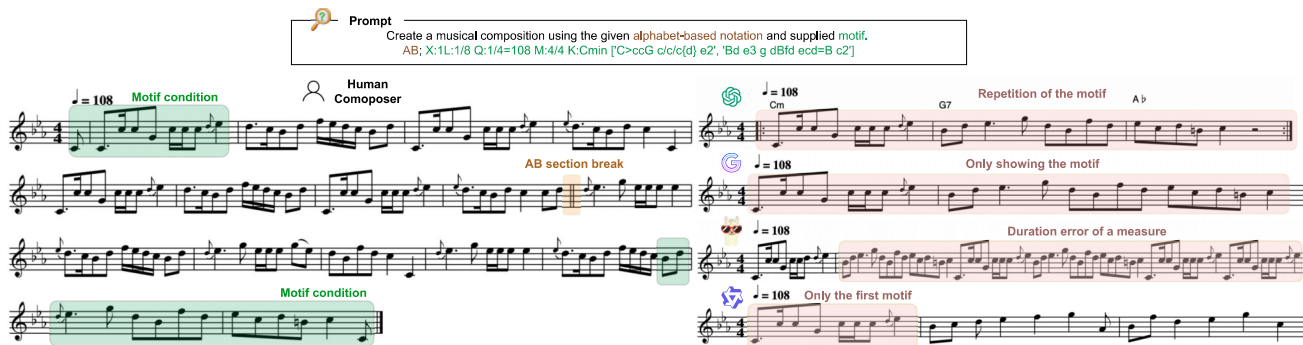


Figure 1. A comparison of different LLMs’ responses with the same instruction of the musical-form-and-motif-conditioned task as the input. The ABC notation contained in the response is extracted and displayed as scores the quality of all responses is marked with diverse symbols.

Tasks	Numbers	Max/Avg. tokens
Music theory exercise	367	733/103.56
Motif extraction (ME)	2470	1165/194.28
Musical form extraction (MFE)	483	650/187.35
Chord-conditioned generation (CCG)	1721	283/94.83
Melody harmonization (MH)	355	551/166.03
Musical-form-and-motif-conditioned generation (MFMC)	4881	285/53.82

Table 1. Statistics of each task. The number of items and the max and average length of tokens are provided.

3.2 Prompt Engineering

Before examining each LLM’s task performance, we conducted preliminary tests to verify their understanding of the relevant musical concepts. These tests confirmed that all models possess foundational knowledge of the six music tasks assessed in this study.

We employed two prompt modes in our experiments of all tasks, **Default** and **Chain-of-Thoughts (CoT)**. Default mode means forcing the model to respond without any analysis. Additionally, for music theory exercises, to make the model better understand the questions and options, and return the answer in a unified format, we also include the **In-Context-Learning (ICL)** mode by adding some question-answer pairs as examples shown to the models in the prompt. Taking the task of music theory exercises understanding as an example, three modes of prompts as the prefix of inputs followed by each item in the datasets are shown in Figure 2. Different from the music theory exercise, we specifically design prompts to support a multi-round chat conversation with LLMs for the generation tasks. Figure 3 shows an example of a four-round prompt set of chord-conditioned generation. We invite graduates who majored in music composition to write down their multi-step thoughts when completing the generation tasks involved in this paper. We summarize the common steps of all answers, adapt them to the prompt set, and make sure LLMs can understand or at least intend to follow the instructions. An example of GPT-4’s response to the instruction in Figure 3 is shown on the website¹.

Music Theory Exercise

Default: "You will see JSON-formatted instruction data followed by questions. Your responses should only indicate the selected option (using uppercase letters), without providing any analysis."

CoT: "You will see a JSON-formatted instruction data followed by questions. Your responses should include an analysis step by step. The returned JSON format is as follows: {\"reason\": \"**Let’s think step by step**\", \"answer\": \"A\"}"

ICL: CoT + "Here is an example of a question and its answer:
Read the following questions from the four options (A, B, C and D) given in each question. Choose the best option. Which of the following is the name of the note in the example?",

"L:1/4 M:4/4 K:Cb, D,4 |]",
Options: {"A": "B-flat", "B": "D", "C": "B", "D": "D-flat"},
Answer: "D".

Figure 2. A prompt example of the music theory exercise in different modes.

3.3 Pre-processing Responses

The responses of models are supposed to have correct ABC notations, but it may have certain syntax or formatting issues, and some outputs may even contain a large amount of natural language. We select the main features of ABC notation including field names and bar line symbols to help us extract the ABC sequence. If the extracted ABC sequence can be rendered into MIDI files using Music21² successfully and can be later rendering into audio file using midi2audio³, we consider it capable of producing valid ABC notation.

¹ https://github.com/SylviaZiyazhou/LLMs_music_reasoning/blob/main/CoT_music_generation_GPT4_response.pdf

² <https://web.mit.edu/music21>

³ <https://pypi.org/project/midi2audio>

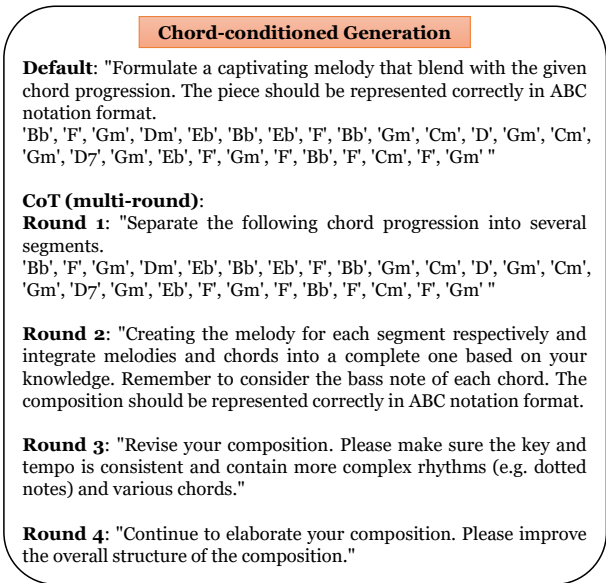


Figure 3. A prompt example of the chord-conditioned generation in different modes.

3.4 Multi-step Reasoning Analysis

In order to compare each model’s reasoning capability on both understanding and generation tasks, we first conduct a subjective assessment to evaluate how different models’ reasoning processes influence their performances. Participants are all familiar with basic music theory and can understand each task as well as the ABC notation. Secondly, based on the results of the subjective assessment, we further perform an error analysis in detail to show the intermediate answers during the reasoning process of each model.

3.4.1 Human Assessment Pipeline

In this section, we will provide a detailed description of our subjective experiments on four popular and open-source LLMs, including Gemma-7B-it, Llama2-7B-Chat, GPT-4, and Qwen-7B-chat. We ask the participants to evaluate to what extent the model understands the instructions and correctly answer the questions in the understanding tasks, and to what extent the responses contain the conditions and make creative works in the generation tasks. Specifically, the questions in the human assessment are as follows:

- For both understanding and generation tasks: 1) To what extent does the model understand and follow the instructions?
- Specifically for the understanding tasks: 1) To what extent does the model correctly answer the question? 2) To what extent does the model reason like human beings?
- Specifically for the generation tasks: 1) AB test: please choose the better one between a pair of music excerpts by considering their "Musicality"; 2) To what extent does the model contain the conditions?

Except for the AB test in the generation task, each question should be rated in a scoring range from 0 to 10 points. We invited music experts who are familiar with ABC notations as the participants in the human assessment, ensuring that each item was evaluated by at least two experts.

4. EVALUATION RESULTS

In this section, we provide the evaluation results based on the methodology we discussed in the last section. The quantitative results include the correctly parsing rate of ABC notation in the generation tasks, and the accuracy of music theory exercises. The qualitative results include the statistical analysis of human assessment and the detailed error analysis. Due to space limitation, we provide the examples at [github](#) and the online links of the corresponding files will be attached in the illustration.

4.1 Quantitative Results

Figure 4 shows the success rate of rendering valid audio from each LLM’s responses under different generation tasks. The pre-processing methodology is introduced in Section 3.3. Except for GPT-4, the other three models all have an audio generation rate of less than 50%, finding it difficult to generate the correct ABC notation format to be converted into audio.

Table 2 displays the accuracy of the music theory exercises in three modes. The reason why some models have an accuracy rate below 25% in multiple-choice questions with four options is that most of their responses seek additional information about the questions rather than answering them. Gemma-7B-it has a comparable performance with GPT-4 in the *Reason*. subset in the *Default* mode even with a much smaller model size. However, CoT and ICL modes, which significantly improve the GPT-4’s performance, show very limited effect or even deficiency in other models. This may inspire us to reconsider the utilization of classical CoT and ICL approaches in solving music tasks.

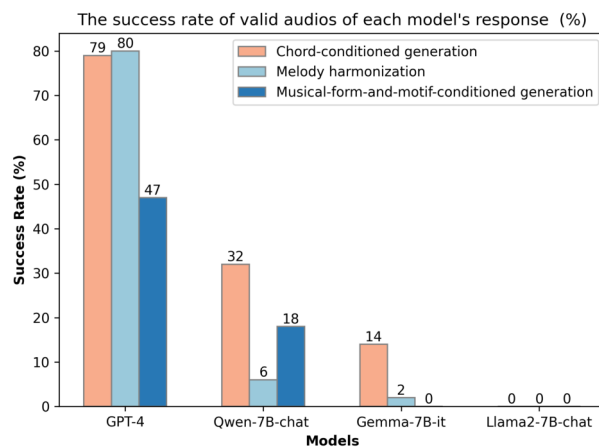


Figure 4. The success rate of rendering audio from each LLM’s responses in the music generation tasks.

Model (and Mode)	Know. (%)	Reas. (%)
GPT-4 (Default)	58.2	25.6
GPT-4 (CoT)	68.4	36.7
GPT-4 (ICL)	69.9	34.9
Llama2-7B-chat (Default)	11.9	10.2
Llama2-7B-chat (CoT)	29.8	16.3
Llama2-7B-chat (ICL)	10.4	15.3
Gemma-7B-it (Default)	45.7	31.6
Gemma-7B-it (CoT)	36.1	17.3
Gemma-7B-it (ICL)	33.1	31.6
Qwen-7B-chat (Default)	42.0	17.4
Qwen-7B-chat (CoT)	40.2	22.4
Qwen-7B-chat (ICL)	35.7	24.5

Table 2. Accuracy of the music theory exercises of each model. All three modes of results are provided. *Know.* means the music knowledge part and *Reas.* means the music reasoning part. They are two subsets of which the former tests the models’ memory of basic music concepts and the latter needs further reasoning and calculation to be completed. GPT-4’s results come from [1].

Type	Model	Inst. Fl.		Correct.		Reason.	
		μ	σ	μ	σ	μ	σ
ME	GPT-4	10.0	0.0	6.5	2.6	7.8	1.3
	Gemma	8.2	2.1	5.1	2.8	7.4	3.2
	Llama2	7.8	1.9	4.7	2.8	4.7	2.4
	Qwen	7.6	0.7	3.8	1.5	2.1	1.3
MFE	GPT-4	10.0	0.0	5.0	2.5	5.6	2.0
	Gemma	3.5	3.6	2.1	2.2	2.9	2.1
	Llama2	5.4	1.8	3.2	2.8	4.3	2.3
	Qwen	2.6	1.5	2.3	1.9	3.3	2.0

Table 3. The human assessment results of different LLMs on the understanding task. *Inst. Fl.*, *Correct.* and *Reason.* respectively indicate to what extent the model follows the instructions, correctly answers the questions, and reasons like humans. μ and σ respectively denote the average scores and the standard variance.

Type	Model	Inst. Fl.		Condi.	
		μ	σ	μ	σ
MFMC	GPT-4	5.7	1.4	6.3	1.5
	Gemma	4.0	1.8	4.6	2.2
	Llama2	4.3	1.6	4.3	2.3
	Qwen	4.9	2.1	2.9	2.2
MH	GPT-4	6.5	3.5	5.5	2.5
	Gemma	3.0	1.0	4.5	2.5
CCG	GPT-4	5.2	3.3	5.8	3.8
	Gemma	1.6	1.0	1.3	0.8

Table 4. The human assessment results of different LLMs on the generation task. *Condi.* indicates to what extent the model contains the condition given in the instructions and ABC format.

4.2 Qualitative Results

For human assessment, Table 3 shows LLMs on ME and MFE tasks under the *CoT* mode. We randomly sampled 40 examples of each task. In the instruction following question, GPT-4 demonstrates very good results, while other LLMs more or less can accomplish the tasks, indicating a certain level of capability. However, when it comes to the correctness, even GPT-4 finds it challenging to provide satisfactory answers to the prompts. When testing the logical reasoning of LLMs, the average scores indicate that all LLMs encounter difficulties in applying logical reasoning when answering questions step by step, leading to fundamental errors in music theory or illogical conclusions. This highlights the LLMs’ limitation of involving music background knowledge.

Table 4 presents the results of human assessment we conducted on generative tasks. In addition to the results shown in the table, we also conducted an AB test based on Musicality. We find that the GPT-4 and Gemma-7B-it achieve comparable results in MFMCG task, while in other tasks GPT-4 always wins. This means Gemma-7B-it has a potential in creating high-quality symbolic music with limited model size.

As depicted in Figure 4, on MH and CCG tasks, Qwen-7B-chat and Llama2-7B-chat struggled to effectively output correct ABC sequences to be rendered into audios. Therefore, for MH and CCG tasks, we only include the AB test results for GPT-4 and Gemma-7B-it. Despite GPT-4 achieving relatively better scores in generative tasks, it still falls far away from humans’ expectations. Interestingly, beyond the data, LLMs’ generative results occasionally exhibit instances of copying motifs provided in the prompt, as well as displaying unstructured harmonic repetitions or completely off-key notes. We believe that although LLMs can adhere to the ABC format condition provided in the prompt, their lack of musical information and knowledge makes it challenging to understand the high-level information within the condition, resulting in less satisfactory generated outcomes.

In terms of the results from subjective experiments, we identified a common issue prevalent in LLMs. Firstly, LLMs, apart from GPT-4, struggle to generate data in the correct ABC format with high probability, despite being able to provide a perfect answer when asked what ABC notation is. This phenomenon led us to speculate that while LLMs are trained extensively and comprehensively, LLMs can hardly understand all the information they have been exposed to and utilize them in different scenarios. Besides, LLMs can generate music in a seemingly appropriate ABC format in generative tasks, but what appears to be a correctly-formatted response is merely copying the prompt without grasping the semantic and structural information in the given condition.

4.2.1 Multi-step Reasoning Analysis

To better illustrate each model’s reasoning capability when it is used to complete the music theory exercises, we provide an example of a question in the music theory exer-



Figure 5. Human composer’s work for the chord-conditioned generation task.

cises subset and step-by-step responses of the four models⁴. The question is about recognizing the interval property of an ABC sequence referring to a compound in a music sheet. From the responses, we can see that GPT-4 is the only model which can actually perform the calculation but still unable to understand the musical notes in the ABC notation. In the GPT-4’s responses in the *CoT* mode, “4”, which is mistaken as “a fourth apart”, should be a note duration. Accordingly, this mistake influences the whole reasoning process of the calculation of intervals. The response of Llama2-7B-chat also shows its incapability of involving correct music knowledge understanding of notes intervals in the reasoning process. What’s more, Qwen-7B-chat even accidentally contains Chinese in the English text and Gemma-7B-it failed to recognize musical notes in the ABC sequence (see in the supplementary materials), although they can return the correct answer if they are merely asked about “the definition of note intervals”.

Besides, the responses of generation tasks such as MFMC generation also have similar problems. In the *CoT* mode, we find all LLMs except GPT-4, are hard to follow the multi-step instructions and output music in a correct ABC format, so we only provide a GPT-4 response respectively in the raw text⁵ and music sheet⁶ form given the prompt in Figure 3. Although GPT-4 can well understand the instructions in every step, it generates repetitive and simple rhythm without enough progression and variation, compared to the human composer’s work in Figure 5.

5. CONCLUSION AND DISCUSSION

In conclusion, our experimental analysis highlights current LLMs’ limitations in the realm of music understanding and generation, particularly from the perspective of song-level multi-step reasoning. These findings are crucial as they underline the challenges LLMs face when tasked with generating coherent and contextually rich musical compositions,

⁴ https://github.com/SylviaZiyaZhou/LLMs_music_reasoning/blob/main/CoT_music_theory_exercise_all_LLMs.pdf

⁵ https://github.com/SylviaZiyaZhou/LLMs_music_reasoning/blob/main/CoT_music_generation_GPT4_response.pdf

⁶ https://github.com/SylviaZiyaZhou/LLMs_music_reasoning/blob/main/Music_Sheet_of_music_generation_GPT4_response.pdf

which often require both complex sequential processing and creative fineness. From the human assessment results and the error analysis, we find that all these models failed to inject correct music theory and knowledge in the music understanding, reasoning and generation process. This knowledge generalization gap is analogous to the reversal curse problem illustrated in [27] where LLMs trained on “A is B” fail to learn “B is A”. Without making sure the fundamental concepts are correctly mentioned in the generated responses, it is hard to alleviate the LLMs’ hallucination and guarantee the responses’ quality. Therefore, it is significant to implement the knowledge augmentation module in the Supervised Fine-Tuning (SFT) stage to ensure the LLMs can reason based on correct music knowledge by curating more SFT data with enough knowledge-based contexts and practical reasoning processes.

Specifically, several insights for the multi-step SFT dataset construction can be concluded from the process where professional musicians are asked to create music following the instructions. Firstly, more expert knowledge should be involved in the dataset construction to guarantee its quality. For example, in the chord-conditioned generation task in Chatmusician’s dataset, the bass note sequence of the given chords does not conform to the musicians’ expectation of the progression generally. Secondly, some conditions in the old one-step form are too lengthy and informative with limitations that the human composers feel difficult to follow. For example, when they are given an “AB” structure with two different motives in the MFMC task, all of them find hard to integrate two segments with different motives into a complete piece of music in an “AB” form. Therefore, it might not be reasonable to ask the LLMs to output a completely and well composed music in a one-step approach.

What’s more, although four models are all claimed to be able to handle the input size from 4K to 8K tokens, which is much longer than the instructions in the dataset we used, they do not show their long-context processing advantages in the symbolic music domain. Our experimental results show that the widely-used *CoT* and *ICL* approaches are not always effective in improving the model’s performance. In this way, more step-by-step learning strategies should be specifically developed for instruction-based symbolic music tasks by focusing on correctly answering music theory exercises, explicitly extracting motifs and implicitly extracting musical forms, and consistently following the conditions in the instructions.

6. ACKNOWLEDGEMENT

The research was supported by Early Career Scheme (ECS-HKUST22201322), Theme-based Research Scheme (T45-205/21-N) from Hong Kong RGC, NSFC (No. 62206234), and Generative AI Research and Development Centre from InnoHK. Yinghao Ma is a research student at the UKRI Centre for Doctoral Training in Artificial Intelligence and Music, supported by UK Research and Innovation [grant number EP/S022694/1].

We would like to express our sincere gratitude to Jia

Ding and Xiaoduan Li, professional musicians who majored in music composition at Central Conservatory of Music, for their valuable contributions and suggestions throughout the multi-step prompt engineering in conditioned generation tasks.

7. REFERENCES

- [1] R. Yuan, H. Lin, Y. Wang, Z. Tian, S. Wu, T. Shen, G. Zhang, Y. Wu, C. Liu, Z. Zhou, Z. Ma, L. Xue, Z. Wang, Q. Liu, T. Zheng, Y. Li, Y. Ma, Y. Liang, X. Chi, R. Liu, Z. Wang, P. Li, J. Wu, C. Lin, Q. Liu, T. Jiang, W. Huang, W. Chen, E. Benetos, J. Fu, G. Xia, R. Dannenberg, W. Xue, S. Kang, and Y. Guo, “ChatMusician: Understanding and Generating Music Intrinsically with LLM,” Feb. 2024, arXiv:2402.16153 [cs, eess]. [Online]. Available: <http://arxiv.org/abs/2402.16153>
- [2] T. H. Garland and C. V. Kahn, *Math and Music: Harmonious Connections*. ERIC, 1995.
- [3] D. Wright, *Mathematics and music*. American Mathematical Soc., 2009, vol. 28.
- [4] S. Ding, Z. Liu, X. Dong, P. Zhang, R. Qian, C. He, D. Lin, and J. Wang, “Songcomposer: A large language model for lyric and melody composition in song generation,” *arXiv preprint arXiv:2402.17645*, 2024.
- [5] X. Liang, X. Du, J. Lin, P. Zou, Y. Wan, and B. Zhu, “ByteComposer: a Human-like Melody Composition Method based on Language Model Agent,” Mar. 2024, arXiv:2402.17785 [cs, eess]. [Online]. Available: <http://arxiv.org/abs/2402.17785>
- [6] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat *et al.*, “Gpt-4 technical report,” *arXiv preprint arXiv:2303.08774*, 2023.
- [7] G. Team, T. Mesnard, C. Hardin, R. Dadashi, S. Bhupatiraju, S. Pathak, L. Sifre, M. Rivière, M. S. Kale, J. Love *et al.*, “Gemma: Open models based on gemini research and technology,” *arXiv preprint arXiv:2403.08295*, 2024.
- [8] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale *et al.*, “Llama 2: Open foundation and fine-tuned chat models,” *arXiv preprint arXiv:2307.09288*, 2023.
- [9] J. Bai, S. Bai, Y. Chu, Z. Cui, K. Dang, X. Deng, Y. Fan, W. Ge, Y. Han, F. Huang *et al.*, “Qwen technical report,” *arXiv preprint arXiv:2309.16609*, 2023.
- [10] D. Yu, K. Song, P. Lu, T. He, X. Tan, W. Ye, S. Zhang, and J. Bian, “Musicagent: An ai agent for music understanding and generation with large language models,” *arXiv preprint arXiv:2310.11954*, 2023.
- [11] F. Yu, H. Zhang, and B. Wang, “Nature language reasoning, a survey,” *arXiv preprint arXiv:2303.14725*, 2023.
- [12] S. Hao, Y. Gu, H. Ma, J. J. Hong, Z. Wang, D. Z. Wang, and Z. Hu, “Reasoning with language model is planning with world model,” *arXiv preprint arXiv:2305.14992*, 2023.
- [13] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou *et al.*, “Chain-of-thought prompting elicits reasoning in large language models,” *Advances in neural information processing systems*, vol. 35, pp. 24 824–24 837, 2022.
- [14] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa, “Large language models are zero-shot reasoners,” *Advances in neural information processing systems*, vol. 35, pp. 22 199–22 213, 2022.
- [15] W. Chen, X. Ma, X. Wang, and W. W. Cohen, “Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks,” *arXiv preprint arXiv:2211.12588*, 2022.
- [16] L. Wang, W. Xu, Y. Lan, Z. Hu, Y. Lan, R. K.-W. Lee, and E.-P. Lim, “Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models,” *arXiv preprint arXiv:2305.04091*, 2023.
- [17] H. Zhang, H. Song, S. Li, M. Zhou, and D. Song, “A survey of controllable text generation using transformer-based pre-trained language models,” *ACM Computing Surveys*, vol. 56, no. 3, pp. 1–37, 2023.
- [18] H. Chen, X. Yi, M. Sun, W. Li, C. Yang, and Z. Guo, “Sentiment-controllable chinese poetry generation.” in *IJCAI*, 2019, pp. 4925–4931.
- [19] S. Dathathri, A. Madotto, J. Lan, J. Hung, E. Frank, P. Molino, J. Yosinski, and R. Liu, “Plug and play language models: A simple approach to controlled text generation,” *arXiv preprint arXiv:1912.02164*, 2019.
- [20] M. Khalifa, H. Elsahar, and M. Dymetman, “A distributional approach to controlled text generation,” *arXiv preprint arXiv:2012.11635*, 2020.
- [21] H. Tang, M. Li, and B. Jin, “A topic augmented text generation model: Joint learning of semantics and structural features,” in *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, 2019, pp. 5090–5099.
- [22] R. Puduppully, L. Dong, and M. Lapata, “Data-to-text generation with content selection and planning,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 6908–6915.

- [23] L. F. Ribeiro, M. Schmitt, H. Schütze, and I. Gurevych, “Investigating pretrained language models for graph-to-text generation,” *arXiv preprint arXiv:2007.08426*, 2020.
- [24] N. Simon and C. Muise, “Tattletale: storytelling with planning and large language models,” in *ICAPS Workshop on Scheduling and Planning Applications*, 2022.
- [25] K. Xie and M. Riedl, “Creating suspenseful stories: Iterative planning with large language models,” *arXiv preprint arXiv:2402.17119*, 2024.
- [26] Z. Zhang, M. Rayhan, T. Herda, M. Goisauf, and P. Abrahamsson, “Llm-based agents for automating the enhancement of user story quality: An early report,” *arXiv preprint arXiv:2403.09442*, 2024.
- [27] L. Berglund, M. Tong, M. Kaufmann, M. Balesni, A. C. Stickland, T. Korbak, and O. Evans, “The reversal curse: Llms trained on “a is b” fail to learn “b is a”,” 2024.

MUSIC2LATENT: CONSISTENCY AUTOENCODERS FOR LATENT AUDIO COMPRESSION

Marco Pasini¹

Stefan Lattner²

György Fazekas¹

¹Queen Mary University of London, UK

²Sony Computer Science Laboratories, Paris, France

m.pasini@qmul.ac.uk

ABSTRACT

Efficient audio representations in a compressed continuous latent space are critical for generative audio modeling and Music Information Retrieval (MIR) tasks. However, some existing audio autoencoders have limitations, such as multi-stage training procedures, slow iterative sampling, or low reconstruction quality. We introduce Music2Latent, an audio autoencoder that overcomes these limitations by leveraging consistency models. Music2Latent encodes samples into a compressed continuous latent space in a single end-to-end training process while enabling high-fidelity single-step reconstruction. Key innovations include conditioning the consistency model on upsampled encoder outputs at all levels through cross connections, using frequency-wise self-attention to capture long-range frequency dependencies, and employing frequency-wise learned scaling to handle varying value distributions across frequencies at different noise levels. We demonstrate that Music2Latent outperforms existing continuous audio autoencoders in sound quality and reconstruction accuracy while achieving competitive performance on downstream MIR tasks using its latent representations. To our knowledge, this represents the first successful attempt at training an end-to-end consistency autoencoder model. Pretrained weights are available under [this link].¹

1. INTRODUCTION

The ability to faithfully and efficiently represent high-dimensional audio data in a compressed latent space is crucial for a variety of applications, including generative modeling, music information retrieval (MIR), and audio compression. Generative models trained on latent representations of audio can be significantly more efficient than models trained directly on the data space, especially considering the high dimensionality of high-sample rate waveform samples. Additionally, a well-designed latent space can facilitate downstream MIR tasks by including musically relevant features in low-dimensional embeddings. However,

existing state-of-the-art audio autoencoders often present limitations, such as a multi-stage training process, the use of an unstable adversarial objective that requires multiple discriminators, and slow iterative sampling to reconstruct audio waveforms.

In this work, we introduce Music2Latent, a novel consistency autoencoder that encodes audio samples into a continuous latent space with a high compression ratio. Music2Latent is trained fully end-to-end using a single consistency loss function, making it easier to train than many existing audio autoencoders that require a careful balance between multiple competing loss terms [1–4]. Additionally, considering the underlying consistency model [5, 6], Music2Latent can reconstruct samples with high fidelity in a single step, enabling fast and efficient decoding. We evaluate Music2Latent on audio compression metrics, that measure the discrepancy between input and reconstructed samples, and on audio quality metrics, that establish the general audio quality of the reconstructions. Despite not being the primary focus of our model, we also investigate the downstream performance of encoded representations on standard Music Information Retrieval (MIR) tasks. Our experiments demonstrate that Music2Latent reconstructs samples more accurately and with higher audio quality compared to existing continuous autoencoder baselines while providing comparable or better performance on downstream tasks. Our contributions are as follows:

- We introduce Music2Latent, a consistency autoencoder that encodes waveforms into a continuous latent space with a 4096x time compression ratio.
- We show how it is possible to achieve high-quality reconstructions with a fully end-to-end training process relying on a single loss function.
- We introduce a frequency-wise self-attention and a frequency learned scaling mechanism, and demonstrate how they improve audio quality.
- We demonstrate that Music2Latent surpasses existing continuous autoencoder models in terms of reconstruction accuracy and audio quality while achieving competitive performance on downstream MIR tasks.

To the best of our knowledge, we are the first to successfully use consistency training in the music and audio field, and we are the first across all fields to successfully train an end-to-end consistency autoencoder model.

¹ <https://github.com/SonyCSLParis/music2latent>



2. RELATED WORK

2.1 Autoencoders for Latent Generative Modeling

Several autoencoder approaches have been explored in both the image and audio domains.

Image Domain: Vector Quantized Variational Autoencoders (VQ-VAE) [7] introduced the concept of learning discrete latent representations of images through vector quantization. VQ-VAE-2 [8] extended this approach to hierarchical codebooks, enabling the generation of realistic images using autoregressive models trained on the learned discrete latent codes. Vector Quantized Generative Adversarial Networks (VQGAN) [9] combine the VQ-VAE framework with adversarial training, incorporating a discriminator network to improve the perceptual quality of generated images. Latent Diffusion Models (LDMs) [10] leverage diffusion models trained on the latent space of a pre-trained autoencoder. By operating on a compressed representation of the data, LDMs achieve high-quality image synthesis with reduced computational requirements compared to pixel-based diffusion models. Diffusion autoencoders [11] combine a learnable encoder with a diffusion model as the decoder, aiming to learn a meaningful and decodable representation of images in a fully end-to-end manner. However, they still require a slow iterative sampling process to reconstruct samples.

Audio Domain: The audio autoencoder proposed in the Musika music generation system [1] encodes audio into a continuous latent space by reconstructing the magnitude and phase components of a spectrogram. While Musika achieves fast inference, it requires a two-stage training process combined with an unstable adversarial objective. Moûsai introduces a diffusion autoencoder [12] to learn a compressed invertible audio representation. However, it requires multiple sampling steps for reconstruction. Several audio autoencoders employ Residual Vector Quantization (RVQ) to learn discrete latent representations. Examples include SoundStream [2], EnCodec [3], and Descript Audio Codec (DAC) [4]. These models are well-suited for training autoregressive models on the latent representations but are less suitable for other generative models such as diffusion, consistency, or GAN-based methods. They also generally produce (discrete) representations at a significantly lower time compression ratio than continuous models, and are thus not directly comparable to our work.

2.2 Consistency Models

Consistency models [5, 6] offer a novel approach for efficient generative modeling by learning a mapping from any point on a diffusion trajectory to the trajectory’s starting point. They have been successfully applied to image generation tasks [13], achieving high-quality results with single-step sampling. The application of consistency models to audio generation is still relatively unexplored. CoMoSpeech [14] explores consistency distillation for speech synthesis, but it requires a pre-trained diffusion model to be trained.

3. BACKGROUND

3.1 Consistency Models

Consistency models represent a novel family of generative models capable of producing high-quality samples in a single step, without the need for adversarial training or iterative sampling. They are grounded in the probability flow ordinary differential equation (ODE) introduced by [15]:

$$\frac{dx}{d\sigma} = -\sigma \nabla_x \log p_\sigma(x), \quad \sigma \in [\sigma_{\min}, \sigma_{\max}] \quad (1)$$

Here, $p_\sigma(x)$ represents the perturbed data distribution obtained by adding Gaussian noise with zero mean and standard deviation σ to the original data distribution $p_{\text{data}}(x)$. The term $\nabla_x \log p_\sigma(x)$ is known as the score function, which plays a crucial role in score-based generative models [16–18]. The probability flow ODE establishes a bijective mapping between a noisy data sample $x_\sigma \sim p_\sigma(x)$ and $x_{\sigma_{\min}} \sim p_{\sigma_{\min}}(x) \approx x \sim p_{\text{data}}(x)$. This mapping, denoted as $f(x_\sigma, \sigma) \mapsto x_{\sigma_{\min}}$, is termed the consistency function, which satisfies the boundary condition $f(x_{\sigma_{\min}}, \sigma_{\min}) = x_{\sigma_{\min}}$. A consistency model $f_\theta(x_\sigma, \sigma)$ is a neural network trained to approximate the consistency function $f(x_\sigma, \sigma)$. To meet the boundary condition, consistency models are parameterised as:

$$f_\theta(x_\sigma, \sigma) = c_{\text{skip}}(\sigma)x_\sigma + c_{\text{out}}(\sigma)F_\theta(x_\sigma, \sigma) \quad (2)$$

where $F_\theta(x_\sigma, \sigma)$ is a free-form neural network, and $c_{\text{skip}}(\sigma)$ and $c_{\text{out}}(\sigma)$ are differentiable functions such that $c_{\text{skip}}(\sigma_{\min}) = 1$ and $c_{\text{out}}(\sigma_{\min}) = 0$.

Consistency models can be trained using either consistency distillation (CD) or consistency training (CT). CD requires pre-training a diffusion model to estimate the score function $\nabla_x \log p_\sigma(x)$ via score matching [19]. CT, on the other hand, allows training consistency models in isolation and is the method that is considered in this work.

3.2 Consistency Training

In consistency training, the probability flow ODE is discretised using a sequence of noise levels $\sigma_{\min} = \sigma_1 < \sigma_2 < \dots < \sigma_N = \sigma_{\max}$. The consistency model $f_\theta(x_\sigma, \sigma)$ is then trained by minimising the following consistency training loss over θ :

$$\mathcal{L}_{\text{CT}} = \mathbb{E} \left[\lambda(\sigma_i, \sigma_{i+1}) d \left(f_\theta(x_{\sigma_{i+1}}, \sigma_{i+1}), f_{\theta^-}(x_{\sigma_i}, \sigma_i) \right) \right] \quad (3)$$

where $d(x, y)$ is a metric function such as mean squared error and $\lambda(\sigma_i, \sigma_{i+1})$ is a noise level-dependent loss scaling. In the above equations, f_θ and f_{θ^-} are referred to as the student network and the teacher network respectively. The teacher’s parameters θ^- are obtained by applying a stop-gradient operation to the student’s parameters θ during training:

$$\theta^- \leftarrow \text{stopgrad}(\theta) \quad (4)$$

After training, the consistency model $f_\theta(x, \sigma)$ can directly generate a sample x by starting with $z \sim \mathcal{N}(0, I)$ and computing $x = f_\theta(\sigma_{\max}z, \sigma_{\max})$. This enables efficient one-step sampling, a key advantage of consistency models over diffusion models.

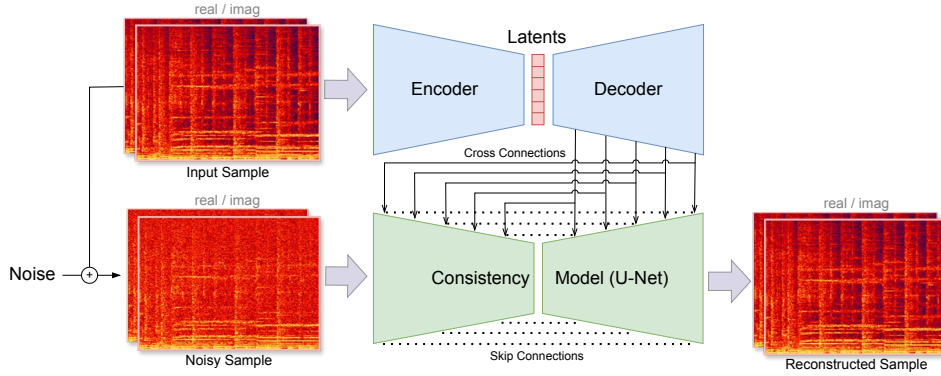


Figure 1. Training process of Music2Latent. The input sample is first encoded into a sequence of latent vectors. The latents are then upsampled with a decoder model. The consistency model is trained via consistency training, with an additional information leakage coming from the cross connections.

4. MUSIC2LATENT

In the following sections, we provide a detailed explanation of the audio representation, architecture, and training framework underlying Music2Latent.

4.1 Audio Representation

Music2Latent utilises complex-valued STFT spectrograms as the representation of waveform audio. This choice is motivated by several factors. First, previous works [20, 21] have demonstrated the effectiveness of complex spectrograms in capturing the intricate structure of audio signals and enabling the generation of high-fidelity audio. Second, 2-dimensional spectrograms allow for the direct application of UNet architectures [22] that have been successfully used in the image domain with diffusion and consistency models. However, the distribution of values across different frequencies in a STFT spectrogram can vary significantly, with substantially higher magnitudes in low frequencies compared to high frequencies. This can hinder the ability of the model to accurately reconstruct all frequency components, as the learning signal for high frequencies may be overshadowed by the stronger signal from lower frequencies. To address this issue, we apply the amplitude transformation proposed in [23] and later used in [24] which scales up lower energy components in the spectrogram:

$$\tilde{c} = \beta |c|^\alpha e^{i\angle(c)} \quad (5)$$

where c is the original complex STFT coefficient, \tilde{c} is the transformed coefficient, $\alpha \in (0, 1]$ is a compression exponent that emphasizes lower-energy frequency components, $\angle(c)$ represents the phase angle of c , and $\beta \in \mathbb{R}^+$ is a scaling factor to normalize amplitudes within a desired range (e.g., $[0, 1]$). This transformation ensures that the model receives a more balanced representation of the audio signal, facilitating accurate reconstruction across all frequencies. We consider the complex STFT spectrogram as a 2-channel representation, with each channel representing real and imaginary components respectively.

4.2 Architecture

The architecture of Music2Latent consists of an encoder, a decoder, and a consistency model.

Encoder: The encoder receives as input the audio sample in the form of an STFT spectrogram with real and imaginary components in each channel. It then gradually down-samples the feature maps along the time axis and outputs a sequence of latent vectors with dimensionality d_{lat} . Instead of being trained with a VAE objective [10, 25] to keep the distribution of latent values under control, the latent encodings of the model are kept in the $(-1, 1)$ range using a \tanh activation function, which was proven to be a successful approach in previous works for downstream latent generative modeling tasks [1, 12].

Decoder: The decoder mirrors the encoder architecture but performs upsampling instead of downsampling. The decoder takes as input a sequence of latent vectors from the encoder and progressively upsamples them to match the dimensionality of the feature maps of the consistency model. The only purpose of the decoder is to ensure that the conditioning information from the latent encodings is available to the consistency model at all levels of its architecture (the reason for this architectural choice is provided in the next section).

Consistency Model: The consistency model uses a UNet architecture with a downsampling branch and an upsampling branch connected via additive skip connections. The output of the decoder at each upsampling layer is also added to the corresponding layer of the consistency model. This provides cross connections that allow the consistency model to directly access the conditioning information about the sample it is attempting to reconstruct at all levels of its architecture. This design choice is crucial for single-step reconstruction, as it ensures that the model has access to the necessary information to accurately reconstruct the target sample from the very beginning of the UNet architecture.

Adaptive Frequency Scaling: The distribution of values along the frequency axis in the input spectrograms changes

significantly with respect to the noise level σ . Specifically, when σ is close to σ_{min} , the magnitudes at low frequencies are on average much higher than the ones at high frequencies, while with σ approaching σ_{max} , there is an equal distribution of values across all frequencies since the sample is pure noise. To address this, we introduce a frequency-wise scaling mechanism that adaptively scales the input and output of the consistency model based on the current noise level. Specifically, we employ a Multi-Layer Perceptron (MLP) that takes as input the noise level σ in the form of a sinusoidal embedding [26] and outputs a scaling factor for each frequency bin:

$$s_f(\sigma) = \text{MLP}(\sigma), \quad (6)$$

where $s_f(\sigma) \in \mathbb{R}^F$ is a vector of scaling factors, one for each of the F frequency bins of the noisy spectrogram. We calculate different scaling factors to scale both the input x_σ and the output of the consistency model $F_\theta(x_\sigma)$ as follows:

$$\tilde{x}_\sigma = x_\sigma \odot s_{f,\text{in}}(\sigma) \quad \tilde{F}_\theta(x_\sigma) = F_\theta(x_\sigma) \odot s_{f,\text{out}}(\sigma) \quad (7)$$

where \odot denotes element-wise multiplication.

Frequency-wise self-attention: To capture long-range dependencies within the frequency domain while keeping a memory footprint that scales linearly with the time axis, Music2Latent employs frequency-wise self-attention. This mechanism allows the model to attend to information from all frequency bins at a given time step, enabling it to learn complex relationships between different frequency components. Considering that only the time dimension of the input can vary at inference time, using frequency-wise attention compared to full self-attention does not incur in a memory requirement that scales quadratically with time. After computing the query Q , key K , and value V via linear projections of the input features, we calculate the attention matrix A by performing an outer product on individual timesteps t :

$$A_t = \text{softmax} \left(\frac{Q_t K_t^T}{\sqrt{d}} \right) \quad (8)$$

where d is the channel dimension, and after concatenating the attention weights from all timesteps together we have $A \in \mathbb{R}^{T \times F \times F}$. The softmax operation is then applied across the frequency dimension, ensuring that the attention weights for each frequency bin sum to one.

4.3 Training Process

Music2Latent is trained using the consistency training (CT) objective [5, 6]. As described in Sec. 3.2, the objective minimizes the discrepancy between the outputs of the consistency model at adjacent noise levels σ_i and σ_{i+1} . As for the distance metric in the consistency training loss function (Eq. 3), we use the Pseudo-Huber loss function [27] which smoothly transitions from the ℓ_1 to the squared ℓ_2 metrics:

$$d(x, y) = \sqrt{|x - y|^2 + c^2} - c, \quad (9)$$

where c is a hyperparameter that controls the transition. In [6], it was shown that for image generation with consistency models, this loss provides smoother gradients during training and performs substantially better compared to the more common squared ℓ_2 loss. The consistency model is parameterised as described in Eq. 2, with the exception that in addition to providing as input the noisy sample x_σ , we allow for information leakage of the clean sample x through the features \mathbf{y}_x provided by the decoder via cross connections:

$$\begin{aligned} \text{lat}_x &= \text{Enc}_\theta(x) & \mathbf{y}_x &= \text{Dec}_\theta(\text{lat}_x) \\ f_\theta(x_\sigma, \sigma, \mathbf{y}_x) &= c_{\text{skip}}(\sigma)x_\sigma + c_{\text{out}}(\sigma)F_\theta(x_\sigma, \sigma, \mathbf{y}_x) \end{aligned} \quad (10)$$

which results in the following consistency loss that is used to train the system fully end-to-end:

$$\mathcal{L} = \mathbb{E} [\lambda(\sigma_i, \sigma_{i+1})d(f_\theta(x_{\sigma_{i+1}}, \sigma_{i+1}, \mathbf{y}_x), f_{\theta^-}(x_{\sigma_i}, \sigma_i, \mathbf{y}_x))] \quad (11)$$

With respect to the noise level-dependent loss scaling $\lambda(\sigma_i, \sigma_{i+1})$, we follow [6] and use:

$$\lambda(\sigma_i, \sigma_{i+1}) = \frac{1}{\sigma_{i+1} - \sigma_i} \quad (12)$$

which assigns a higher weight to the loss when there is a small gap between consecutive noise levels. We also adopt the lognormal sampling of σ introduced by [28] and adopted for consistency training by [6] to focus training on a more relevant range of noise levels.

Continuous Noise Levels: Unlike the formulation presented in previous consistency model literature [5, 6], which use a discrete set of noise levels for training, Music2Latent employs a continuous noise schedule. This change is inspired by recent state-of-the-art diffusion models which notably sample noise levels from a continuous distribution [28]. Parallel work on improving the performance of consistency models also demonstrates how employing a continuous noise schedule improves results compared to the original discrete schedule [29]. Specifically, we use an exponential schedule during training to determine the step size between consecutive noise levels used for the consistency loss:

$$\Delta t_k = \Delta t_0^{\frac{k}{K}(e_K - 1) + 1} \quad (13)$$

where Δt_k is the step size at training iteration k , Δt_0 is the initial step size at iteration 0, and e_K is the exponent at final iteration K . This schedule ensures that the step size decreases exponentially as training progresses, allowing the model to gradually learn finer details of the data distribution. In order to calculate σ_i and σ_{i+1} , we first sample a timestep $t_{i+1} \in [0, 1]$ with the sampling weights given by the lognormal distribution, and calculate the adjacent timestep $t_i = \max(t_{i+1} - \Delta t_k, 0)$. Finally we calculate σ_i using the time step-to-noise level mapping from [28]:

$$\sigma_i = \left(\sigma_{\min}^{\frac{1}{\rho}} + t_i \left(\sigma_{\max}^{\frac{1}{\rho}} - \sigma_{\min}^{\frac{1}{\rho}} \right) \right)^\rho \quad (14)$$

where $\rho = 7$. We use the same mapping to calculate σ_{i+1} .

	MagnaTagATune		Beatport		TinySOL-pitchclass		TinySOL-instrument	
	AUC-ROC	AUC-PR	Micro Acc.	Macro Acc.	Micro F1	Macro F1	Micro F1	Macro F1
Musika	84.8	32.9	45.2	41.0	93.5	93.4	93.3	84.5
LatMusic	85.9	34.9	37.4	30.2	88.9	88.8	92.6	80.7
Moûsai_v2	86.2	35.4	48.2	42.0	95.1	95.1	82.8	68.6
Moûsai_v3	85.8	34.5	39.8	31.9	95.5	95.6	93.1	82.3
Music2Latent	<u>88.6</u>	<u>40.0</u>	<u>65.5</u>	<u>60.1</u>	<u>99.8</u>	<u>99.8</u>	92.6	81.0
MusiCNN-MSD	87.6	37.5	13.5	7.3	17.2	15.7	68.2	60.8
CLMR	89.9	42.6	13.9	7.8	16.8	16.2	93.5	89.7
MERT-v1-95M	90.8	44.9	50.7	44.3	98.3	98.3	97.1	95.8

Table 1. Downstream task performance on MagnaTagATune (autotagging), Beatport (key estimation), TinySOL (pitch and instrument classification). Best results among autoencoder baselines are underlined.

4.4 Implementation Details

With respect to the UNet architecture of the consistency model, we use the NCSN++ architecture introduced in [17], which consists of convolutional residual blocks with 3x3 kernels, Swish activation function [30] and Group Normalisation layers. The same residual blocks are used in both the encoder and decoder. We use sinusoidal embeddings to encode the noise level, using $\frac{\log(\sigma)}{4}$ as the input. The skip connections between the downsampling and the upsampling branches of the UNet are added instead of being concatenated, as recent works on diffusion models [31] show that addition provides better performance. Consequently, the cross connections from the decoder are also added to the corresponding UNet features, following a linear projection layer. In the encoder, before the final bottleneck layer with a \tanh activation function, used to constrain the latent encodings to the $(-1,1)$ range, the 2D features are reshaped into 1D features by flattening the frequency dimension into the channel dimension, and a series of 4 residual blocks with 1D convolutions with kernel size of 3 are used. We choose $d_{lat} = 64$, which results in a $4096x$ time compression ratio and a $64x$ total compression ratio. The decoder perfectly mirrors the architecture of the encoder, while not receiving any incoming skip connections, since all the information necessary to reconstruct the clean input sample must be contained in the latent encodings. For the consistency model and encoder/decoder models we use 5 levels corresponding to 4 upsampling/downsampling operations, and in each level we use 2 residual blocks for the consistency model, and 1 residual block for the encoder and decoder. The base channels for all models are set to 64 and the channel multiplier for each of the 5 levels is set to $[1, 2, 4, 4, 4]$ for all models. We use 512 channels for the 1D convolutional blocks in the encoder and decoder. We use frequency-wise self-attention layers with 4 heads in the 3 last levels for all models, in order not to use it with higher frequency dimensions. The channels used for sinusoidal embeddings and the MLPs used for both noise level embeddings and frequency scalings are set to 256. The model has ~ 58 million parameters. The consistency training framework follows the same implementation of [6] with respect to the scaling factors $c_{in}, c_{skip}, c_{out}$, the parameter c for the pseudo-Huber loss function, the minimum and maximum noise parameters $\sigma_{min}, \sigma_{max}$, the standard deviation of the data samples σ_{data} , and the lognormal distribution values

of P_{mean}, P_{std} . Regarding the input STFT spectrograms, we extract them using $hop = 512$, $window = 4 \cdot hop$ and we transform them using the formula presented in Sec. 4.1, with $\alpha = 0.65, \beta = 0.35$. Regarding the step size schedule for the continuous noise levels, we choose $\Delta t_0 = 0.1$ and $e_K = 3$. We train the model on waveforms of 34, 304 samples, which correspond to ~ 0.78 s of 44.1 kHz audio. The model thus produces latent representations of 44.1 kHz audio at a sampling rate of ~ 11 Hz. We use a batch size of 16 and train for $K = 800k$ iterations using the RAdam optimizer [32] with $lr_0 = 1e^{-4}, \beta_1 = 0.9, \beta_2 = 0.999$. We use a cosine learning rate decay with $lr_K = 1e^{-6}$ and we keep an Exponential Moving Average (EMA) of the parameters of all models with a momentum of 0.9999. Training takes ~ 5 days on a single RTX 3090 GPU.

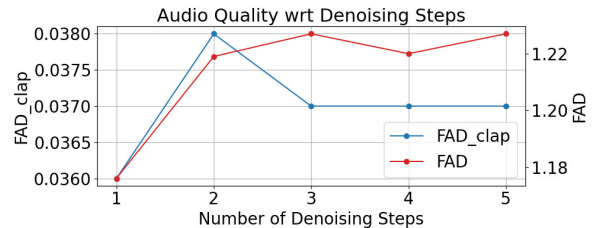


Figure 2. Audio quality of reconstructed samples with respect to the number of denoising steps of the consistency model.

5. EXPERIMENTS AND RESULTS

5.1 Datasets

We train the model on MTG Jamendo [33] and on the clean speech segments from DNS Challenge 4 [34], sampling from each dataset with equal probability. We keep the original sample rates of 44.1 kHz and 48 kHz. We include speech in the training data to both improve the reconstruction of vocal content in music samples, and to make Music2Latent useful also for speech-related tasks. We use MusicCaps [35] as our evaluation dataset.

5.2 Baselines

We compare Music2Latent to different audio autoencoders that encode audio samples into a continuous latent space to enable downstream latent generative modeling. We include the autoencoder introduced in Musika [1] and the

autoencoder introduced by [36] to train a latent diffusion model for music accompaniment generation (we name this model LatMusic in our comparison). Both models encode audio samples with the same compression ratio of $64x$ as Music2Latent. We also include the diffusion autoencoder introduced in Moûsai [12], which has a compression ratio of $32x$ (Moûsai_v3), and a different autoencoder model that is made available by the authors of Moûsai² with a comparable compression ratio of $64x$ (Moûsai_v2).

5.3 Audio Compression and Quality

	SI-SDR \uparrow	ViSQOL \uparrow	FAD _{clap} \downarrow	FAD \downarrow
Musika	-25.81	3.80	0.103	2.308
LatMusic	-27.32	3.95	0.050	1.630
Moûsai_v2	-21.44	2.36	0.731	4.687
Moûsai_v3	-17.47	2.28	0.647	4.473
Music2Latent	-3.85	3.84	0.036	1.176
DAC	9.48	4.21	0.041	0.966

Table 2. Audio compression and quality results.

We adopt the same objective evaluation metrics as in [3] and use Scale-Invariant Signal-to-Noise Ratio (SI-SDR) [37] and ViSQOL [38–40]. SI-SDR is a distance calculated between input and reconstructed waveforms, while ViSQOL estimates a MOS-like score on perceptual quality from the difference between the two signals. Considering that Music2Latent is trained as a generative model, we also use Fréchet Audio Distance (FAD [41]) to evaluate the general audio quality of reconstructed samples without relying on paired samples. In addition to the original FAD implementation, we also evaluate on FAD_{clap} using CLAP [42] features, which was shown to correlate significantly better with perceived audio quality [43]. In Tab. 2 we show that Music2Latent is competitive with respect to ViSQOL to Musika and LatMusic, while vastly outperforming all baselines on the remaining metrics. Note that all four baselines discard phase information from the input of the autoencoder, which may explain the poor SI-SDR performance. DAC, while not being directly comparable, scores favourably in reconstruction metrics, while matches Music2Latent in terms of audio quality. In Fig. 2 we also show that the audio quality of reconstructions remains almost constant when using more than a single denoising step. We provide audio samples and additional supplementary material on the accompanying website³.

5.4 Ablation Study

	FAD _{clap} \downarrow	FAD \downarrow
Base Model	0.0563	1.808
+ Freq-wise Attention	0.0547	1.710
+ Adaptive Freq Scaling	0.0537	1.665

Table 3. Ablation study. *Base Model* is trained without frequency-wise attention and adaptive frequency scaling.

² <https://github.com/archinetai/archisound>

³ <https://sonyyslparis.github.io/music2latent-companion/>

To demonstrate the effectiveness of both frequency-wise attention and learned frequency scaling, we perform an ablation study and report the FAD and FAD_{clap} results in Table 3. With respect to the model with no attention and no scaling, we use channel multipliers [1, 2, 4, 4, 5] to roughly match the number of parameters that are lost. All ablated models are trained for 200k iterations. The remaining training details are the ones described in Sec. 4.4.

5.5 Downstream Performance

Since training representation learning models on compressed audio representations instead of raw data was shown to be a promising approach [44–47], our goal is to investigate whether there are well disentangled audio features in the feature space of audio autoencoders. We evaluate downstream performance on MagnaTagATune [48] for auto-tagging, Beatport [49] for key estimation, and TinySOL [50] for instrument and pitch class classification. For each dataset, we extract the encoder features from the layer with the highest number of output channels from each of the models (after flattening the 2D features for Music2Latent and before the last linear layer for the remaining models), average them along the time axis, and train a 2-layer MLP with [256, 128] units. We also show the results obtained by performing the same evaluation on features from the classification model MusiCNN-MSD [51] and well-established representation learning models CLMR [52] and MERT-v1-95M [47] (with averaged features from layers 9 to 12). We extract features from these models following [53] and perform all evaluations using the `mir_ref` library⁴ [54]. In Tab. 1 we show how Music2Latent outperforms autoencoder baselines in almost all tasks, and in the case of key and pitch classification it even outperforms state-of-the-art representation learning models. We hypothesize that the loss is more sensitive to pitch information than timbre content (explaining the weak comparison on TinySOL-instrument to the representation learning models).

6. CONCLUSION

In this work we introduced Music2Latent, a consistency autoencoder that efficiently compresses high-dimensional audio waveforms into a continuous latent space. By leveraging consistency training, Music2Latent achieves high-fidelity single-step reconstruction, and enables efficient downstream latent generative modeling. We propose a learned frequency scaling mechanism to handle varying frequency distributions across diffusion noise levels. Experiments show Music2Latent matches or outperforms baselines in reconstruction accuracy and audio quality, while having comparable or better performance on downstream tasks. To our knowledge, Music2Latent represents the first successful end-to-end consistency autoencoder. Future work could explore extensions to other modalities and higher compression ratios. Overall, we believe Music2Latent is a significant contribution to audio generative modeling and representation learning.

⁴ https://github.com/chrispla/mir_ref

7. ACKNOWLEDGEMENTS

This work is supported by the EPSRC UKRI Centre for Doctoral Training in Artificial Intelligence and Music (EP/S022694/1) and Sony Computer Science Laboratories Paris.

8. REFERENCES

- [1] M. Pasini and J. Schlüter, “Musika! Fast Infinite Waveform Music Generation,” in *Proceedings of the 23rd International Society for Music Information Retrieval Conference, ISMIR 2022, Bengaluru, India, December 4-8, 2022*, 2022, pp. 543–550.
- [2] N. Zeghidour, A. Luebs *et al.*, “SoundStream: An End-to-End Neural Audio Codec,” *IEEE ACM Trans. Audio Speech Lang. Process.*, vol. 30, pp. 495–507, 2022.
- [3] A. Défossez, J. Copet *et al.*, “High Fidelity Neural Audio Compression,” Oct. 2022, arXiv:2210.13438 [cs, eess, stat].
- [4] R. Kumar, P. Seetharaman *et al.*, “High-Fidelity Audio Compression with Improved RVQGAN,” Jun. 2023, arXiv:2306.06546 [cs, eess].
- [5] Y. Song, P. Dhariwal *et al.*, “Consistency Models,” May 2023, arXiv:2303.01469 [cs, stat].
- [6] Y. Song and P. Dhariwal, “Improved techniques for training consistency models,” *arXiv preprint arXiv:2310.14189*, 2023.
- [7] A. van den Oord, O. Vinyals *et al.*, “Neural discrete representation learning,” in *Advances in Neural Information Processing Systems 30*, Dec. 2017, pp. 6306–6315.
- [8] A. Razavi, A. van den Oord *et al.*, “Generating diverse high-fidelity images with VQ-VAE-2,” in *Advances in Neural Information Processing Systems 32*, Dec. 2019, pp. 14 837–14 847.
- [9] P. Esser, R. Rombach *et al.*, “Taming transformers for high-resolution image synthesis,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Computer Vision Foundation / IEEE, Jun. 2021, pp. 12 873–12 883.
- [10] R. Rombach, A. Blattmann *et al.*, “High-resolution image synthesis with latent diffusion models,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 10 684–10 695.
- [11] K. Preechakul, N. Chatthee *et al.*, “Diffusion autoencoders: Toward a meaningful and decodable representation,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*. IEEE, 2022, pp. 10 609–10 619.
- [12] F. Schneider, Z. Jin *et al.*, “Mo^usai: Text-to-Music Generation with Long-Context Latent Diffusion,” Jan. 2023, arXiv:2301.11757 [cs, eess].
- [13] S. Luo, Y. Tan, L. Huang, J. Li, and H. Zhao, “Latent consistency models: Synthesizing high-resolution images with few-step inference,” *arXiv preprint arXiv:2310.04378*, 2023.
- [14] Z. Ye, W. Xue *et al.*, “Comospeech: One-step speech and singing voice synthesis via consistency model,” in *Proceedings of the 31st ACM International Conference on Multimedia, MM 2023, Ottawa, ON, Canada, 29 October 2023- 3 November 2023*. ACM, 2023, pp. 1831–1839.
- [15] J. Song, C. Meng *et al.*, “Denoising Diffusion Implicit Models,” in *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.
- [16] Y. Song and S. Ermon, “Generative modeling by estimating gradients of the data distribution,” in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 2019, pp. 11 895–11 907.
- [17] Y. Song, J. Sohl-Dickstein *et al.*, “Score-based generative modeling through stochastic differential equations,” *arXiv preprint arXiv:2011.13456*, 2020.
- [18] Y. Song and S. Ermon, “Improved techniques for training score-based generative models,” in *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.
- [19] Y. Song, S. Garg *et al.*, “Sliced Score Matching: A Scalable Approach to Density and Score Estimation,” in *Proceedings of the Thirty-Fifth Conference on Uncertainty in Artificial Intelligence, UAI 2019, Tel Aviv, Israel, July 22-25, 2019*, ser. Proceedings of Machine Learning Research, vol. 115. AUAI Press, 2019, pp. 574–584.
- [20] J. Nistal, S. Lattner *et al.*, “DRUMGAN: synthesis of drum sounds with timbral feature conditioning using generative adversarial networks,” in *Proceedings of the 21th International Society for Music Information Retrieval Conference (ISMIR)*, Oct. 2020, pp. 590–597.
- [21] —, “Comparing representations for audio synthesis using generative adversarial networks,” in *28th European Signal Processing Conference (EUSIPCO)*. IEEE, Jan. 2020, pp. 161–165.
- [22] O. Ronneberger, P. Fischer *et al.*, “U-net: Convolutional networks for biomedical image segmentation,” in *Medical Image Computing and Computer-Assisted*

- Intervention - MICCAI 2015 - 18th International Conference Munich, Germany, October 5 - 9, 2015, Proceedings, Part III*, ser. Lecture Notes in Computer Science, vol. 9351. Springer, 2015, pp. 234–241.
- [23] J. Richter, S. Welker *et al.*, “Speech enhancement and dereverberation with diffusion-based generative models,” *IEEE ACM Trans. Audio Speech Lang. Process.*, vol. 31, pp. 2351–2364, 2023.
- [24] G. Zhu, Y. Wen *et al.*, “Edmsound: Spectrogram based diffusion models for efficient and high-quality audio synthesis,” *arXiv preprint arXiv:2311.08667*, 2023.
- [25] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” in *2nd International Conference on Learning Representations (ICLR)*, Apr. 2014.
- [26] A. Vaswani, N. Shazeer *et al.*, “Attention is all you need,” in *Advances in Neural Information Processing Systems 30*, Dec. 2017, pp. 5998–6008.
- [27] P. Charbonnier, L. Blanc-Feraud *et al.*, “Deterministic edge-preserving regularization in computed imaging,” *IEEE Transactions on Image Processing*, vol. 6, no. 2, pp. 298–311, 1997.
- [28] T. Karras, M. Aittala *et al.*, “Elucidating the Design Space of Diffusion-Based Generative Models,” Oct. 2022, arXiv:2206.00364 [cs, stat].
- [29] Z. Geng, W. Luo *et al.*, “Consistency models made easy,” 2024.
- [30] P. Ramachandran, B. Zoph *et al.*, “Searching for activation functions,” in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Workshop Track Proceedings*. OpenReview.net, 2018.
- [31] P. Esser, S. Kulal *et al.*, “Scaling rectified flow transformers for high-resolution image synthesis,” *arXiv preprint arXiv:2403.03206*, 2024.
- [32] L. Liu, H. Jiang *et al.*, “On the variance of the adaptive learning rate and beyond,” in *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.
- [33] D. Bogdanov, M. Won *et al.*, “The mtg-jamendo dataset for automatic music tagging,” in *Machine Learning for Music Discovery Workshop, International Conference on Machine Learning (ICML 2019)*, Long Beach, CA, United States, 2019.
- [34] H. Dubey, V. Gopal *et al.*, “Icassp 2022 deep noise suppression challenge,” in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2022, Virtual and Singapore, 23-27 May 2022*. IEEE, 2022, pp. 9271–9275.
- [35] A. Agostinelli, T. I. Denk *et al.*, “MusicLM: Generating Music From Text,” Jan. 2023, arXiv:2301.11325 [cs, eess].
- [36] M. Pasini, M. Grachten *et al.*, “Bass accompaniment generation via latent diffusion,” in *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 1166–1170.
- [37] J. L. Roux, S. Wisdom *et al.*, “SDR - half-baked or well done?” in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2019, Brighton, United Kingdom, May 12-17, 2019*. IEEE, 2019, pp. 626–630.
- [38] A. Hines, J. Skoglund *et al.*, “Visqol: an objective speech quality model,” *EURASIP J. Audio Speech Music. Process.*, vol. 2015, p. 13, 2015.
- [39] C. Sloan, N. Harte *et al.*, “Objective assessment of perceptual audio quality using visqolaudio,” *IEEE Trans. Broadcast.*, vol. 63, no. 4, pp. 693–705, 2017.
- [40] M. Chinen, F. S. C. Lim *et al.*, “Visqol v3: An open source production ready objective speech and audio metric,” in *Twelfth International Conference on Quality of Multimedia Experience, QoMEX 2020, Athlone, Ireland, May 26-28, 2020*. IEEE, 2020, pp. 1–6.
- [41] K. Kilgour, M. Zuluaga *et al.*, “Fréchet audio distance: A reference-free metric for evaluating music enhancement algorithms,” in *20th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, Sep. 2019, pp. 2350–2354.
- [42] Y. Wu, K. Chen *et al.*, “Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation,” in *IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP 2023, Rhodes Island, Greece, June 4-10, 2023*. IEEE, 2023, pp. 1–5.
- [43] M. Tailleur, J. Lee *et al.*, “Correlation of fréchet audio distance with human perception of environmental audio is embedding dependant,” *arXiv preprint arXiv:2403.17508*, 2024.
- [44] R. Castellon, C. Donahue *et al.*, “Codified audio language modeling learns useful representations for music information retrieval,” in *Proceedings of the 22nd International Society for Music Information Retrieval Conference, ISMIR 2021, Online, November 7-12, 2021*, 2021, pp. 88–96.
- [45] L. Pepino, P. Riera *et al.*, “Encodecmae: Leveraging neural codecs for universal audio representation learning,” *arXiv preprint arXiv:2309.07391*, 2023.
- [46] P. Alonso-Jiménez, L. Pepino *et al.*, “Leveraging pre-trained autoencoders for interpretable prototype learning of music audio,” *arXiv preprint arXiv:2402.09318*, 2024.

- [47] Y. Li, R. Yuan *et al.*, “Mert: Acoustic music understanding model with large-scale self-supervised training,” 2023.
- [48] D. Wolff, S. Stober *et al.*, “A systematic comparison of music similarity adaptation approaches,” in *Proceedings of the 13th International Society for Music Information Retrieval Conference, ISMIR 2012, Mosteiro S.Bento Da Vitória, Porto, Portugal, October 8-12, 2012*. FEUP Edições, 2012, pp. 103–108.
- [49] Ángel Faraldo, “Beatport edm key dataset,” Jan. 2018.
- [50] C. Emanuele, D. Ghisi *et al.*, “TinySOL: an audio dataset of isolated musical notes,” Jan. 2020.
- [51] J. Pons and X. Serra, “musicnn: Pre-trained convolutional neural networks for music audio tagging,” *arXiv preprint arXiv:1909.06654*, 2019.
- [52] J. Spijkervet and J. A. Burgoyne, “Contrastive learning of musical representations,” in *Proceedings of the 22nd International Society for Music Information Retrieval Conference, ISMIR 2021, Online, November 7-12, 2021*, 2021, pp. 673–681.
- [53] C. Plachouras, “Beyond Benchmarks: A Toolkit for Music Audio Representation Evaluation,” Ph.D. dissertation, Universitat Pompeu Fabra, Sep. 2023.
- [54] C. Plachouras, P. Alonso-Jiménez *et al.*, “mir_ref: A representation evaluation framework for music information retrieval tasks,” in *37th Conference on Neural Information Processing Systems (NeurIPS), Machine Learning for Audio Workshop*, New Orleans, LA, USA, 2023.

ROBUST AND ACCURATE AUDIO SYNCHRONIZATION USING RAW FEATURES FROM TRANSCRIPTION MODELS

Johannes Zeitler, Ben Maman and Meinard Müller
International Audio Laboratories Erlangen, Germany

{johannes.zeitler, ben.maman, meinard.mueller}@audiolabs-erlangen.de

ABSTRACT

In music information retrieval (MIR), precise synchronization of musical events is crucial for tasks like aligning symbolic information with music recordings or transferring annotations between audio versions. To achieve high temporal accuracy, synchronization approaches integrate onset-related information extracted from music recordings using either traditional signal processing techniques or exploiting symbolic representations obtained by data-driven automated music transcription (AMT) approaches. In line with this research direction, our paper introduces a high-resolution synchronization approach that combines recent AMT techniques with traditional synchronization methods. Rather than relying on the final symbolic AMT results, we show how to exploit raw onset and frame predictions obtained as intermediate outcomes from a state-of-the-art AMT approach. Through extensive evaluations conducted on piano recordings under varied acoustic conditions across different transcription models, audio features, and dynamic time warping variants, we illustrate the advantages of our proposed method in both audio–audio and audio–score synchronization tasks. Specifically, we emphasize the effectiveness of our approach in aligning historical piano recordings with poor audio quality. We underscore how additional fine-tuning steps of the transcription model on the target dataset enhance alignment robustness, even in challenging acoustic environments.

1. INTRODUCTION AND RELATED WORK

Aligning different versions of a musical piece is a common task in music information retrieval (MIR). For example, score–audio synchronization with the objective to align score-based note information with time positions of an audio recording is used in automatic score following [1, 2], score-informed audio decomposition techniques [3], or the derivation of note labels for the training and evaluation of automated music transcription (AMT) systems [4]. Aligning different audio recordings of the same musical piece (audio–audio synchronization) enables applications

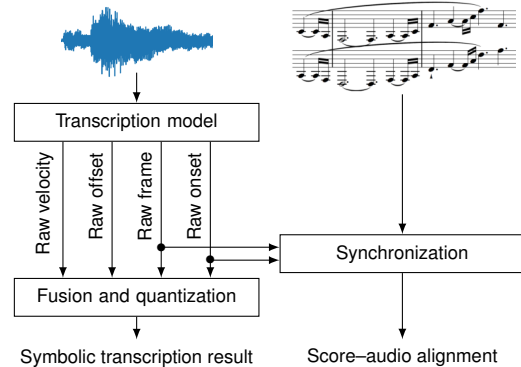


Figure 1: Schematic overview of the proposed audio–score synchronization pipeline using raw features from a transcription model.

like track switching [5], cross-version analysis [6], automated accompaniment of instrumentalists using existing backing tracks [7, 8], and the transfer of annotations from one recording to another [9, 10].

Alignment pipelines based on dynamic time warping (DTW) typically use chroma or onset features, or a combination of both [11, 12]. While such features can easily be obtained from symbolic score information, they need to be estimated from audio recordings. Traditionally, many alignment pipelines rely on features estimated with classical signal processing methods, e.g., using a constant-Q transform [13] or a multirate filterbank [11, 14]. With the advancements in deep learning (DL) techniques, several systems for multi-pitch estimation (MPE) [15–20] as well as learning-based methods for onset estimation [21–24] have been introduced. Along with the creation of large datasets of pairs of audio recordings and note labels such as MAESTRO [25] or MusicNet [16], modern transcription models precisely estimate note on- and offset, as well as velocity and pedaling information [26–28].

In this work, we investigate the advantages of using features estimated by AMT systems for audio–audio and audio–score alignment tasks. We demonstrate how to leverage intermediate predictions from transcription models for aligning audio recordings and symbolic representations, as illustrated in Figure 1. In particular, we investigate alignments within a carefully curated dataset of the first movements of the 32 piano sonatas by Ludwig van Beethoven, with all sonatas performed by eleven artists, encompassing live performances, historic recordings with low audio quality, performances on historic instruments,



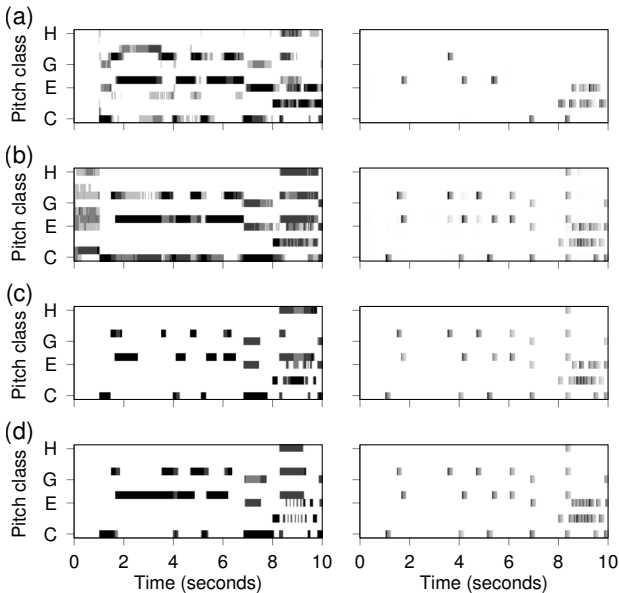


Figure 2: Chroma (left) and DLNCO (right) features for Beethoven’s Appassionata (Op. 57-1) played by F. Dupree. (a) FB_RAW. (b) T1_RAW. (c) T1_SYM. (d) DK_SYM.

and modern studio-quality recordings. By analyzing the alignment precision and robustness across different feature extractors and representations, synchronization algorithms, and audio versions, we demonstrate that our approach allows for robust synchronization of real-world data.

The outline of this paper is as follows. First, in Section 2, we describe the synchronization pipeline and discuss its components, followed by an introduction to our dataset of Beethoven’s piano sonatas in Section 3. Since there are no reference alignments available for the audio recordings in this real-world dataset, we rely on heuristics for evaluation, see Section 4. In Section 5, we experimentally show how using raw intermediate features from transcription models increases the alignment stability, as compared to using the final symbolic transcription results. Furthermore, we give detailed insights into the peculiarities of aligning datasets encompassing historic music recordings and demonstrate how to adapt to such data. We conclude in Section 6 with an outlook on future work.

2. SYNCHROZINATION PIPELINE

In this section, we provide an overview of the synchronization pipeline. First, we describe the feature extractors and types of feature representations, before we outline the DTW-based alignment step. An overview of all elements in the pipeline and their abbreviations is provided in Table 1. Following the notation in [26], we distinguish between two types of features: *frame* features encode when a note is active (in the piano case, this corresponds to the time until a key is released, or until the sustain phase ends), and *onset* features encode only the beginning of a note (in the piano case when a key is pressed).

Feature Extraction Model	
FB	Filterbank
T1	Onsets and frames transcription model [4]
T2	High-resolution transcription model [27]
DK	Disklavier
Feature Representation	
RAW	Continuous pitch and onset probabilities
SYM	Thresholded and discretized pitches and onsets
Alignment Technique	
O	Onset features using standard DTW
OF	Onset and frame features using MrMsDTW from [14]

Table 1: Overview of short notation for all components of the processing chain.

2.1 Feature Extraction Model

Filterbank. Before the advent of today’s DL-based feature extraction models, traditional signal processing techniques were a common way to extract features from audio recordings. For example, the standard implementation of Sync Toolbox [14] uses a multirate filterbank (FB) to estimate frame-wise note activity and onsets.

Transcription Model. In recent years, AMT systems based on DL have shown significant improvements in performance [29]. One of the ground-breaking architectures is the *Onsets and Frames* architecture by Hawthorne et al. [26], which has separate prediction heads to estimate onsets and frames. Maman et al. [4] proposed a strategy to train a model based on the onsets and frames architecture on diverse and unaligned pairs of audio data and musical scores. This has led to improved performance on unseen datasets, generalizing across instrumentations, acoustic conditions, and styles. In the following, we refer to the transcription model from [4], trained on MusicNet [16] with re-aligned labels, as T1. Kong et al. [27] extend the onsets and frames architecture by additionally modeling sustain pedal activity, therefore providing more robust training in the presence of misaligned offset information. We refer to the transcription system from [27], trained on MAESTRO [25], as T2.¹

Disklavier. Certain datasets such as MAESTRO [25] include pairs of audio recordings and reference note information by having the pieces performed on a Disklavier. We refer to features directly extracted from the symbolic Disklavier track as DK, and use them as an upper bound for the performance of an MPE feature extractor.

2.2 Feature Representation

Raw features. For each sequence of input audio, the feature extractors FB, T1, and T2 predict continuous pitch- and frame-wise probabilities $\mathbf{P}_{raw}^{frame}, \mathbf{P}_{raw}^{onset} \in [0, 1]^{88 \times N}$ for frame activity and note onset, respectively. These feature matrices can be thought of as RAW features and are commonly stored in a pianoroll-like representation for 88 pitches and N time frames (T1 and T2 additionally predict

¹ Note that we do not include transcription models that directly output a tokenized sequence of MIDI messages (where we can not access raw pitch probabilities), such as the MT3 model by Hawthorne et al. [30].

such probabilities for note velocity and offset). Figure 2a/b illustrates RAW features for a piece from the ASAP dataset [31], computed by the FB and T1 feature extractors .

Symbolic features. In AMT, the raw predictions for frames, onsets, and offsets are fused and quantized in a postprocessing step, yielding binary estimates about which keys have been pressed. In particular, note sustain (frame activity) is conditioned on a previously occurring note onset [26,27]. This postprocessing step outputs a sequence of symbolic control messages for note onset and offset events, with additional control messages for pedal information in the case of T2. We denote these binary and symbolic-like features as SYM features and, for usage in our synchronization pipeline, store these in the form of two discretized pianorolls $\mathbf{P}_{\text{sym}}^{\text{frame}}, \mathbf{P}_{\text{sym}}^{\text{onset}} \in \{0, 1\}^{88 \times N}$ for frame activity and onset events, respectively. Note that in the case of Disklavier (DK), no RAW features are available and thus only SYM features are used. Figure 2c/d illustrates SYM features for T1 and DK.

Comparison. In Figure 2 we qualitatively compare RAW features from FB and T1 as well as SYM features from T1 to the DK reference. While FB_RAW in Figure 2a shows many false positive chroma events and misses many onsets compared to DK in Figure 2d, the chroma features of T1_RAW in Figure 2b are relatively stable and onsets perfectly coincide with DK. Thresholding the RAW transcription results to T1_SYM features (Figure 2c) yields varying and often shortened note durations in the chroma representation compared to DK, indicating possible instabilities when using these features for computing an alignment.

2.3 Alignment Technique

We use two variants of DTW to compute the optimal alignment between two feature sequences.

Onset features. As a first approach and in line with previous work [4, 32], we use only onset features and convert them to a twelve-dimensional pitch class representation. Using the Euclidean distance function, we compute the cost matrix between the onset feature sequences of the two versions to be aligned. We use standard DTW with unit steps in the horizontal, vertical, and diagonal direction with step weights (1.5, 1.5, 2) to compute the minimum cost path between the two sequences [12]. We refer to this approach, using only onset features, as \circ .

Onset and frame features. As a second alignment variant, we choose a high-resolution approach [11] that combines frame and onset features. Using frame features yields robustness on the coarse temporal level, while onset features provide precision on the fine level by precisely aligning note onsets [24]. In this approach, we again convert frame and onset features into pitch class representations and additionally add a decay to the onset features. We refer to [11] for a description of these decaying locally normalized chroma onset (DLNCO) features. Next, we compute separate cost matrices for frame features (using the cosine distance) and for onset features (using the Euclidean distance). Afterward, we add the two cost matrices for frame and onset features and use DTW with step weights

ID	Performer	Year	Duration
AS35	Artur Schnabel	1935	03:33:35
FG58	Friedrich Gulda	1958	03:34:00
FJ62	Fritz Jank	1962	03:41:26
WK64	Wilhelm Kempff	1964	03:45:31
FG67	Friedrich Gulda	1967	03:25:02
VA81	Vladimir Ashkenazy	1981	03:46:27
DB84	Daniel Barenboim	1984	03:58:37
JJ90	Jeno Jando	1990	03:39:14
AB96	Alfred Brendel	1996	03:52:28
MB97	Malcolm Bilson et al.	1997	03:46:08
MC22	Muriel Chemin	2022	04:05:11
Total			41:07:45

Table 2: Overview of audio versions in the BPSD. The versions with identifiers AS35, FG58, FJ62, and WK64 are in the public domain and are freely accessible within the BPSD. Durations given in hh:mm:ss.

(1.5, 1.5, 2) to compute the optimum alignment path on the combined cost matrix. We refer to [14, 33] for an efficient multi-resolution and multi-scale implementation of DTW. We denote the described approach, using a combination of onset and frame features, as $\circ\text{F}$. Note that we do not consider using only frame features (commonly called chroma features), as previous work has shown a lack of precision in this case. For example, Ewert et al. observe a 100% increase of the alignment error when using frame features instead of combined frame and onset features for the case of piano music, where onsets are clearly defined [11].

3. DATASETS

In our experiments, we consider the case of piano music, as there are large-scale datasets available [25, 31, 34], note onsets are well-defined, reference note information can be obtained from performances on a Disklavier, and many transcription models are primarily trained on piano music [26, 27]. To this end, we evaluate alignment accuracy not only in acoustically controlled scenarios such as MAESTRO. Instead, we consider a much more challenging scenario using real-world piano recordings under complex acoustic conditions, which we find in a dataset of Beethoven’s piano sonatas [35]. The 32 piano sonatas by Ludwig van Beethoven are recognized as pivotal works in Western classical music and hold a significant place in cultural history. Being one of the most performed and recorded corpus of classical music, alignments between a multitude of different versions can be studied.

3.1 Beethoven Piano Sonata Dataset

As a main evaluation corpus, we choose the Beethoven Piano Sonata Dataset (BPSD) [35], which comprises eleven complete audio recordings of the first movements of all 32 piano sonatas, along with sheet music in machine-readable format. An aspect of central importance is the coherent structure of the dataset: all audio versions and the symbolic sheet music share the same musical timeline, which was enforced by manually editing the score and audio versions. Thus, there is no incoherence due to, e.g., additional

or missing repetitions. The BPSD includes over 41 h of audio recorded under various acoustic conditions, being far more diverse than common piano datasets [25, 34]. For example, MAESTRO was entirely performed on Yamaha Disklaviers, and training on MAESTRO does not provide good generalization on other datasets [4, 32, 36]. In contrast, the BPSD comprises modern studio recordings in high audio quality, vintage recordings published on vinyl, including pitch drift due to wobbling of the vinyl records, performances on historical instruments such as the fortepiano, and significant deviations from today’s standard tuning frequency of 440 Hz (A4). Measure positions were annotated manually for all 32 sonatas recorded by Wilhelm Kempff in 1964 (WK64). An overview of the eleven audio versions in the BPSD is provided in Table 2.

3.2 ASAP

To be able to use reference note information from Disklavier recordings in our experiments, we additionally leverage the ASAP dataset [31]. To achieve consistency across all experiments, we identify the performances of the first movements of Beethoven’s piano sonatas in ASAP which share the same structure as recordings in the BPSD. This subset consists of 13 individual recordings with a total length of 103 min.

4. QUANTIFYING SYNCHRONIZATION ACCURACY

In this section, we describe the heuristics used to assess the accuracy of our score–audio and audio–audio synchronization pipelines. We refer to [37] for a detailed discussion about the analysis of synchronization accuracy without ground-truth annotations.

4.1 Notation

We first introduce some notation for aligning time points between two different versions V_1 and V_2 of a piece. We assume that these versions have continuous time axes $[0, T_1]$ and $[0, T_2]$, which can either be in physical time (for audio recordings, in seconds) or in musical time (for score-related data, in measures). From the alignment algorithms described in Section 2.3, we obtain a monotonous mapping function $\mathcal{M}^{V_1 \rightarrow V_2} : [0, T_1] \rightarrow [0, T_2]$ to transfer time instants from the timeline of one version to the other. Note that even though the alignment result obtained from DTW maps discrete time axes, our assumption of having continuous time axes can be obtained by using suitable interpolation techniques, see [37].

4.2 BPSD: Measure Transfer

In the following, we consider three versions: $V_1 = S$ being a score, and $V_2 = A_1$ and $V_3 = A_2$ being different audio versions of the same piece. We choose A_2 to be the recordings by Wilhelm Kempff (WK64), for which we have access to manually annotated measure positions t_{A_2} . Using audio–audio synchronization, we obtain a mapping $\mathcal{M}^{A_2 \rightarrow A_1}$ to transfer these measure positions to the first

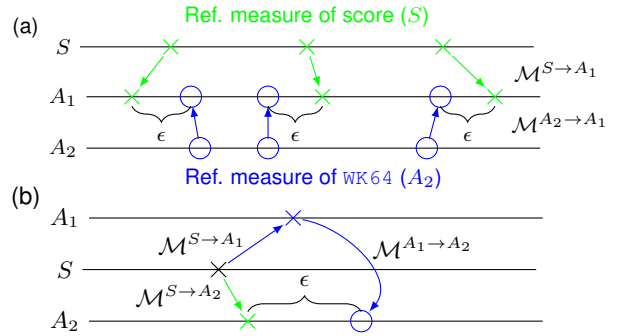


Figure 3: Schematic illustration of (a) measure transfer and (b) note onset transfer heuristics.

audio version A_1 . Similarly, we transfer measure positions t_S obtained from the score S to the first audio A_1 using a mapping $\mathcal{M}^{S \rightarrow A_1}$. In a last step, as illustrated in Figure 3a, we calculate the absolute error of the measure positions transferred from S and A_2 :

$$\epsilon = \left| \mathcal{M}^{S \rightarrow A_1}(t_S) - \mathcal{M}^{A_2 \rightarrow A_1}(t_{A_2}) \right|. \quad (1)$$

4.3 ASAP: Note Onset Transfer

In order to evaluate score–audio synchronization on the ASAP dataset, we can not resort to the heuristic described in Section 4.2, as there are no reliable manual measure annotations available. Therefore, we use an approach that transfers note onsets, illustrated in Figure 3b, to assess the synchronization accuracy.

First, we obtain audio features for two different audio recordings A_1, A_2 of the same piece, as well as features for the score S . For each version, we perform score–audio synchronization to obtain alignment functions $\mathcal{M}^{S \rightarrow A_1}$ and $\mathcal{M}^{S \rightarrow A_2}$ from musical to physical time. Using these mapping functions, we map note event onsets t_S in musical time from the score to the physical time of the audio recordings. In a second step, we transfer the aligned onset positions from the first to the second audio using audio–audio synchronization via the mapping function $\mathcal{M}^{A_1 \rightarrow A_2}$ and compute the absolute error

$$\epsilon = \left| \mathcal{M}^{S \rightarrow A_2}(t_S) - \mathcal{M}^{A_1 \rightarrow A_2}(\mathcal{M}^{S \rightarrow A_1}(t_S)) \right| \quad (2)$$

between these transferred time points and the ones obtained from score–audio synchronization.

In both heuristics, we assume that the synchronization accuracy is high if the time instances transferred via two different branches have small deviations. Note that this is only a *necessary* and not a *sufficient* condition for alignment quality; nevertheless, this metric gives a good indicator of the alignment performance (see also [37]).

5. EVALUATION

While our main focus is on the BPSD due to its realism and diversity, we first analyze synchronization accuracy on the ASAP dataset in order to compare features estimated from audio recordings to those derived from the Disklavier reference. In the next step, we evaluate the alignment performance on the BPSD across all audio versions. Finally, we

Feature	Mean	Median	Conf. 90	Conf. 95
T1_SYM_O	89	0	281	467
T1_SYM_OF	66	12	153	293
T1_RAW_OF	29	8	64	146
T2_SYM_O	31	0	102	192
T2_SYM_OF	22	2	49	111
T2_RAW_OF	21	7	43	99
DK_SYM_O	37	0	130	216
DK_SYM_OF	25	2	57	123
FB_RAW_OF	64	20	146	268

Table 3: ASAP: Absolute error in milliseconds for note onset transfer heuristic.

conduct a detailed analysis of the performance on individual versions, identify problematic recordings, and illustrate how to improve alignment robustness by adapting a transcription model to the target data.

5.1 ASAP: Estimated Features vs. Reference Notes

First, we evaluate synchronization accuracy on the ASAP dataset using our note onset transfer approach described in Section 4.3. For each pair of audio files, we calculate the mean, median, 90 and 95 percentiles of the absolute alignment error in ms, and report averaged results in Table 3.

Analyzing the median absolute error, which we consider an indicator for the achievable accuracy under average conditions, we find perfect alignments ($\epsilon = 0$ ms) for at least 50% of all note onsets when using only symbolic onset features (SYM_O) for the transcription models T1 and T2, as well as the Disklavier DK. To assess the methods’ robustness and the severeness of outliers, we next investigate the 95% quantiles of the absolute alignment error. Using only symbolic onset features and standard DTW (SYM_O) yields the highest errors for all T1 (467 ms), T2 (192 ms), and DK (216 ms) variants, indicating a lack of robustness despite excellent median accuracy.

In the next step, we jointly use the frame and onset information from the symbolic features (SYM_OF) and observe a slight rise in the median error to 12 ms for T1, and to 2 ms for T2 and DK, respectively. While this indicates that the best achievable precision slightly deteriorates, we monitor a significant reduction of the 90% and 95% confidence intervals by approximately 50% for all SYM_OF variants, indicating a vastly improved robustness towards outliers when combining frame and onset features in the computation of the alignment.

Lastly, we directly use the intermediate predictions for frames and onsets (RAW_OF) in our alignment pipeline. While the median absolute error is comparable to the one based on symbolic features, we again observe a significant decrease in the 90% and 95% confidence intervals. Using RAW_OF features in the T2 transcriber yields the lowest mean (21 ms) and confidence intervals (43 and 99 ms), even outperforming the usage of reference note informa-

Feature	Mean	Median	Conf. 90	Conf. 95
T1_SYM_O	66	17	115	234
T1_SYM_OF	52	20	102	160
T1_RAW_OF	41	12	70	121
T2_SYM_O	109	20	272	466
T2_SYM_OF	56	14	138	251
T2_RAW_OF	47	15	97	207
FB_RAW_OF	44	20	70	128

Table 4: BPSD: Absolute error in milliseconds for measure transfer heuristic.

tion obtained from the Disklavier.² We illustrate this finding with the intuitive example of a chord where notes are not played simultaneously, either due to a playing mistake or as a stylistic element, leading to a deviation of symbolic and actually performed note order. While the DK features strictly assign each note onset to one particular time frame and thus cause alignment instabilities in the given example, the continuous RAW predictions can smoothly cover neighboring time frames and thus allow for a robust alignment.

5.2 BPSD: General Performance

Next, we analyze the overall matching of score–audio and audio–audio synchronization on the more realistic and more diverse BPSD by using the measure-transfer heuristic as described in Section 4.2. In Table 4, we again report the mean, median, 90 and 95% confidence intervals for the absolute error between measure positions obtained from score–audio and audio–audio transfer.

Analyzing the median absolute error in Table 4, all features yield a precision between approximately 12 ms and 20 ms, without a clear tendency towards one particular method. However, it is the robustness (measured by the 90% and 95% confidence intervals) where we find a clear trend: using only onsets from symbolic features (SYM_O) yields large alignment outliers, with the 95% confidence interval of the absolute error being 234 ms for T1 and even 466 ms for T2. Using additional frame features (SYM_OF) lowers the 95% confidence interval to 160 ms and 251 ms, respectively. In line with our observations on the ASAP dataset, using intermediate transcription results (RAW_OF) further reduces the mean as well as the confidence intervals for both transcription models. The T1 transcriber, which was trained on audio from the acoustically diverse MusicNet [16] dataset, exhibits significantly lower errors than T2 (121 ms vs. 207 ms for the 95% conf. interval), which was trained only on MAESTRO. While using filterbank features (FB) resulted in relatively high errors on ASAP, on the BPSD we observe metrics that are similar to those of the T1 transcriber, and considerably better than those of the T2 model. This indicates a lack of robustness of the DL-based transcription models on the diverse acoustic conditions of the BPSD, which we will investigate and mitigate in the following section.

² We note that the T2 model was trained on MAESTRO [25], which is the basis of ASAP [31]. Therefore, a separation for train and test data is not guaranteed for T2 in the experiments on ASAP. However, the DK features nevertheless are the upper limit of the achievable transcription accuracy.

5.3 BPSD: Detailed Analysis and Finetuning

To further investigate why the alignment pipelines using transcription features ($T1, T2$) do not yield significantly better results on the BPSD than those features using the filterbank baseline (FB), we further break down the investigation to the BPSD’s individual audio versions. Transcription models (and DL systems in general) are known to exhibit a degraded performance when there is a domain shift between the test data and the training data [32, 36]. Such effects can be caused by poor audio quality in general, or, for music data, by a difference in timbre or tuning.

Identifying problematic versions. We restrict our analysis to raw frame and onset features (RAW_OF) from the filterbank (FB) and the $T1$ transcriber, as these showed the overall most robust results on the complete BPSD (see Section 5.2) and illustrate the median and 95% confidence intervals for all audio versions of the BPSD in Table 5. Analyzing the 95% confidence values for $T1_RAW_OF$ in Table 5, we identify two problematic versions, namely the 1958 recordings by Friedrich Gulda (FG58) with 788 ms and the 1997 recordings by Malcolm Bilson et al. (MB97) with 146 ms. By inspection of the recorded pieces, we find two different reasons for the alignment instabilities.

Musical and acoustic reasons for instabilities. Friedrich Gulda recorded his first cycle of Beethoven’s Piano Sonatas (FG58) over a relatively long time span between 1950 and 1958, playing different pianos in different environments. Among the FG58 recordings, we identify Sonata No. 26 (“Les adieux”) and No. 29 (“Hammerklavier”) as especially problematic, showing large differences in tuning, along with high background noise.

The pianist Malcolm Bilson (MB97) is committed to historically informed performance practice. His interpretations on historical instruments introduce a novel approach to performance in an era predominantly defined by the use of modern instruments. Malcolm Bilson and colleagues recorded their 1997 cycle of Beethoven’s Piano Sonatas on nine fortepianos, including original historical instruments. Compared to modern pianos, the overall sound of fortepianos is significantly different, due to different mechanics, strings, and resonance bodies. Furthermore, the timbre varies across registers, e.g., bass notes sound fundamentally different compared to high-octave notes, and the reference pitch deviates from today’s standard of 440 Hz (A4). In summary, these deviations in timbre, tuning, and recording noise lead to a so-called “domain shift”, i.e., the FG58 and MB97 recordings are not close enough to the transcriber’s training data. As a result, the model’s predictions are highly unstable and do often not correspond to the actually played notes.

Fine-tuning the transcriber. Despite the aforementioned issues, our goal is to obtain highly accurate alignments on the BPSD. Therefore, we choose to adapt the $T1$ transcriber to the BPSD’s audio versions by fine-tuning the model on the target data itself. Note that this is a valid procedure for the purpose of this study, as we do not evaluate the transcription accuracy itself, and we only use unaligned pairs of audio and score data for finetuning (see [4] for a

Version	FB_RAW_OF		T1_RAW_OF		T3_RAW_OF	
	med.	cf 95.	med.	cf. 95	med.	cf. 95
AB96	19	132	11	58	12	58
AS35	20	157	11	62	12	76
DB84	20	149	12	71	14	93
FG58	19	137	27	788	12	80
FG67	20	138	10	49	11	59
FJ62	22	185	11	80	13	84
JJ90	17	102	10	56	12	56
MB97	23	217	12	146	12	103
MC22	22	178	13	80	14	89
VA81	19	143	11	61	12	69
WK64	16	99	10	49	11	45
average	20	128	12	121	12	62

Table 5: Median and 95% confidence interval of the absolute synchronization error for individual performances in the BPSD. All experiments use raw frame and onset features (RAW_OF). Values are given in milliseconds.

detailed description of the training process using unaligned pairs of audio and score data). Therefore, we do not overfit the model towards a reference alignment. We denote the fine-tuned transcriber as $T3$.

Results with fine-tuned transcriber. After fine-tuning on the BPSD, the $T3$ model significantly improves the 95% confidence interval for the two problematic versions FG58 and MB97 from 788 ms to 80 ms and from 146 ms to 103 ms, respectively. For all other audio versions, the median absolute error and the 95% confidence interval of the fine-tuned transcriber $T3$ remain in a similar range as the original model $T1$. We note that the averaged 95% confidence interval of 62 ms for the fine-tuned transcriber $T3$ is in the range of the typical tolerance in beat-tracking applications (70 ms), making the proposed synchronization approach with raw features even useful for the creation of datasets with high demands regarding timing.

6. CONCLUSION AND OUTLOOK

In this paper, we analyzed audio synchronization using raw features from transcription models. By conducting quantitative analysis on two different datasets of piano music, we show that the amount of alignment outliers is vastly reduced when using raw instead of symbolic features. We put a particular emphasis on the analysis of synchronization robustness of real-world audio recordings including historic instruments and recordings of low quality, and outline which acoustic conditions lead to alignment mismatch. By fine-tuning a transcription model on the target dataset and using the predicted raw features, we achieve synchronization accuracy that enables usage of the datasets even in time-critical applications such as beat tracking. As the raw features are computed anyway when using transcription models, we propose to use these raw features by default in synchronization pipelines. While raw features from transcription models yield excellent synchronization robustness for piano music, a yet unanswered question that we plan to address in future work is the performance in other genres, e.g., vocal or orchestral music.

7. ACKNOWLEDGEMENTS

This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Grant No. 521420645 (MU 2686/17-1) and Grant No. 500643750 (MU 2686/15-1). The International Audio Laboratories Erlangen are a joint institution of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) and Fraunhofer Institute for Integrated Circuits IIS.

8. REFERENCES

- [1] D. Schwarz, N. Orio, and N. Schnell, “Robust polyphonic midi score following with hidden Markov models,” in *International Computer Music Conference (ICMC)*, Miami, Florida, USA, 2004.
- [2] A. Arzt, G. Widmer, and S. Dixon, “Automatic page turning for musicians via real-time machine listening,” in *ECAI 2008 - 18th European Conference on Artificial Intelligence, Patras, Greece, July 21-25, 2008, Proceedings*, ser. Frontiers in Artificial Intelligence and Applications, vol. 178. IOS Press, 2008, pp. 241–245.
- [3] S. Ewert, B. Pardo, M. Müller, and M. Plumbley, “Score-informed source separation for musical audio recordings: An overview,” *IEEE Signal Processing Magazine*, vol. 31, no. 3, pp. 116–124, April 2014.
- [4] B. Maman and A. H. Bermamo, “Unaligned supervision for automatic music transcription in the wild,” in *Proceedings of the International Conference on Machine Learning (ICML)*, 2022, pp. 14 918–14 934.
- [5] F. Zalkow, S. Rosenzweig, J. Graulich, L. Dietz, E. M. Lemnaouar, and M. Müller, “A web-based interface for score following and track switching in choral music,” in *Demos and Late Breaking News of the International Society for Music Information Retrieval Conference (ISMIR)*, Paris, France, 2018.
- [6] C. Dittmar, B. Lehner, T. Prätzlich, M. Müller, and G. Widmer, “Cross-version singing voice detection in classical opera recordings,” in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Málaga, Spain, October 2015, pp. 618–624.
- [7] R. B. Dannenberg, “An on-line algorithm for real-time accompaniment,” in *Proceedings of the International Computer Music Conference (ICMC)*, Paris, France, 1984, pp. 193–198.
- [8] Y. Özer, S. Schwär, V. Arifi-Müller, J. Lawrence, E. Sen, and M. Müller, “Piano concerto dataset (PCD): A multitrack dataset of piano concertos,” *Transaction of the International Society for Music Information Retrieval (TISMIR)*, vol. 6, no. 1, pp. 75–88, 2023.
- [9] C. Weiß, F. Zalkow, V. Arifi-Müller, M. Müller, H. V. Koops, A. Volk, and H. Grohgan, “Schubert Winterreise dataset: A multimodal scenario for music analysis,” *ACM Journal on Computing and Cultural Heritage (JOCCH)*, vol. 14, no. 2, pp. 25:1–18, 2021.
- [10] C. Weiß, V. Arifi-Müller, M. Krause, F. Zalkow, S. Klauk, R. Kleinertz, and M. Müller, “Wagner Ring Dataset: A complex opera scenario for music processing and computational musicology,” *Transaction of the International Society for Music Information Retrieval (TISMIR)*, vol. 6, no. 1, pp. 135–149, 2023.
- [11] S. Ewert, M. Müller, and P. Grosche, “High resolution audio synchronization using chroma onset features,” in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Taipei, Taiwan, 2009, pp. 1869–1872.
- [12] M. Müller, *Fundamentals of Music Processing – Audio, Analysis, Algorithms, Applications*. Springer Verlag, 2015.
- [13] C. Schörkhuber and A. P. Klapuri, “Constant-Q transform toolbox for music processing,” in *Proceedings of the Sound and Music Computing Conference (SMC)*, Barcelona, Spain, 2010.
- [14] M. Müller, Y. Özer, M. Krause, T. Prätzlich, and J. Driedger, “Sync Toolbox: A Python package for efficient, robust, and accurate music synchronization,” *Journal of Open Source Software (JOSS)*, vol. 6, no. 64, pp. 3434:1–4, 2021.
- [15] R. Kelz, M. Dorfer, F. Korzeniowski, S. Böck, A. Arzt, and G. Widmer, “On the potential of simple framewise approaches to piano transcription,” in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, New York City, New York, USA, 2016, pp. 475–481.
- [16] J. Thickstun, Z. Harchaoui, and S. M. Kakade, “Learning features of music from scratch,” in *Proceedings of the International Conference on Learning Representations (ICLR)*, Toulon, France, 2017.
- [17] F. Korzeniowski and G. Widmer, “Feature learning for chord recognition: The deep chroma extractor,” in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, New York City, New York, USA, 2016, pp. 37–43.
- [18] J. Thickstun, Z. Harchaoui, D. P. Foster, and S. M. Kakade, “Invariances and data augmentation for supervised music transcription,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Calgary, Canada, 2018, pp. 2241–2245.
- [19] Y. Wu, B. Chen, and L. Su, “Polyphonic music transcription with semantic segmentation,” in *Proceedings of the IEEE International Conference on Acoustics,*

- Speech, and Signal Processing (ICASSP)*, Brighton, UK, 2019, pp. 166–170.
- [20] C. Weiß, J. Zeitler, T. Zunner, F. Schuberth, and M. Müller, “Learning pitch-class representations from score–audio pairs of classical music,” in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Online, 2021, pp. 746–753.
- [21] J. Schlüter and S. Böck, “Improved musical onset detection with convolutional neural networks,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Florence, Italy, May 2014, pp. 6979–6983.
- [22] S. Böck, A. Arzt, F. Krebs, and M. Schedl, “Online real-time onset detection with recurrent neural networks,” in *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, York, UK, September 2012.
- [23] S. Böck, F. Krebs, and G. Widmer, “Joint beat and downbeat tracking with recurrent neural networks,” in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, New York City, New York, USA, 2016, pp. 255–261.
- [24] Y. Özer, M. Istvanek, V. Arifi-Müller, and M. Müller, “Using activation functions for improving measure-level audio synchronization,” in *Proceedings of the 23rd International Society for Music Information Retrieval Conference, ISMIR 2022, Bengaluru, India, December 4-8, 2022*, 2022, pp. 749–756.
- [25] C. Hawthorne, A. Stasyuk, A. Roberts, I. Simon, C. A. Huang, S. Dieleman, E. Elsen, J. H. Engel, and D. Eck, “Enabling factorized piano music modeling and generation with the MAESTRO dataset,” in *Proceedings of the International Conference on Learning Representations (ICLR)*, New Orleans, Louisiana, USA, 2019.
- [26] C. Hawthorne, E. Elsen, J. Song, A. Roberts, I. Simon, C. Raffel, J. H. Engel, S. Oore, and D. Eck, “Onsets and frames: Dual-objective piano transcription,” in *Proceedings of the International Society for Music Information Retrieval Conference, (ISMIR)*, Paris, France, 2018, pp. 50–57.
- [27] Q. Kong, B. Li, X. Song, Y. Wan, and Y. Wang, “High-resolution piano transcription with pedals by regressing onset and offset times,” *IEEE ACM Trans. Audio Speech Lang. Process.*, vol. 29, pp. 3707–3717, 2021.
- [28] J. Gardner, I. Simon, E. Manilow, C. Hawthorne, and J. H. Engel, “MT3: multi-task multitrack music transcription,” *Computing Research Repository (CoRR)*, vol. abs/2111.03017, 2021.
- [29] E. Benetos, S. Dixon, Z. Duan, and S. Ewert, “Automatic music transcription: An overview,” *IEEE Signal Processing Magazine*, vol. 36, no. 1, pp. 20–30, 2019.
- [30] C. Hawthorne, I. Simon, R. Swavely, E. Manilow, and J. H. Engel, “Sequence-to-sequence piano transcription with transformers,” pp. 246–253, 2021.
- [31] F. Foscarin, A. McLeod, P. Rigaux, F. Jacquemard, and M. Sakai, “ASAP: a dataset of aligned scores and performances for piano transcription,” in *Proceedings of the 21th International Society for Music Information Retrieval Conference, ISMIR 2020, Montreal, Canada, October 11-16, 2020*, 2020, pp. 534–541.
- [32] X. Riley, D. Edwards, and S. Dixon, “High resolution guitar transcription via domain adaptation,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Seoul, South Korea, 2024, pp. 1051–1055.
- [33] T. Prätzlich, J. Driedger, and M. Müller, “Memory-restricted multiscale dynamic time warping,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Shanghai, China, March 2016, pp. 569–573.
- [34] V. Emiya, R. Badeau, and B. David, “Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 6, pp. 1643–1654, 2010.
- [35] J. Zeitler, C. Weiß, V. Arifi-Müller, and M. Müller, “BPSD: A coherent multi-version dataset for analyzing the first movements of Beethoven’s piano sonatas.” *Transactions of the International Society for Music Information Retrieval (TISMIR)*, submitted 2024.
- [36] D. Edwards, S. Dixon, E. Benetos, A. Maezawa, and Y. Kusaka, “A data-driven analysis of robust automatic piano transcription,” *IEEE Signal Process. Lett.*, vol. 31, pp. 681–685, 2024.
- [37] T. Prätzlich and M. Müller, “Triple-based analysis of music alignments without the need of ground-truth annotations,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Shanghai, China, March 2016, pp. 266–270.

HARNESSING THE POWER OF DISTRIBUTIONS: PROBABILISTIC REPRESENTATION LEARNING ON HYPERSPHERE FOR MULTIMODAL MUSIC INFORMATION RETRIEVAL

Takayuki Nakatsuka Masahiro Hamasaki Masataka Goto
National Institute of Advanced Industrial Science and Technology (AIST), Japan
{takayuki.nakatsuka, masahiro.hamasaki, m.goto}@aist.go.jp

ABSTRACT

Probabilistic representation learning provides intricate and diverse representations of music content by characterizing the latent features of each content item as a probability distribution within a certain space. However, typical Music Information Retrieval (MIR) methods based on representation learning utilize a feature vector of each content item, thereby missing some details of their distributional properties. In this study, we propose a probabilistic representation learning method for multimodal MIR based on contrastive learning and optimal transport. Our method trains encoders that map each content item to a hypersphere so that the probability distributions of a positive pair of content items become close to each other, while those of an irrelevant pair are far apart. To achieve such training, we design novel loss functions that utilize both probabilistic contrastive learning and spherical sliced-Wasserstein distances. We demonstrate our method’s effectiveness on benchmark datasets as well as its suitability for multimodal MIR through both a quantitative evaluation and a qualitative analysis.

1. INTRODUCTION

Multimodal representation learning of music content, such as music audio and a video [1] and music audio and text [2], has been an important topic of research, given its wide applications to Music Information Retrieval (MIR) tasks. Previous studies have typically used a deterministic approach to train encoders, where the trained encoders are utilized to map each content item to a latent space as a single vector. However, representing an arbitrary content item as a vector has various drawbacks. For example, one-to-many and many-to-many relationships need to be handled in multimodal content, such as those between an album cover image and a set of songs in that album, and between different songs that have the same title and their title text. It is difficult to represent such complex relationships in vectors. To address this challenge, *probabilistic representation learning*,

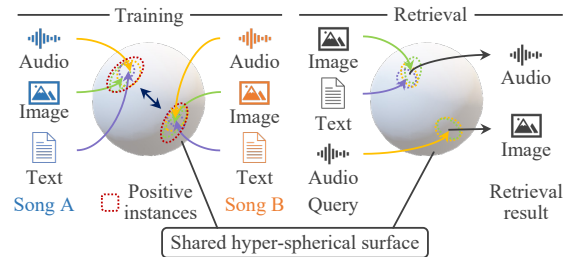


Figure 1. Probabilistic representation learning on hypersphere. **(Left)** Encoders are trained so that the probability distributions of the positive instances (music audio, an image, and text for the same song) are close to each other on the shared hyper-spherical surface, while those of irrelevant instances (different songs, artists, etc.) are far apart. **(Right)** The trained encoders are helpful for multimodal MIR. Given a single-modal query or a multimodal query such as a query that combines an image and text, our method can retrieve content items that match the query by calculating the distance between their probability distributions.

in which each content item is represented as a probability distribution in a latent space, has been studied [3–5].

Probabilistic representation learning (Figure 1) is a promising approach that can provide intricate and diverse representations by characterizing each content item as a probability distribution. This approach requires training encoders that estimate the optimal distribution for each content item. The key here is how to design an appropriate loss function for that training. In the literature, three approaches have been proposed, and in this paper, we propose a fourth approach. The first approach uses the probability product kernel [6], which calculates the expected value between distributions. This is used in probabilistic word embedding [7], face recognition [8], and image classification [9]. The second approach uses Hedged Instance Embeddings (HIB) [10]. It formulates a contrastive loss of the match probability, which calculates the distance between a pair of vectors randomly sampled from each distribution. This is used in cross-modal retrieval of text and images [3, 4], as well as in self-supervised video representation learning [11]. The third approach is to replace variables in an existing loss function (e.g., triplet loss) with probability distributions. For example, a loss function designed for deterministic methods can be calculated by using samples obtained from a Gaussian distribution [5, 12, 13] or a von Mises-Fisher distribution [14–16] via a reparameterization trick [17, 18].



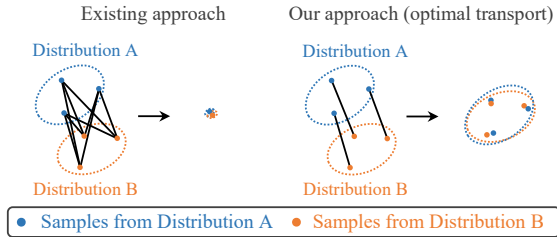


Figure 2. Advantage of optimal transport. **(Left)** To match two positive instance pairs of distributions, A and B, existing approaches ([4, 5, 16], etc.) simply calculate distances between randomly sampled pairs and cannot precisely match distributional shapes, possibly resulting in an undesirable single point when distances of positive sample pairs are minimized for probabilistic representation learning. **(Right)** Optimal transport can select optimal sample pairs to appropriately match their distributional shapes, thereby harnessing the power of rich probability distributions.

These approaches have been applied to text-to-image (or vice versa) cross-modal retrieval [3, 4], and more recently, to multimodal image retrieval [5]. Chun et al. [4] proposed Probabilistic Cross-Modal Embedding (PCME), which is a pioneering work on probabilistic representation learning for cross-modal retrieval. Li et al. [3] proposed Average Semantic Precision (ASP), which can calculate the semantic correlation scores of a dataset, and differentiable ASP approximation, which utilizes ASP as a loss function. Neculai et al. [5] proposed Multimodal Probabilistic Composer (MPC), which can use a multimodal query combining image and text for image retrieval. However, these approaches calculate distances based on sample-wise similarity, with an arbitrary sample pair randomly selected from each distribution (left side of Figure 2). This manner often results in most sample pairs being non-optimal, and as a result, the details of the distributional properties are lost. This disadvantage leads to a decrease in performance.

In light of this background, we propose two novel loss functions, one based on contrastive learning and the other on optimal transport, to be used together for multimodal MIR on a hypersphere. Contrastive learning is an effective tool to jointly map each content item of multiple modalities to a shared latent space [19, 20]. In the context of probabilistic representation learning, utilizing the angular distance between distributions has proven more effective than using their Euclidean distance [15]. Furthermore, the von Mises-Fisher (vMF) distribution (i.e., the distribution on a hypersphere) exhibits a better performance than the Gaussian distribution since vMF-based methods simplify the variance estimation by using a single scalar κ , thereby avoiding the dimension-wise estimation in Gaussian-based methods [14]. Given these insights, we propose a contrastive loss function on a hypersphere for multiple modalities based on probabilistic contrastive learning [16]. In addition, optimal transport [21] offers a robust and effective tool for calculating distances between probability distributions. It allows encoders to bring the probability distributions of a positive pair closer together, thus ensuring a more accurate representation learning (right side of Figure 2). This

unique attribute of optimal transport can benefit multimodal MIR tasks. Hence, we propose a loss function based on a Spherical Sliced-Wasserstein (SSW) [22] p -distance, contemplating the compatibility between contrastive learning and optimal transport on a hypersphere.

By using the proposed loss functions, we can train encoders that map each content item to a hypersphere, as shown in Figure 1. During training, we assume that pairwise combinations of music audio of a song, a cover image for the song, and text generated from the song’s metadata are positive, and that those for irrelevant ones (different songs, music genres, or artists, etc.) are negative (left side of Figure 1). Once the encoders are trained, we can utilize them to obtain the probabilistic representation of each content item for multimodal MIR (right side of Figure 1). The main advantage of probabilistic representation lies in its ability to seamlessly integrate multiple content items in a latent space as a *multimodal query*, which is a great benefit in retrieval tasks. We conduct both a quantitative evaluation and a qualitative analysis on the public YT8M-MusicVideo dataset and a private dataset to demonstrate the effectiveness of our proposed method.

2. PRELIMINARY

2.1 Problem Specification

We use a mel spectrogram of music audio as the input of an audio encoder, an RGB image as the input of an image encoder, and a tokenized text as the input of a text encoder. We follow previous studies [19, 20] regarding the setup of the input representations.

Let $\mathbf{A} = \{\mathbf{a}_n \in \mathbb{R}^{D^a}\}_{n=1}^N$, $\mathbf{I} = \{\mathbf{i}_n \in \mathbb{R}^{D^i}\}_{n=1}^N$, and $\mathbf{T} = \{\mathbf{t}_n \in \mathbb{R}^{D^t}\}_{n=1}^N$ be a set of spectrograms, a set of images corresponding to \mathbf{A} , and a set of tokenized texts corresponding to \mathbf{A} , respectively, where D^a is the number of dimensions of each spectrogram, D^i is the number of dimensions of each image, D^t is the number of dimensions of each tokenized text, and N is the number of songs.

Next, let $\mathbf{Z}^A = \{\mathbf{z}_n^a \in \mathbb{R}^d\}_{n=1}^N$, $\mathbf{Z}^I = \{\mathbf{z}_n^i \in \mathbb{R}^d\}_{n=1}^N$, and $\mathbf{Z}^T = \{\mathbf{z}_n^t \in \mathbb{R}^d\}_{n=1}^N$ be sets of the latent variables of spectrograms, images, and tokenized texts, respectively, where d is the number of dimensions of each latent variable.

We train the audio encoder f_A that maps \mathbf{A} to \mathbf{Z}^A , the image encoder f_I that maps \mathbf{I} to \mathbf{Z}^I , and the text encoder f_T that maps \mathbf{T} to \mathbf{Z}^T so that probability distributions $p(\mathbf{z}_n^a | \mathbf{a}_n)$, $p(\mathbf{z}_n^i | \mathbf{i}_n)$, and $p(\mathbf{z}_n^t | \mathbf{t}_n)$ are close to each other on a shared hyper-spherical surface $S_{\text{shared}}^{d-1} = \{\|\mathbf{z}_n\| = 1\}$.

2.2 Probabilistic Contrastive Learning

Contrastive learning is an established deep learning technique widely utilized in recent research [23]. In particular, methods like N -pairs loss [24], InfoNCE loss [25], and MoCo [26], which calculate the loss based on N -pairs of instances (i.e., one positive pair and $N - 1$ negative (or irrelevant) pairs), serve as powerful tools for multimodal representation learning [1, 2, 19, 20]. However, these contrastive loss functions are designed for deterministic methods and cannot be directly applied to probabilistic approaches.

Recently, Kirchof et al. [16] introduced MCInfoNCE, an adaptation of InfoNCE for probabilistic contrastive learning that uses Monte-Carlo samples from each distribution. The MCInfoNCE loss \mathcal{L}_{MC} is defined as follows:

$$\mathcal{L}_{MC} = -\frac{1}{m} \sum_{j=1}^m \sum_{l=1}^L \log \frac{e^{\text{sim}(\mathbf{z}_j^l, \mathbf{z}_+^l)/\tau}}{e^{\text{sim}(\mathbf{z}_j^l, \mathbf{z}_+^l)/\tau} + \sum_{\mathbf{z}_-} e^{\text{sim}(\mathbf{z}_j^l, \mathbf{z}_-^l)/\tau}}, \quad (1)$$

where m is a mini-batch size, L is the number of samples, τ is a hyperparameter called temperature scaling, which controls the scale of the loss function, $\mathbf{z}_n \sim p(\mathbf{z}_n)$ is an anchor, $\mathbf{z}_+ \sim p(\mathbf{z}_+|\mathbf{z}_n)$ and $\mathbf{z}_- \sim p(\mathbf{z}_-|\mathbf{z}_n)$ respectively indicate positive and negative samples of the anchor, and $\text{sim}(\cdot, \cdot)$ is a function that calculates the similarity (such as cosine similarity) between two distributions. Since MCInfoNCE is originally designed as the single-modal loss, we are the first to modify it for our multimodal loss in Section 3.1.

2.3 Optimal Transport

Optimal transport has been gaining popularity for a variety of computer vision tasks [27–29], but calculating the optimal transport distance between distributions is known to be computationally intensive [30]. This problem can be solved when the distributions are on a particular manifold [22, 30].

We therefore delve into a recent powerful innovation, the Spherical Sliced-Wasserstein (SSW) [22] p -distance, which is specialized on a hypersphere and is highly efficient and useful, but has not yet been used for representation learning.

2.3.1 Definition of Spherical Sliced-Wasserstein (SSW)

The SSW p -distance for $p \geq 1$ is defined between two probability measures $\mu, \nu \in \mathcal{P}_{p,ac}(S^{d-1})$, the set of absolutely continuous probability measures on a hypersphere S^{d-1} with a finite p -th moment, as follows:

$$SSW_p(\mu, \nu) = \int_{\mathbb{V}_{d,2}} W_p(\mu \circ P^{U^{-1}}, \nu \circ P^{U^{-1}}) d\sigma, \quad (2)$$

where $\mathbb{V}_{d,2} = \{U \in \mathbb{R}^{d \times 2}, U^\top U = I_2\}$ is the Stiefel manifold [31], σ is the uniform distribution over $\mathbb{V}_{d,2}$, P^U is the function that projects a point $\mathbf{z} \in S^{d-1}$ onto a great circle S^1 generated by U (for *a.e.* $\mathbf{z} \in S^{d-1}$, P^U can be written in a practical form of $P^U(\mathbf{z}) = \frac{U^\top \mathbf{z}}{\|U^\top \mathbf{z}\|_2}$ [22]), and W_p is the optimal transport distance on S^1 [32, 33]. To avoid any effects stemming from the choice of U , Bonet et al. [22] calculated the SSW distance several times for a set of random U , and we also calculate it in the same way.

2.3.2 Optimal Transport Distance on Great Circle

We focus on the simplest $p = 1$ in Equation (2) to calculate $W_p|_{p=1}$ between two probability measures $\mu', \nu' \in \mathcal{P}(S^1)$ that are after being projected from a hypersphere S^{d-1} onto one of the generated great circles S^1 . The W_1 is defined as

$$W_1(\mu', \nu') = \int_0^1 |F_{\mu'}(t) - F_{\nu'}(t) - \text{LevMed}(F_{\mu'} - F_{\nu'})| dt, \quad (3)$$

where $F_{\mu'}, F_{\nu'}$ are the cumulative distribution function of μ', ν' , respectively, and $\text{LevMed}(\cdot)$ is the level median [34],

defined as follows:

$$\text{LevMed}(f) = \min \left\{ \arg \min_{\alpha \in \mathbb{R}} \int_0^1 |f(t) - \alpha| dt \right\}, \quad (4)$$

where α is a shift parameter. The SSW_1 , which is utilized in our proposed loss functions (Section 3), can thus be calculated by using Equations (2)–(4). Surprisingly, we can approximate the integral in Equation (3) simply by sorting the samples on S^1 in order to calculate $F_{\mu'}, F_{\nu'}$, and $\text{LevMed}(\cdot)$. To illustrate this intuitively, the optimal sample pairing on the right of Figure 2 is dramatically expedited by this sorting on the *one-dimensional* great circle without examining many pairings. We present the algorithm and pseudocode of SSW_1 in our supplementary materials¹.

3. PROPOSED METHOD FOR MULTIMODAL MIR

We design two novel loss functions for probabilistic representation learning: a *multimodal probabilistic contrastive loss function* for multiple modalities (Section 3.1) and an *SSW-based loss function* (Section 3.2) based on optimal transport. To train the encoders as shown in Figure 1, we assign them different roles. The former loss is designed for distancing irrelevant instance pairs of probability distributions on S_{shared}^{d-1} , resulting in closer positive instance pairs. The latter loss focuses on placing positive instance pairs close to each other by matching their distributional shapes, and does not deal with irrelevant pairs at all. Their integration is therefore important. The trained encoders can be applied to multimodal MIR (Section 3.3).

The standard approach for probabilistic representation learning assumes that the latent variables of each content item have a probability distribution of a given form, such as a Gaussian distribution [5, 12, 13] or a von Mises-Fisher (vMF) distribution [14–16]. We use the vMF distribution as the probability distribution on S_{shared}^{d-1} as follows:

$$p(\mathbf{z}_n^{\mathbf{a}}|\mathbf{a}_n) = \text{vMF}(\mathbf{z}_n^{\mathbf{a}}; \mu(\mathbf{a}_n), \kappa(\mathbf{a}_n)), \quad (5)$$

$$p(\mathbf{z}_n^{\mathbf{i}}|\mathbf{i}_n) = \text{vMF}(\mathbf{z}_n^{\mathbf{i}}; \mu(\mathbf{i}_n), \kappa(\mathbf{i}_n)), \quad (6)$$

$$p(\mathbf{z}_n^{\mathbf{t}}|\mathbf{t}_n) = \text{vMF}(\mathbf{z}_n^{\mathbf{t}}; \mu(\mathbf{t}_n), \kappa(\mathbf{t}_n)), \quad (7)$$

where the variables are as defined in Section 2.1. Using the proposed loss functions, we train three encoders so that they can estimate the appropriate parameters, the mean direction $\mu(\cdot)$ and the concentration $\kappa(\cdot)$, of each vMF distribution.

During training, we utilize L samples taken from each vMF distribution via a rejection-sampling reparameterization trick [18] in practice. Our proposed loss functions in Sections 3.1 and 3.2 use the following notations:

$$\zeta_n \sim \text{vMF}(\mathbf{z}_n^*; \mu(\star_n), \kappa(\star_n)), \quad (8)$$

$$\eta_n \sim \text{vMF}(\mathbf{z}_n^*; \mu(\star_n), \kappa(\star_n)), \quad (9)$$

where ζ_n and η_n ($\star, \star \in \{\mathbf{a}, \mathbf{i}, \mathbf{t}\}, \star \neq \star$) are L samples from the vMF distribution of respective content items.

3.1 Multimodal Probabilistic Contrastive Loss Function for Probabilistic Contrastive Learning

Contrastive learning is an effective approach to jointly train encoders for the representation learning of multiple modal-

¹ <https://github.com/T39Nakatsuka/ISMIR2024>

ities [1, 2, 19, 20]. By modifying Equation (1), we design our own multimodal loss function \mathcal{L}_C for all pairwise combinations of multiple modalities (we name this *multimodal probabilistic contrastive loss*) as follows:

$$\mathcal{L}_C = -\frac{1}{m} \sum_{\langle \zeta, \eta \rangle} \sum_{j=1}^m \log \frac{e^{\text{sim}(\zeta_j, \eta_+)/\tau}}{\sum_{k=1}^m e^{\text{sim}(\zeta_j, \eta_k)/\tau}}, \quad (10)$$

where m is a mini-batch size, τ is a temperature scaling, $+$ indicates a positive sample of an anchor, and $\text{sim}(\cdot, \cdot)$ is a function that calculates the similarity between two distributions by leveraging the L samples as follows:

$$\begin{aligned} \text{sim}(\zeta_j, \eta_k) &\simeq \text{sim} \left(\left\{ \mathbf{z}_j^{*,l} \right\}_{l=1}^L, \left\{ \mathbf{z}_k^{*,l} \right\}_{l=1}^L \right) \\ &= \frac{1}{L} \sum_{l=1}^L \frac{\mathbf{z}_j^{*,l\top} \mathbf{z}_k^{*,l}}{\|\mathbf{z}_j^{*,l}\| \|\mathbf{z}_k^{*,l}\|}. \end{aligned} \quad (11)$$

This loss \mathcal{L}_C can thus distance the centroids of the distributions of irrelevant instance pairs for the contrastive learning.

3.2 SSW-based Loss Function for Optimal Transport

We formulate our SSW-based loss function \mathcal{L}_S using the SSW_1 distance (Equations (2)–(4)) as follows:

$$\mathcal{L}_S = \frac{1}{m} \sum_{\langle \zeta, \eta \rangle} \sum_{j=1}^m SSW_1(\zeta_j, \eta_j). \quad (12)$$

Intuitively, both the L samples from ζ_j and the L samples from η_j on S_{shared}^{d-1} are projected onto S^1 , sorted (paired), and used to calculate the cumulative distribution functions, resulting in the optimal transport distance between those positive instance pairs. This loss \mathcal{L}_S can thus make the distributions of positive instance pairs closer.

To leverage the advantages of both \mathcal{L}_C and \mathcal{L}_S , our method uses a loss function that integrates them as follows:

$$\mathcal{L} = \mathcal{L}_C + \lambda_S \mathcal{L}_S, \quad (13)$$

where λ_S is a weight.

3.3 Probabilistic Multimodal MIR

Once the encoders have been trained, we can leverage them to map each content item as a probability distribution on S_{shared}^{d-1} and calculate the distances between their distributions. For a single-modal query, we calculate the cosine similarity between the mean (i.e., Fréchet mean [35, 36]) over samples obtained from the distribution of a query and that of each content item in a dataset. For a multimodal query, we calculate the Fréchet mean over all samples obtained from the distribution of each query and use it like a single-modal query. When the similarity score between a pair of content items is high, it indicates that they are matched. We thus sort the similarity scores in descending order and retrieve the content item in the dataset that scored higher with respect to the query.

4. EXPERIMENTS AND RESULTS

This section describes comparison experiments to quantitatively evaluate how closely the probability distributions of

positive instances were located on S_{shared}^{d-1} , and a qualitative analysis of the proposed method to further investigate the nature of the learned representation of each content item.

4.1 Experimental Setup

4.1.1 Dataset

For the experiments, we used the following two benchmark datasets with different characteristics. We determined the size of each test set by following the setup in [1, 37].

YT8M-MusicVideo dataset [1] is a subset of the YouTube-8M dataset [38], comprising videos tagged as “music video.” We collected 73,113 triplets consisting of music audio (average length of 4 min with a 48 kHz sampling rate), its thumbnail image (an RGB image with an aspect ratio of 16:9), and its metadata including title, channel name, and upload date from 60,785 YouTube channels. We randomly split the dataset into training (64,001 songs), validation (7,112 songs), and test (2,000 songs) sets with no YouTube channels overlapping across these sets. For evaluation, we conducted our experiments three times with different seed values when training the encoders.

AS5M dataset (Album Songs 5 Million dataset) is a private dataset that contains triplets of a music audio excerpt (a 30 s audio preview for trial listening, with a 44.1 kHz sampling rate), its cover image (a square RGB image), and its metadata including song title, artist name, collection name, music genre, and release date. The dataset contains 5,920,828 audio excerpts and their metadata by 174,629 artists, and 1,115,668 cover images. Because multiple excerpts from a music album are associated with a single cover image, each image corresponds to about 5.3 excerpts on average. The songs encompass a variety of music genres (over 250). We randomly split the dataset into training, validation, and test sets with an eight-one-one ratio and with no artists or images overlapping across these sets. For evaluation, we constructed ten folds of test subsets by randomly selecting 2,000 triplets of an audio excerpt, a cover image, and a text prompt for each fold from the test set.

4.1.2 Implementation Details

Encoder architecture: We used an audio model of contrastive language-audio pretraining (CLAP) [20] as the backbone network for the audio encoder, and used image and text models of contrastive language-image pretraining (CLIP) [19] as the backbone network for the image and text encoders. Before training, we set the parameters of the pre-trained models available at Transformers [39] (i.e., “laion/clap-hsat-fused” for CLAP (audio model) and “vit_base_patch16_224” for CLIP (vision and text models)) to the encoders. During training, we updated the projection layers of the encoders.

Audio: The music audio of each song was converted to a mel spectrogram through a CLAP feature extractor available at Transformers [39], and the audio encoder was trained using the spectrogram as input. In training the audio encoder, we applied a masking technique including frequency masking and time masking [40] and a random crop

technique regarding the time domain to the spectrogram for data augmentation [41].

Image: We used an RGB image resized to $224 \text{ px} \times 224 \text{ px}$ as the input of the image encoder. In training the image encoder, we applied a random resized crop (scale=[0.08, 1.0], ratio=[0.75, 1.33]), random horizontal flip (probability=0.5), and random erasing (probability=0.2) [42] to all images for data augmentation.

Text: We tokenized text generated by using a keyword-to-caption augmentation technique [20]² with a maximum length of 77, which is the same setup as CLIP [19]. In training the text encoder, words corresponding to metadata are randomly dropped [43] at a ratio of 0.05 for each metadata.

Training: We used 16 NVIDIA A100 GPUs under each experimental condition, and each GPU computed 64 triplets of audio, images, and text per iteration. Our implementation was based on PyTorch [44]. In training the encoders, we used the Adam optimizer [45] with a learning rate of 1.0×10^{-4} . We used $d = 512$ (dimensions of latent variables) following the setup in [5]. For the vMF distribution, we set $\kappa(\cdot) \in (64, 128)$ to obtain a clear distribution following the setup of [16]. We empirically set the number of samples L to 16. For \mathcal{L}_C , we set the temperature-scaling value (Equation (10)) to $\tau = 0.07$, which was originally used in MoCo [26]. For \mathcal{L}_S , we calculated the SSW_1 distance 100 times for a set of random U , following [22] (i.e., 16 samples from ζ_j and 16 samples from η_j were projected onto 100 different great circles to match distributional shapes from 100 different views). On the basis of preliminary studies, we set the weight λ_S to 1.0.

4.1.3 Ranking-Based Evaluation Metrics

We used three standard evaluation metrics for retrieval tasks: the mean reciprocal rank (MRR) [46], the recall@ k ($R@k$), and the median rank (MR) [1]. MRR is a statistic measure utilized to evaluate the quality of retrieval results. Given a set of queries, MRR calculates the average of reciprocal ranks of the first correct (i.e., original) content item. A higher MRR value indicates a more accurate and efficient retrieval method. $R@k$ evaluates how correctly content items are retrieved in the top results. For retrieval tasks, a higher $R@k$ means that the retrieval method is more practical. We set $k = 1$ for the $R@k$ and displayed $R@1$ as a percentage. MR represents the median value of the ranks of the retrieved correct content item. In our context, a lower MR is desirable because it indicates that the correct content item is ranked closer to the top of the retrieval results.

4.2 Conditions

We compared our method (**Proposed** based on \mathcal{L}) with two competitive methods that utilize probabilistic representation learning for text-image retrieval, **PCME** [4] and **MPC** [5].

² Since the text prompt generation using a template sentence with metadata is known to be effective for retrieval tasks [19], for the YT8M-MusicVideo dataset, we generated a text prompt using: “title” is a music video uploaded by “channel name” on “upload date.” For the AS5M dataset, we generated a text prompt using: “song title” is a(n) “music genre” song by “artist name”, released on “release date.” “song title” is collected to “collection name.”

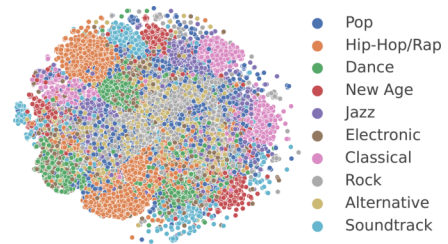


Figure 3. Visualization of the learned representations of audio, images, and text in the test subsets of the AS5M dataset with respect to music genre tags using t-SNE [51].

For music audio and other modalities, probabilistic representation learning for multimodal MIR has not yet been investigated, so we solely used the multimodal probabilistic contrastive loss \mathcal{L}_C (Section 3.1) as a baseline method (**Baseline**) in order to investigate the effectiveness of \mathcal{L}_S .

4.3 Results

As shown in Tables 1–6, our method outperformed PCME [4] and MPC [5], which are competitive methods for text-image retrieval, in all the retrieval tasks on both datasets. Likewise, our method was superior to the baseline method based on the modified MCInfoNCE [16] in nearly all retrieval tasks. We thus confirmed that \mathcal{L}_S was effective in achieving better performances. The results also showed that a multimodal query outperformed a single-modal query for most tasks. Our method can seamlessly create multimodal queries from multiple probability distributions, bringing benefits to multimodal MIR.

The performance differences between the datasets can be partly explained by their sizes since our method uses transformer models as the encoders. Several studies have shown that the performance of transformer models follows a scaling law [47–50]. This scaling law has been confirmed in experiments with data from various modalities [47–49] and in transfer learning [50]. In practice, the YT8M-MusicVideo dataset is two orders of magnitude smaller than the AS5M dataset, resulting in a decrease in performance. The performance differences between the tasks, as well as between the datasets, can also be explained on the basis of the scaling law. In our experiments, we used the CLAP audio model, which was trained on the LAION-Audio-630K dataset [20]. This dataset is several orders of magnitude smaller than the one used for training the CLIP models, which can lead to the decreased performance in audio-related retrieval tasks.

We provide additional comparison experiments that demonstrate the effectiveness of our proposed method in our supplementary materials¹.

4.4 Qualitative Analysis

We investigated the nature of the learned representations of music audio, images, and text by visualizing them regarding music genres. We utilized music audio, images, and text for 12,180 songs for the top 10 most popular genres in test subsets of the AS5M dataset. We calculated the Fréchet mean over all samples obtained from the distribution of each content item and mapped each of them to a two-dimensional

Table 1. Comparison on YT8M-MusicVideo dataset for multimodal image retrieval.

Method	Audio → Image			Text → Image			Audio & Text → Image		
	MRR ↑	R@1 (%) ↑	MR ↓	MRR ↑	R@1 (%) ↑	MR ↓	MRR ↑	R@1 (%) ↑	MR ↓
PCME	–	–	–	0.025 ± 0.003	0.73 ± 0.08	369	–	–	–
MPC	–	–	–	0.014 ± 0.001	0.2 ± 0.11	425	–	–	–
Baseline	0.024 ± 0.001	0.73 ± 0.09	272	0.048 ± 0.001	1.92 ± 0.12	166	0.044 ± 0.001	1.55 ± 0.11	166
Proposed	0.028 ± 0.001	0.65 ± 0.08	247	0.115 ± 0.0	6.68 ± 0.1	92	0.119 ± 0.002	6.8 ± 0.29	72

Table 2. Comparison on YT8M-MusicVideo dataset for multimodal text retrieval.

Method	Audio → Text			Image → Text			Audio & Image → Text		
	MRR ↑	R@1 (%) ↑	MR ↓	MRR ↑	R@1 (%) ↑	MR ↓	MRR ↑	R@1 (%) ↑	MR ↓
PCME	–	–	–	0.023 ± 0.002	0.73 ± 0.16	372	–	–	–
MPC	–	–	–	0.013 ± 0.001	0.13 ± 0.05	427	–	–	–
Baseline	0.026 ± 0.001	0.6 ± 0.18	226	0.046 ± 0.001	1.47 ± 0.1	167	0.054 ± 0.002	1.83 ± 0.3	131
Proposed	0.039 ± 0.001	1.17 ± 0.09	180	0.118 ± 0.002	6.87 ± 0.21	89	0.139 ± 0.002	7.97 ± 0.46	55

Table 3. Comparison on YT8M-MusicVideo dataset for multimodal audio retrieval.

Method	Image → Audio			Text → Audio			Image & Text → Audio		
	MRR ↑	R@1 (%) ↑	MR ↓	MRR ↑	R@1 (%) ↑	MR ↓	MRR ↑	R@1 (%) ↑	MR ↓
Baseline	0.021 ± 0.001	0.52 ± 0.05	263	0.028 ± 0.001	0.68 ± 0.08	219	0.032 ± 0.002	0.83 ± 0.26	191
Proposed	0.027 ± 0.001	0.58 ± 0.06	235	0.041 ± 0.003	1.25 ± 0.37	173	0.05 ± 0.002	1.75 ± 0.25	141

Table 4. Comparison on AS5M dataset for multimodal image retrieval.

Method	Audio → Image			Text → Image			Audio & Text → Image		
	MRR ↑	R@1 (%) ↑	MR ↓	MRR ↑	R@1 (%) ↑	MR ↓	MRR ↑	R@1 (%) ↑	MR ↓
PCME	–	–	–	0.069 ± 0.004	2.82 ± 0.34	131	–	–	–
MPC	–	–	–	0.026 ± 0.002	0.62 ± 0.15	240	–	–	–
Baseline	0.046 ± 0.002	1.37 ± 0.19	141	0.125 ± 0.005	6.21 ± 0.56	50	0.1 ± 0.004	4.39 ± 0.53	60
Proposed	0.074 ± 0.004	2.94 ± 0.46	94	0.539 ± 0.005	45.37 ± 0.65	2	0.508 ± 0.008	41.35 ± 1.12	2

Table 5. Comparison on AS5M dataset for multimodal text retrieval.

Method	Audio → Text			Image → Text			Audio & Image → Text		
	MRR ↑	R@1 (%) ↑	MR ↓	MRR ↑	R@1 (%) ↑	MR ↓	MRR ↑	R@1 (%) ↑	MR ↓
PCME	–	–	–	0.067 ± 0.003	2.73 ± 0.27	131	–	–	–
MPC	–	–	–	0.025 ± 0.002	0.57 ± 0.13	239	–	–	–
Baseline	0.062 ± 0.002	1.93 ± 0.27	82	0.126 ± 0.006	5.99 ± 0.59	47	0.146 ± 0.007	6.96 ± 0.76	30
Proposed	0.113 ± 0.004	4.99 ± 0.37	46	0.541 ± 0.007	44.21 ± 0.99	2	0.58 ± 0.009	47.75 ± 1.19	2

Table 6. Comparison on AS5M dataset for multimodal audio retrieval.

Method	Image → Audio			Text → Audio			Image & Text → Audio		
	MRR ↑	R@1 (%) ↑	MR ↓	MRR ↑	R@1 (%) ↑	MR ↓	MRR ↑	R@1 (%) ↑	MR ↓
Baseline	0.045 ± 0.002	1.32 ± 0.2	138	0.067 ± 0.003	2.11 ± 0.24	77	0.069 ± 0.003	2.43 ± 0.32	74
Proposed	0.072 ± 0.004	2.62 ± 0.33	92	0.115 ± 0.005	4.86 ± 0.47	44	0.126 ± 0.006	5.54 ± 0.62	37

space using t-SNE [51]. Figure 3 shows that their learned representations form clusters regarding music genres. That is, audio, images, and text in each of these genres are closely associated with each other.

5. CONCLUSION

We proposed a method for multimodal MIR that leverages the probabilistic representations of content items. Our contributions can be summarized as follows. First, we leveraged the von Mises-Fisher (vMF) distribution, which has been used for single-modal tasks [14–16] but has not been used for multimodal retrieval tasks. In addition, the recently-invented spherical sliced-Wasserstein (SSW) [22] p -distance for optimal transport is surprisingly computationally efficient and useful, but has not yet been used in the MIR community. Moreover, we designed the two novel loss functions, \mathcal{L}_C and \mathcal{L}_S , using both probabilistic contrastive

learning and optimal transport to facilitate probabilistic multimodal representation learning. To our knowledge, this is the first work to utilize these reusable insights for probabilistic representation learning. Second, we confirmed the effectiveness of integrating the contrastive loss function \mathcal{L}_C with the loss function \mathcal{L}_S based on the optimal transport distance through quantitative evaluations, and showed that the proposed method can retrieve more appropriate content items for single-modal and multimodal queries. Third, we conducted a qualitative analysis, showing that music audio, images, and text for the same music style are located close to each other on S_{shared}^{d-1} . These results demonstrated that the proposed method is effective for multimodal MIR.

The underlying principles of the proposed method can work for any retrieval tasks regardless of modalities, which will lead to a broader scope of application. As such, we believe that the proposed method will shed light on other challenging retrieval tasks and usher in practical solutions.

6. ACKNOWLEDGMENTS

This work was supported in part by JST CREST Grant Number JPMJCR20D4 and JSPS KAKENHI Grant Number 22K18017, Japan.

7. REFERENCES

- [1] D. Surís, C. Vondrick, B. Russell, and J. Salamon, “It’s time for artistic correspondence in music and video,” in *Proceedings of the IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2022, pp. 10 564–10 574.
- [2] Q. Huang, A. Jansen, J. Lee, R. Ganti, J. Y. Li, and D. P. Ellis, “MuLan: A joint embedding of music audio and natural language,” in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2022, pp. 559–566.
- [3] H. Li, J. Song, L. Gao, P. Zeng, H. Zhang, and G. Li, “A differentiable semantic metric approximation in probabilistic embedding for cross-modal retrieval,” in *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, vol. 35, 2022, pp. 11 934–11 946.
- [4] S. Chun, S. J. Oh, R. S. De Rezende, Y. Kalantidis, and D. Larlus, “Probabilistic embeddings for cross-modal retrieval,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 8415–8424.
- [5] A. Neculai, Y. Chen, and Z. Akata, “Probabilistic compositional embeddings for multimodal image retrieval,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 4547–4557.
- [6] T. Jebara, R. Kondor, and A. Howard, “Probability product kernels,” *J. Mach. Learn. Res.*, vol. 5, pp. 819–844, 2004.
- [7] L. Vilnis and A. McCallum, “Word representations via gaussian embedding,” in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014, pp. 1–12.
- [8] Y. Shi and A. K. Jain, “Probabilistic face embeddings,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 6902–6911.
- [9] M. Kirchhof, K. Roth, Z. Akata, and E. Kasneci, “A non-isotropic probabilistic take on proxy-based deep metric learning,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022, pp. 435–454.
- [10] S. J. Oh, K. Murphy, J. Pan, J. Roth, F. Schroff, and A. Gallagher, “Modeling uncertainty with hedged instance embedding,” *arXiv preprint arXiv:1810.00319*, 2018.
- [11] J. Park, J. Lee, I.-J. Kim, and K. Sohn, “Probabilistic representations for video contrastive learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 14 711–14 721.
- [12] J. Chang, Z. Lan, C. Cheng, and Y. Wei, “Data uncertainty learning in face recognition,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 5710–5719.
- [13] B. D. Roads and B. C. Love, “Enriching imagenet with human similarity judgments and psychological embeddings,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 3547–3557.
- [14] S. Li, J. Xu, X. Xu, P. Shen, S. Li, and B. Hooi, “Spherical confidence learning for face recognition,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 15 629–15 637.
- [15] T. R. Scott, A. C. Gallagher, and M. C. Mozer, “von mises-fisher loss: An exploration of embedding geometries for supervised learning,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 10 612–10 622.
- [16] M. Kirchhof, E. Kasneci, and S. J. Oh, “Probabilistic contrastive learning recovers the correct aleatoric uncertainty of ambiguous inputs,” in *Proceedings of the International Conference on Machine Learning (ICML)*, 2023, pp. 17 085–17 104.
- [17] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” in *Proceeding of the International Conference on Learning Representations (ICLR)*, 2014.
- [18] T. R. Davidson, L. Falorsi, N. De Cao, T. Kipf, and J. M. Tomczak, “Hyperspherical variational auto-encoders,” in *Proceeding of the Conference on Uncertainty in Artificial Intelligence (UAI)*, 2018, pp. 856–865.
- [19] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, “Learning transferable visual models from natural language supervision,” in *Proceedings of the International Conference on Machine Learning (ICML)*, 2021, pp. 8748–8763.
- [20] Y. Wu, K. Chen, T. Zhang, Y. Hui, T. Berg-Kirkpatrick, and S. Dubnov, “Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.
- [21] C. Villani, *Topics in optimal transportation*. American Mathematical Soc., 2021, vol. 58.

- [22] C. Bonet, P. Berg, N. Courty, F. Septier, L. Drumetz, and M.-T. Pham, “Spherical sliced-wasserstein,” in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023.
- [23] P. H. Le-Khac, G. Healy, and A. F. Smeaton, “Contrastive representation learning: A framework and review,” *IEEE Access*, vol. 8, pp. 193 907–193 934, 2020.
- [24] K. Sohn, “Improved deep metric learning with multi-class N-pair loss objective,” in *Proceedings of the 30th International Conference on Neural Information Processing Systems (NIPS)*, vol. 29, 2016, pp. 1857–1865.
- [25] A. v. d. Oord, Y. Li, and O. Vinyals, “Representation learning with contrastive predictive coding,” *arXiv preprint arXiv:1807.03748*, 2018.
- [26] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, “Momentum contrast for unsupervised visual representation learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 9729–9738.
- [27] C. Frogner, C. Zhang, H. Mobahi, M. Araya, and T. A. Poggio, “Learning with a wasserstein loss,” in *Proceeding of the Advances in Neural Information Processing Systems (NIPS)*, vol. 28, 2015.
- [28] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein generative adversarial networks,” in *Proceedings of the International Conference on Machine Learning (ICML)*, 2017, pp. 214–223.
- [29] D. Garg, Y. Wang, B. Hariharan, M. Campbell, K. Q. Weinberger, and W.-L. Chao, “Wasserstein distances for stereo disparity estimation,” in *Proceeding of the Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020, pp. 22 517–22 529.
- [30] S. Kolouri, K. Nadjahi, U. Simsekli, R. Badeau, and G. Rohde, “Generalized sliced wasserstein distances,” in *Proceeding of the Advances in Neural Information Processing Systems (NeurIPS)*, vol. 32, 2019.
- [31] T. Bendokat, R. Zimmermann, and P.-A. Absil, “A grassmann manifold handbook: Basic geometry and computational aspects,” *arXiv preprint arXiv:2011.13699*, 2020.
- [32] J. Delon, J. Salomon, and A. Sobolevski, “Fast transport optimization for monge costs on the circle,” *SIAM J. Appl. Math.*, vol. 70, no. 7, pp. 2239–2258, 2010.
- [33] J. Rabin, J. Delon, and Y. Gousseau, “Transportation distances on the circle,” *J. Math. Imaging Vis.*, vol. 41, no. 1, pp. 147–167, 2011.
- [34] S. Hundrieser, M. Klatt, and A. Munk, *The Statistics of Circular Optimal Transport*. Springer Nature Singapore, 2022, pp. 57–82.
- [35] M. Fréchet, “Les éléments aléatoires de nature quelconque dans un espace distancié,” *Annales de l’institut Henri Poincaré*, vol. 10, no. 4, pp. 215–310, 1948.
- [36] H. Karcher, “Riemannian center of mass and mollifier smoothing,” *Commun. Pure Appl. Math.*, vol. 30, no. 5, pp. 509–541, 1977.
- [37] L. Prétet, G. Richard, and G. Peeters, “Cross-modal music-video recommendation: A study of design choices,” in *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, 2021, pp. 1–9.
- [38] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan, “YouTube-8M: A large-scale video classification benchmark,” *arXiv preprint arXiv:1609.08675*, 2016.
- [39] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush, “Transformers: State-of-the-art natural language processing,” in *Proceedings of the Empirical Methods in Natural Language Processing: System Demonstrations (EMNLP-SD)*, 2020, pp. 38–45.
- [40] Y. Luo and N. Mesgarani, “Conv-TasNet: Surpassing ideal time–frequency magnitude masking for speech separation,” *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 27, no. 8, pp. 1256–1266, 2019.
- [41] R. Takahashi, T. Matsubara, and K. Uehara, “Data augmentation using random image cropping and patching for deep CNNs,” *IEEE Trans. Circuits. Syst. Video Technol.*, vol. 30, no. 9, pp. 2917–2931, 2019.
- [42] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, “Random erasing data augmentation,” in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2020, pp. 13 001–13 008.
- [43] T. Sellam, D. Das, and A. P. Parikh, “BLEURT: Learning robust metrics for text generation,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020, pp. 7881–7892.
- [44] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, “PyTorch: An imperative style, high-performance deep learning library,” in *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, vol. 32, 2019, pp. 8024–8035.
- [45] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.
- [46] N. Craswell, “Mean reciprocal rank,” in *Encyclopedia of Database Systems*. Springer US, 2009.

- [47] M. Cherti, R. Beaumont, R. Wightman, M. Wortsman, G. Ilharco, C. Gordon, C. Schuhmann, L. Schmidt, and J. Jitsev, “Reproducible scaling laws for contrastive language-image learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 2818–2829.
- [48] J. Droppo and O. Elibol, “Scaling laws for acoustic models,” in *Proceedings of Interspeech 2021*, 2021, pp. 2576–2580.
- [49] D. Hernandez, J. Kaplan, T. Henighan, and S. McCandlish, “Scaling laws for transfer,” *arXiv preprint arXiv:2102.01293*, 2021.
- [50] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei, “Scaling laws for neural language models,” *arXiv preprint arXiv:2001.08361*, 2020.
- [51] L. Van der Maaten and G. Hinton, “Visualizing data using t-SNE,” *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 2579–2605, 2008.

TOWARDS AUTOMATED PERSONAL VALUE ESTIMATION IN SONG LYRICS

Andrew M. Demetriou¹ Jaehun Kim² Sandy Manolios¹ Cynthia C. S. Liem¹

¹ Delft University of Technology

² SiriusXM/Pandora

a.m.demetriou@tudelft.nl / c.c.s.liem@tudelft.nl

ABSTRACT

Most music widely consumed in Western Countries contains song lyrics, with U.S. samples reporting almost all of their song libraries contain lyrics. In parallel, social science theory suggests that personal values - the abstract goals that guide our decisions and behaviors - play an important role in communication: we share what is important to us to coordinate efforts, solve problems and meet challenges. Thus, the values communicated in song lyrics may be similar or different to those of the listener, and by extension affect the listener's reaction to the song. This suggests that working towards automated estimation of values in lyrics may assist in downstream MIR tasks, in particular, personalization. However, as highly subjective text, song lyrics present a challenge in terms of sampling songs to be annotated, annotation methods, and in choosing a method for aggregation. In this project, we take a perspectivist approach, guided by social science theory, to gathering annotations, estimating their quality, and aggregating them. We then compare aggregated ratings to estimates based on pre-trained sentence/word embedding models by employing a validated value dictionary. We discuss conceptually 'fuzzy' solutions to sampling and annotation challenges, promising initial results in annotation quality and in automated estimations, and future directions.

1. INTRODUCTION

Popular music in Western countries almost always contains lyrics, making song lyrics a widely, repeatedly consumed [1] form of text. Over 616 million people subscribe to streaming services worldwide¹, many of whom stream more than an hour of music every day². Lyrics have been shown to be a salient component of music [2], and out

¹ <https://www.musicbusinessworldwide.com/files/2022/12/f23d5bc086957241e6177f054507e67b.png>

² <https://www.gwi.com/reports/music-streaming-around-the-world>

© A. M. Demetriou, J. Kim, S. Manolios, and C. C. S. Liem. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** A. M. Demetriou, J. Kim, S. Manolios, and C. C. S. Liem, "Towards Automated Personal Value Estimation in Song Lyrics", in *Proc. of the 25th Int. Society for Music Information Retrieval Conf.*, San Francisco, United States, 2024.

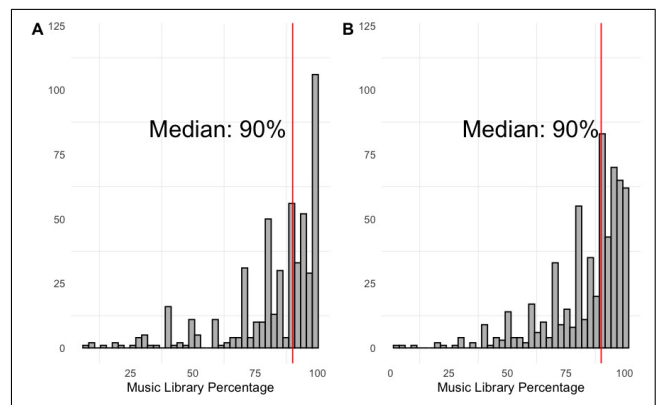


Figure 1. Distribution of self-reported percentage of music library containing lyrics from two representative US samples, $n=505$ and $n=600$ respectively.

of over 1400 number-1 singles in the UK charts, only 30 were instrumental³. The two representative US population samples that were our annotators indicate a median 90% of songs in their libraries contain lyrics (Figure 1).

It is thus not surprising that informative relationships between popular songs and their lyrical content have been shown: e.g., country music lyrics rarely include political concepts [3], and songs with more typical [4] and more negative [5] lyrics appear to be more successful. [6] showed that patients are more likely to choose music with lyrics when participating in music-based pain reduction interventions, although melody had an overall larger effect [7] showed that lyrics enhance self reported emotional responses to music, and [8] showed a number of additional brain regions were active during the listening of sad music with lyrics, vs. sad music without lyrics. In fields closer to MIR, [9] show that estimating psychological concepts from lyrics showed a small benefit in a number of MIR tasks, and [10] showed a correlation between moral principles estimated from song lyrics and music preferences.

A connection between music lyrics and music preferences anticipated by theory involves the personal values perceived in the lyrics by listeners. Prior work has shown correlations between an individual's values, and the music they listen to [10–13], suggesting that we seek music in

³ https://en.wikipedia.org/wiki/List_of_instrumental_number_ones_on_the_UK_Singles_Chart

line with our principles. Yet we have not seen an attempt to measure perceived personal values expressed in the lyrics themselves via human annotation or automated methods.

In this work we take a first step towards the automated estimating the values perceived in song lyrics. As artistic and expressive language, lyrics are ambiguous text: they contain different forms of analogy and wordplay [14]. Thus we take a perspectivist approach to the annotations: because we expect that perceptions will vary substantially more than in other annotation tasks, we aim to represent the general perceptions of only one population. We account for the subjectivity by gathering a large number of ratings (median 27) per song from a targeted population sample (U.S.), of 360 carefully sampled song lyrics, using a psychometric questionnaire that we adjust for this purpose. We treat values in line with theory: as ranked lists, using Robust Ranking Aggregation (RRA) to arrive at our 'ground truth'. We then gather estimates from word embedding models, by measuring semantic similarity between the lyrics and a validated dictionary. We show that ranked lists from estimates correlate moderately with annotation aggregates. We then discuss the implications of our results, the limitations of this project, and anticipated future work.

2. PERSONAL VALUES

The modern study of human values spans over 500 samples in nearly 100 countries over the past 30 years, and has shown a relatively stable structure [15], as illustrated in Figure 2. Personal values are a component of personality, defined as the hierarchy of principles that guide a person's thoughts, behaviors, and the way they evaluate events [16, 17]. Basic human values can be used to describe people or groups: social science theory suggests that each person uses a hierarchical list of values as life-guiding principles [18], such that we prioritize some values over others as we make decisions. Schwartz's theory is the most widely used in social and cultural psychology, and has shown correlations with important behaviors, ranging from political affiliation to personal preferences [15].

We communicate our values in order to gain cooperation and coordinate our efforts, according to Schwartz [19]. Thus our values will manifest in the words that we use [20]. Although personal values are traditionally measured by having individual people complete validated psychological questionnaires, it has been argued that values may be clearly expressed in the speech and text that we produce [20].

A common approach to measuring psychological aspects in text is to validate dictionaries: curated sets of words, with subsets aimed at measuring each component of the psychological aspect in question [22–25]. Some work estimating the values of the authors of text has been conducted on individuals who have written personal essays and social media posts e.g. [25, 26], and in arguments abstracted from various forms of public facing text [27]. However, we have not seen work aimed at measuring values *perceived* in text, measuring them along a scale as in

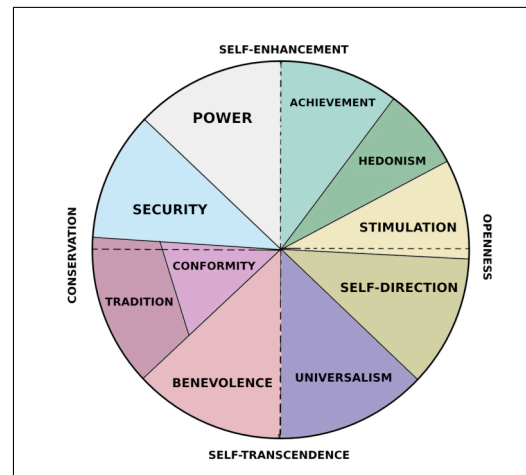


Figure 2. Visualization of the Schwartz 10-value inventory from [19] used in this paper, such that more abstract values of Conservation, vs. Openness to Change, and Self-transcendence vs. Self-enhancement form 4 higher-order abstract values. Illustration adapted from [21].

prior work [19], or ultimately treating them as a hierarchical list in line with theory [18].

3. PRIMARY LYRICS DATA

We aim to collect a sample of lyric data where the lyrics are as accurate as possible, and our sample is as representative as possible. We sampled from the population of songs in the Million Playlist Dataset (MPD)⁴ as it is large and recent compared to other similar datasets. The lyrics themselves were obtained through the API of Musixmatch⁵, a lyrics and music language platform. Musixmatch lyrics are crowdsourced by users who add, correct, sync, and translate them. Musixmatch then engages in several steps to verify quality of content, including spam detection, formatting, spelling and translation checking, as well as manual verification by over 2000 community curators, and a local team of Musixmatch editors. Via their API, Musixmatch provided us with an estimated first 30% of the lyrics of each song.

Using the 'fuzzy' stratified sampling method described below, we sampled 2200 songs. Three members of the research team manually screened approximately 600 of the 2200 songs for inclusion. Each set of lyrics was confirmed to be a match to the actual song, and for suitability⁶. Lyrics were unsuitable if they were: 1) not English, 2) completely onomatopoeic, 3) repetitions of single words or phrases, 4) too few words to estimate values present or, 5) were not a match to the meta-data of the song, e.g. artist title, song name. This resulted in 380 songs, 20 of which were used in a pilot study to determine the number of ratings to gather per song, and 360 were used for annotation.

⁴ <https://research.atspotify.com/2020/09/the-million-playlist-dataset-remastered/>

⁵ <https://www.musixmatch.com/>

⁶ Each member independently screened each lyric and the screening process overall was discussed at length.

3.1 Fuzzy Stratified Sampling

An initial challenge is determining how to represent a corpus. In our case, the population of songs is known to be very large⁷. An ideal scenario would be one in which we aim for a known number of songs, randomly sampled from within clearly defined strata, i.e. relevant subgroups, also known as *stratified random sampling* [28]. However, for music, we do not know how many songs we would need to sample in order to reach saturation, what the relevant strata to randomly sample within should be, and how to measure relevant parameters from each stratum.

Some measurable strata that affect the use of language in song lyrics are clear: e.g., the year of release, which may reflect different events or time-specific colloquial slang. Others are less clear: e.g., there is no single metric of popularity for music, although it can be estimated from various sources such as hit charts. Some may be very subjective, such as genre, for which there may be some overlap of human labelling, but no clear taxonomy exists in the eyes of musicological domain experts [29].

Based upon these considerations, we aimed for a stratified random sampling procedure, based on strata that we acknowledge to be justifiable given our purpose, yet in some cases conceptually ‘fuzzy’: (1) release date; (2) popularity, operationalized as artist playlist frequency from the MPD [30]; (3) genre, estimated from topic modeling on Million Song Dataset artist tags [31]; (4) lyric topic, through a bag-of-words representation of the lyrics data. Popularity and Release date were divided into equally spaced bins; e.g. we divided release year into decades (60s, 70s, 80s, and so on), and genre and lyric topic were divided into categories.

Release date was quantized into 14 bins in 10-year increments from 1890-2030. Popularity was exponentially distributed, and thus manually binned, to make the quantiles per each of the 7 bins as similar as possible. Thus, the first bin contained the lowest 40% of the population in terms of popularity, while the 7th bin contained the highest 9%. Topic modelling was applied on a bag-of-words representation of the lyrics data and artist-tag data to yield 25 estimated genres and 9 lyrics topic strata, respectively.

We observed a skewness of data concentration with regard to several of our strata, e.g., songs that are recent and widely popular are most likely to be drawn. To correct for this and thus get a more representative sample of an overall song catalogue, we oversample from less populated bins. For this, we use the maximum-a-posteriori (MAP) estimate of the categorical distribution of each stratum. The oversampling is controlled by concentration parameter a of the symmetric Dirichlet distribution. We heuristically set this parameter such that songs in underpopulated bins still will make up 5-10% of our overall pool⁸. Through this method, we subsampled our initial 2200 songs lyrics.

⁷ e.g., Spotify reports over 100 million songs in its catalogue <https://newsroom.spotify.com/company-info/>

⁸ Full code of our sampling procedure is at https://anonymous.4open.science/r/Lyrics-value-estimators-CE33/1_stimulus_sampling/stratified_sampling.py

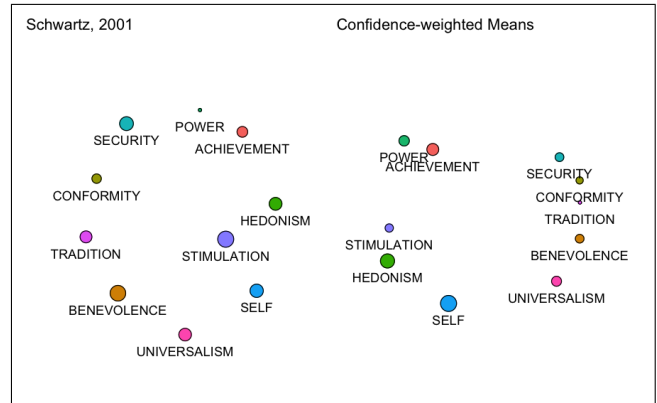


Figure 3. MDS plots derived from the correlation plot reported in [32], and our participant responses as confidence-weighted means

4. GROUND-TRUTHING PROCEDURE

We chose to obtain our annotations from samples of the US population, representative in terms of self-reported sex, ethnicity and age, through the Prolific⁹ platform. Annotator pools comprised of two samples, the first $n=505$ wave participated in a pilot study to estimate the number of ratings per song needed on average, and the second $n=600$ wave comprised our main data collection. Participants completed the survey on the Qualtrics¹⁰ platform.

We clearly differentiate between the Author and the Speaker of lyrics by explaining to participants that the Author of song lyrics may write from the perspective of someone or something else (the Speaker). 17 randomly selected sets of lyrics were then shown to each participant along with instructions to annotate each with the values of the Speaker. We adapted the 10-item questionnaire used in [33] for the value annotations, as it is the shortest questionnaire for assessing personal values whose validity and reliability have been assessed¹¹. As in [33], each questionnaire item is a specific value along with additional descriptive words e.g. POWER (social power, authority, wealth). We adjusted it by asking participants to indicate the values of the Speaker of the lyrics, and by having them indicate on a bar with -100 (opposed to their principles) on one end, and +100 (of supreme importance) on the other end instead of a likert scale. In addition, we asked participants to indicate how confident they were in their ratings, on a scale of 0 (not at all confident) to 100 (extremely confident), inspired by work that has shown that self-reported confidence in ratings can be used to estimate the accuracy of individual ratings [34].

We used a procedure similar to [35] in order to determine the number of raters. Specifically, we recruited a representative 500+ participant sample of the US using the Prolific platform, who completed our survey for 20 songs. We then computed canonical mean ratings of each of the 10

⁹ <https://prolific.co>

¹⁰ <https://qualtrics.com>

¹¹ It has shown correlations ranging from .45-.70 per value with longer more established procedures, test-retest reliability, as well as the typical values structure shown in Figure 2

values per song, and inter-rater reliability using Cronbach’s Alpha. We then estimated Cronbach’s alpha for a range of subsample sizes (5 to 50 participants in increments of 5), for each of the 10 values. We repeated this procedure 10 times per increment, separately for each of the 10 values, and examined the distribution of Cronbach’s Alpha. We specifically looked for the sample size with which Alpha exceeded .7¹². We arrived at a conservative estimate of 25 ratings per set of lyrics, with songs receiving a median 27 ratings (range 22-30).

4.1 Reliability, Agreement and Initial Validation

The rater reliability was estimated via intra-class correlation for each personal value, (type 2k: see [36]) using the ‘psych’ package in R [37], all of which exceeded .9 (excellent reliability). As an initial validation, we compare data simulated from values in the upper triangle of a correlation matrix reported in [32] to those derived from our study. To aggregate our participants rankings for this purpose, we compute confidence-weighted means inspired by [34]: we estimate confidence-weights by dividing participant’s self-reported confidence of a given rating by the highest possible response (100), and then compute aggregated means weighted by these. For both the simulated data and confidence-weighted mean scores, we generate a multi-dimensional scaling plot (MDS) [38] for visual comparison, which has previously been used as method to assess measurements conform to theory [25, 33]. Note: the interpretation is to observe whether each of the values appears next to expected neighboring values, and not each value’s orientation. From these plots (Figure 3), in as little as our 360 annotated lyrics, we surprisingly see similar clusters and relative positioning relations emerging as those obtained from a formal cross-cultural study.

We coerced the annotated scores to ranked lists of values, such that the highest scoring value was at the top. We derived ranked lists per participant per song, and then used Robust Ranking Aggregation (RRA) to extract a single ranked list per song. Aggregation was conducted using R version 4.2.2. [39], and the `RobustRankAggreg` package [40]. Briefly, RRA produces a ranked list by comparing the probability of the observed ranking of items to rankings from a uniform distribution. Essentially, scores are determined by comparing the height of an item on a set of lists to where it would appear if its rank were randomly distributed across lists. These scores are then subjected to statistical tests, where the resulting p value is Bonferroni corrected by the number of input lists [41]. Thus, when an item appears in different positions on a list, the resulting p value is high, as its position appears randomly distributed.

As lyrics are ambiguous, we expect that some songs’ values are completely subjective. We operationalize these as randomly distributed rankings for all personal values for completely subjective songs, i.e. p values above .05 for all 10 items on the ranked list. Results from the RRA show 62 songs with p values above .05 for all 10 values, and 96

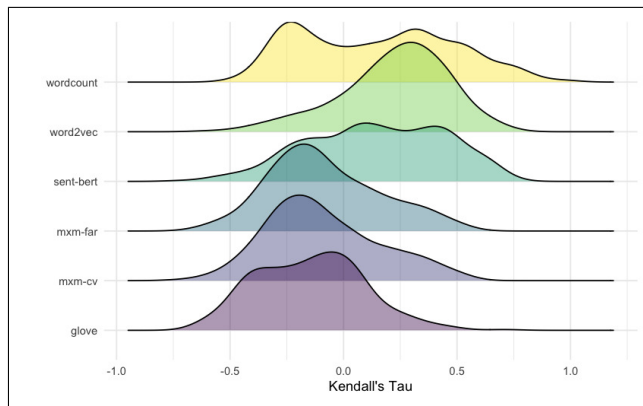


Figure 4. Rank correlations between NLP systems / word counts and Robust Ranking Aggregation lists, by normalization scheme.

songs with only 1 value ranked. At most, 5 values were ranked, which occurred for 35 songs. Thus, we confirm that although there was correspondence in the scores that participants assigned per value per song, ranked lists did not always agree.

5. AUTOMATED SCORING

For automated scoring, we use a dictionary of words associated with the 10 Schwartz values [25]. With this dictionary as reference, we computationally estimate the degree to which each value is reflected in the lyrics text according to traditional word counting [25], as well as by assessing cosine similarity between dictionary words and lyrics texts using four classes of pre-trained word embeddings: `word2vec`, a generic English word embedding trained on Google News dataset [42]; `glove`, another generic English word embedding trained on Common Crawl dataset [43]; `mxm-far-[1~10]`, trained on the collected initial lyrics candidate pool, employing the Glove model [43] (using ten models populated from ten cross-validation folds, whose parameters are tuned based on English word similarity judgement data [44].); `mxm-cv-[1~10]`, ten variants of lyrics based word-embeddings from cross-validation folds selected by Glove loss values on the validation set; and finally, `sent-bert`, a transformer model that encodes sentence into a embedding vector, fine-tuning of a generic self-supervised language model called MPNet, which is trained on a large scale English corpus [45]. Our process thus resulted in 24 sets of scores: 5 from models and one from word-counting, normalized using four methods.

We take the perspective from theory that that value assessments should be seen as ranked lists, and thus coerce scores to ranked lists per model per song. We then compute rank correlations between ranked lists derived from model scores and RRA lists from participants. As RRA lists assess lack of consensus on rankings, personal values with high p values received tied rankings, at the bottom of the list. Correlations were computed using Kendall’s τ which is robust to ties (Figure 4).

¹² .7 is a commonly considered an acceptable level of reliability in the form of internal consistency

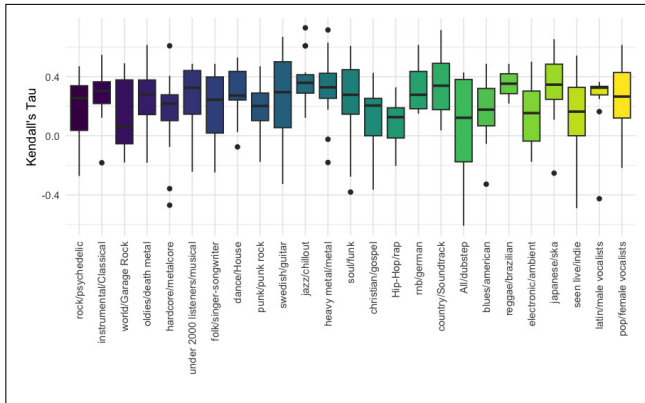


Figure 5. Rank correlations between word2vec scores Robust Ranking Aggregation lists, per genre grouping operationalized as Artist Tag Topic.

In earlier work [25, 46], Pearson correlations of 0.1-0.2 were considered as moderate evidence of the validity of a proposed dictionary in relation to a psychometrically validated instrument. Although we are using a different metric, we observe several models whose mean rank correlations exceed the .10 mark. The mean Kendall’s τ values were highest for the word2vec, sent-bert, and wordcount models with null normalization ($SD=.24$, $.30$, and $.34$ respectively). We further observe that 76% of the rank correlations for word2vec exceed the .10 mark, followed by 56.1% from sent-bert, and 47.8% from wordcounts. Although none of these models had been thoroughly optimized and thus this cannot be interpreted as a thorough benchmark, we do see evidence of higher than expected correlations.

We also explored whether our fuzzy strata might hint towards more or less automatically scorable lyrics. We found most strata to be uninformative. However, when examining the rank correlations for our overall best performing model, word2vec, we did observe higher mean correlations for some artist tag topics than others (Figure 5). In particular, topics 10 (which included the tags: ‘jazz’, ‘chillout’, ‘lounge’, ‘trip-hop’, ‘downtempo’), 11 (which included the tags like: ‘metal’, ‘celtic’, ‘thrash metal’, ‘dutch’, ‘seen live’), and 16 (which included tags like: ‘country’, ‘Soundtrack’, ‘americana’, ‘danish’, ‘Disney’). Although speculative, we do expect that certain genres are more difficult to interpret than others, in particular for people who are generally unfamiliar with such music.

6. DESCRIPTIVE ANALYSES

We conduct a further exploratory data analysis by examining the gathered value annotations with respect to the song strata introduced in Section 3.1. To better understand the overall patterns of value rankings in songs we visualize the average ranking of each value for each level of each stratum. To reflect the uncertainty of aggregated ranking from RRA, we employ ‘truncated’ rankings: the values within each aggregated ranked list are considered ties if their p-values higher than the threshold ($p = 0.05$), hence with

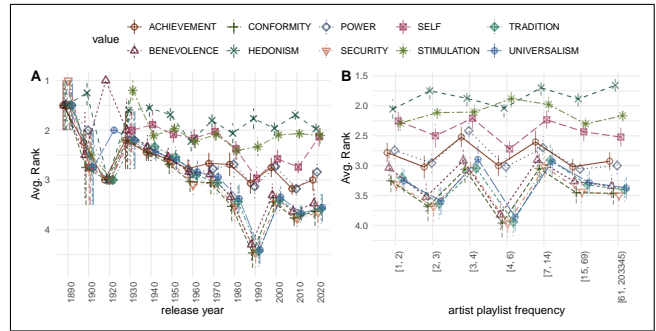


Figure 6. Average value ranking from ‘release year’ (A) and ‘artist-playlist frequency’ (B). x and y axis represent the strata and average ranking measure from RRA, respectively. Each point in different point shapes and vertical bars denote the average ranking value and its confidence interval (at 95% level). For visual convenience, we connected the same values with lines.

high uncertainty in their ranking positions.¹³

In all results, we observe that there is a tendency of overall value ranking: 1) a generally strong presence of HEDONISM in higher ranks in all cases, followed by STIMULATION and SELF (SELF-DIRECTION). 2) ACHIEVEMENT and POWER generally follow next across all figures, and 3) the rest of the values, including BENEVOLENCE, UNIVERSALISM, SECURITY, CONFORMITY, and TRADITION overall rank lower, but show higher variability across strata. We refer to these three groups of values as *Group1* (HEDONISM, STIMULATION and SELF), *Group2* (ACHIEVEMENT and POWER), and *Group3* (the rest) for the rest of the section.

Zooming in each to stratum, in Figure 6, we observe that the ‘release year’ (sub-figure A) strata show the most consistent and visible trend especially for Group3, which generally declines over time. Such a trend is not as obvious in Group1 and only partially observed in Group2. The low presence of Group3 is especially noticeable in the 1990s, although it regained its presence to some degree, a pattern which the SELF value from Group1 partially shares. Such visible movements suggest that the rank of specific values may evolve over time. In sub-figure B, we observe the most flat response across all strata considered: beyond the fluctuation pattern that is shared by all groups, there is no substantial variability among groups, which implies that popularity might not be as correlated as the ‘release year’.

Moving onto Figure 7, we discuss the value presence pattern in two ‘topic’ strata. First, in sub-figure C, we observe that Group3 values show overall higher variability than ‘artist playlist frequency’. It is notable that there are a few distinct topics in which Group3 values show a significant difference; the sixth, seventh and fourteenth topics, which correspond to the ‘under 2000 listeners/musical’, ‘folk/singer-songwriter’, and ‘Hip-Hop/rap’ topics when represented in primary topic terms. Specifically, we see

¹³ We assume that the adjusted exact p-value from RRA monotonically decreases as the rank position ascends (i.e., the lower the p-value is, the higher the ranking position is).

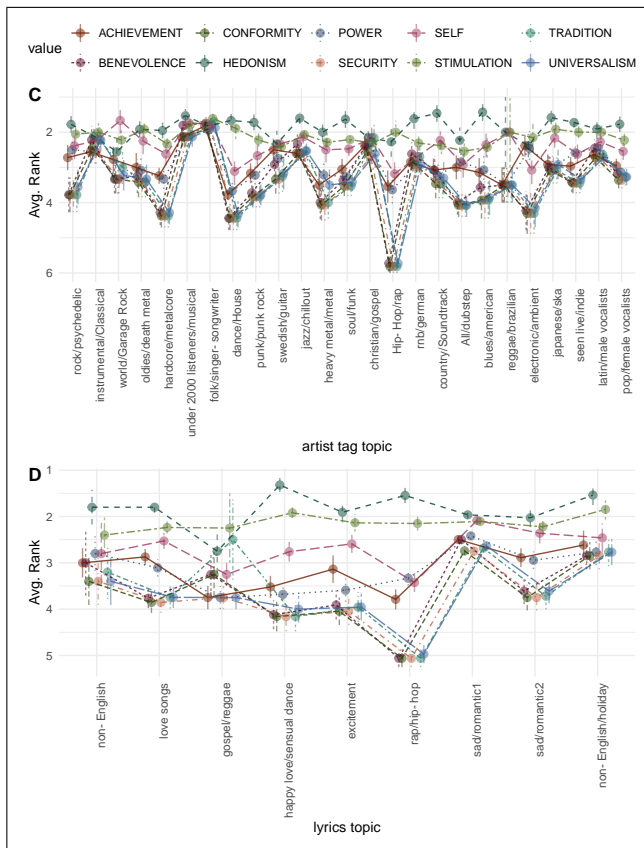


Figure 7. Average value ranking from ‘artist-tag topic’ (C) and ‘lyrics topic’ (D).

that first two topics show a high presence of Group3 values, while the latter topics show the least presence of Group3 values. It suggests that the artists in these styles/genres were perceived on average to present clearly different sets of values through their lyrics, distinguished by the inclusion/exclusion of values such as BENEVOLENCE or UNIVERSALISM.

Finally, considering sub-figure D, we observe a similar pattern as ‘artist playlist frequency’ in 6, albeit with relatively more variability in Group3 values. Notably, the ‘rap/hip-hop’ lyrics topic shows the least presence of Group3 values, which aligns to the observation from previous sub-figure. The ‘sad/romantic1’ topic, on the other hand, shows the highest ranking of Group3 values. Another remarkable topic is ‘gospel/reggae’ topic, where HEDONISM value is least present, which semantically aligns well with the typical lyrical theme of those songs.

7. LIMITATIONS AND FUTURE WORK

In this work we attempt to ground-truth perceptions of ambiguous song lyrics for perceived human values. We adopt a validated questionnaire from the social sciences for this purpose, in addition to a purposeful, if conceptually ‘fuzzy’, stratified sampling strategy, and estimate the average number of ratings needed to estimate the average perception of values in a song. We acknowledge our current sample of 360 lyrics is small and may need expansion

for more typical work, and that, while we had a representative population sample, not every member of the sample rated every song. We thus did gather diverse opinions, but cannot claim they fully represent the target population. In addition, the small sample of songs allowed for only limited observation of patterns that might emerge in larger samples with relation to our defined strata, and indefinite conclusions given the overall massive population of songs in existence. We also did not assess whether variations on the annotation instrument might result in substantial differences in the annotations we received [47], nor did we repeat our procedure [48]. In addition, we acknowledge that participants from different groups will perceive and thus annotate corpora differently [49, 50]. Thus, we expect that lyrics may be especially sensitive to varying perceptions, which we did not explore in this work. Finally, we only provide a preliminary comparison to automated scoring methods, and did not leverage the most contemporary tools for this purpose (e.g. Large Language Models). All of these are rich and promising avenues for future work.

The most interesting avenues are potential relationships that could be revealed with more annotated songs, and eventual automated scoring methods. In particular, we see potential in understanding music consumption more broadly from patterns revealed in the dominant value hierarchies in specific music genres, popularity segments, lyrical topics, and even release year. And for understanding music consumption more narrowly, from patterns revealed in an individual’s music preferences, and the degree to which they conform with their own value hierarchy.

8. CONCLUSION

Song lyrics remain a widely and repeatedly consumed, yet ambiguous form of text, and thus a promising and challenging avenue for research into better understanding the people that consume them. We observe promising initial results for the annotation of personal values in songs, despite our limitations. MDS plots of aggregated ratings showed the beginnings of the expected structure of values, conforming more closely than might be expected from as little as 360 songs. We also observed high inter-rater reliability in the raw scores, suggesting a sufficiently reliable annotation procedure with 25 ratings. Thus, we see promise on our method for ground-truthing lyrics despite their ambiguity. A post-hoc procedure revealed that 15 ratings may be enough on average: we repeatedly subsampled 5, 10, 15 and 20 ratings for each value within each song, and calculated pearson correlations between subsample means and canonical means. From this, we see Pearson correlations to the canonical mean exceed 0.9 for all values from 15 subsampled ratings. Further lyric annotation may thus require fewer annotations per song than what was gathered in this work. In addition, we observe promising rank correlations between ranked rater scores and our automated methods, with over 75% of the rankings in our best performing model above a minimal threshold of .10. Despite inherent challenges in the task, our method shows initial promise, and multiple fruitful avenues for future work.

9. ETHICS STATEMENT

Our study includes data gathered from people, and was approved by the Human Research Ethics board of our university. We follow Prolific’s guidelines on fair compensation to set our compensation rates. Survey design and data handling were pre-discussed with our institutional data management and research ethics advisors, we obtained formal data management plan and human research ethics approval. Participants gave informed consent before proceeding with the survey, which informed them of the intentions of use for their data, and that it could be withdrawn at any time.

10. REFERENCES

- [1] F. Conrad, J. Corey, S. Goldstein, J. Ostrow, and M. Sadowsky, “Extreme re-listening: Songs people love... and continue to love,” *Psychology of Music*, vol. 47, no. 2, pp. 158–172, 2019.
- [2] A. Demetriou, A. Jansson, A. Kumar, and R. M. Bitner, “Vocals in music matter: the relevance of vocals in the minds of listeners.” in *ISMIR*, 2018, pp. 514–520.
- [3] R. W. Van Sickel, “A world without citizenship: On (the absence of) politics and ideology in country music lyrics, 1960–2000,” *Popular music and society*, vol. 28, no. 3, pp. 313–331, 2005.
- [4] A. C. North, A. E. Krause, and D. Ritchie, “The relationship between pop music and lyrics: A computerized content analysis of the united kingdom’s weekly top five singles, 1999–2013,” *Psychology of Music*, p. 0305735619896409, 2020.
- [5] C. O. Brand, A. Acerbi, and A. Mesoudi, “Cultural evolution of emotional expression in 50 years of song lyrics,” *Evolutionary Human Sciences*, vol. 1, 2019.
- [6] C. Howlin and B. Rooney, “Patients choose music with high energy, danceability, and lyrics in analgesic music listening interventions,” *Psychology of Music*, p. 0305735620907155, 2020.
- [7] S. O. Ali and Z. F. Peynircioğlu, “Songs and emotions: are lyrics and melodies equal partners?” *Psychology of music*, vol. 34, no. 4, pp. 511–534, 2006.
- [8] E. Brattico, V. Alluri, B. Bogert, T. Jacobsen, N. Vartiainen, S. K. Nieminen, and M. Tervaniemi, “A functional mri study of happy and sad emotions in music with and without lyrics,” *Frontiers in psychology*, vol. 2, p. 308, 2011.
- [9] J. Kim, A. M. Demetriou, S. Manolios, M. S. Tavella, and C. C. Liem, “Butter lyrics over hominy grit: Comparing audio and psychology-based text features in mir tasks.” in *ISMIR*, 2020, pp. 861–868.
- [10] V. Preniqi, K. Kalimeri, and C. Saitis, ““ more than words”: Linking music preferences and moral values through lyrics,” *arXiv preprint arXiv:2209.01169*, 2022.
- [11] S. Manolios, A. Hanjalic, and C. C. Liem, “The influence of personal values on music taste: towards value-based music recommendations,” in *Proceedings of the 13th ACM Conference on Recommender Systems*, 2019, pp. 501–505.
- [12] A. Gardikiotis and A. Baltzis, ““rock music for myself and justice to the world!”: Musical identity, values, and music preferences,” *Psychology of Music*, vol. 40, no. 2, pp. 143–163, 2012.
- [13] V. Swami, F. Malpass, D. Havard, K. Benford, A. Costescu, A. Sofitiki, and D. Taylor, “Metalheads: The influence of personality and individual differences on preference for heavy metal.” *Psychology of Aesthetics, Creativity, and the Arts*, vol. 7, no. 4, p. 377, 2013.
- [14] M. Sandri, E. Leonardelli, S. Tonelli, and E. Ježek, “Why don’t you do it right? analysing annotators’ disagreement in subjective tasks,” in *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, 2023, pp. 2420–2433.
- [15] L. Sagiv and S. H. Schwartz, “Personal values across cultures,” *Annual review of psychology*, vol. 73, pp. 517–546, 2022.
- [16] S. H. Schwartz and W. Bilsky, “Toward a universal psychological structure of human values.” *Journal of personality and social psychology*, vol. 53, no. 3, p. 550, 1987.
- [17] S. H. Schwartz, “An overview of the schwartz theory of basic values,” *Online readings in Psychology and Culture*, vol. 2, no. 1, p. 11, 2012.
- [18] M. Rokeach, *The nature of human values*. Free press, 1973.
- [19] S. H. Schwartz, “Universals in the content and structure of values: Theoretical advances and empirical tests in 20 countries,” in *Advances in experimental social psychology*. Elsevier, 1992, vol. 25, pp. 1–65.
- [20] R. L. Boyd and J. W. Pennebaker, “Language-based personality: A new approach to personality in a digital world,” *Current opinion in behavioral sciences*, vol. 18, pp. 63–68, 2017.
- [21] G. R. Maio, “Mental representations of social values,” in *Advances in experimental social psychology*. Elsevier, 2010, vol. 42, pp. 1–43.
- [22] J. W. Pennebaker, R. L. Boyd, K. Jordan, and K. Blackburn, “The development and psychometric properties of liwc2015,” Tech. Rep., 2015.

- [23] J. Graham, J. Haidt, and B. A. Nosek, “Liberals and conservatives rely on different sets of moral foundations.” *Journal of personality and social psychology*, vol. 96, no. 5, p. 1029, 2009.
- [24] D. Holtrop, J. K. Oostrom, W. R. J. van Breda, A. Koutsoumpis, and R. E. de Vries, “Exploring the application of a text-to-personality technique in job interviews,” *European Journal of Work and Organizational Psychology*, vol. 31, no. 6, pp. 799–816, 2022.
- [25] V. Ponizovskiy, M. Ardag, L. Grigoryan, R. Boyd, H. Dobewall, and P. Holtz, “Development and validation of the personal values dictionary: A theory-driven tool for investigating references to basic human values in text,” *European Journal of Personality*, vol. 34, no. 5, pp. 885–902, 2020.
- [26] T. Maheshwari, A. N. Reganti, S. Gupta, A. Jamatia, U. Kumar, B. Gambäck, and A. Das, “A societal sentiment analysis: Predicting the values and ethics of individuals by analysing social media content,” in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*. Association for Computational Linguistics, 2017, pp. 731–741.
- [27] J. Kiesel, M. Alshomary, N. Handke, X. Cai, H. Wachsmuth, and B. Stein, “Identifying the human values behind arguments,” in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 2022, pp. 4459–4471.
- [28] R. M. Groves, F. J. Fowler Jr, M. P. Couper, J. M. Lepkowski, E. Singer, and R. Tourangeau, *Survey methodology*. John Wiley & Sons, 2009, vol. 561.
- [29] C. C. S. Liem, A. Rauber, T. Lidy, R. Lewis, C. Raphael, J. D. Reiss, T. Crawford, and A. Hanjalic, “Music Information Technology and Professional Stakeholder Audiences: Mind the Adoption Gap,” in *Dagstuhl Follow-Ups*. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2012, vol. 3. [Online]. Available: <http://drops.dagstuhl.de/opus/volltexte/2012/3475/>
- [30] C. Chen, P. Lamere, M. Schedl, and H. Zamani, “Recsys challenge 2018: automatic music playlist continuation,” in *Proceedings of the 12th ACM Conference on Recommender Systems, RecSys 2018, Vancouver, BC, Canada, October 2-7, 2018*, S. Pera, M. D. Ekstrand, X. Amatriain, and J. O’Donovan, Eds. ACM, 2018, pp. 527–528. [Online]. Available: <https://doi.org/10.1145/3240323.3240342>
- [31] A. Schindler, R. Mayer, and A. Rauber, “Facilitating comprehensive benchmarking experiments on the million song dataset.” in *ISMIR*. International Society for Music Information Retrieval, 2012, pp. 469–474.
- [32] S. H. Schwartz, G. Melech, A. Lehmann, S. Burgess, M. Harris, and V. Owens, “Extending the cross-cultural validity of the theory of basic human values with a different method of measurement,” *Journal of cross-cultural psychology*, vol. 32, no. 5, pp. 519–542, 2001.
- [33] M. Lindeman and M. Verkasalo, “Measuring values with the short schwartz’s value survey,” *Journal of personality assessment*, vol. 85, no. 2, pp. 170–178, 2005.
- [34] F. Cabitza, A. Campagner, and L. M. Sconfienza, “As if sand were stone. new concepts and metrics to probe the ground on which to build trustable ai,” *BMC Medical Informatics and Decision Making*, vol. 20, no. 1, pp. 1–21, 2020.
- [35] L. M. DeBruine and B. C. Jones, “Determining the number of raters for reliable mean ratings,” Aug 2018. [Online]. Available: osf.io/x7fus
- [36] T. K. Koo and M. Y. Li, “A guideline of selecting and reporting intraclass correlation coefficients for reliability research,” *Journal of chiropractic medicine*, vol. 15, no. 2, pp. 155–163, 2016.
- [37] William Revelle, *psych: Procedures for Psychological, Psychometric, and Personality Research*, Northwestern University, Evanston, Illinois, 2023, r package version 2.3.9. [Online]. Available: <https://CRAN.R-project.org/package=psych>
- [38] M. L. Davison and S. G. Sireci, “Multidimensional scaling,” in *Handbook of applied multivariate statistics and mathematical modeling*. Elsevier, 2000, pp. 323–352.
- [39] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2022. [Online]. Available: <https://www.R-project.org/>
- [40] R. Kolde, *RobustRankAggreg: Methods for Robust Rank Aggregation*, 2022, r package version 1.2.1. [Online]. Available: <https://CRAN.R-project.org/package=RobustRankAggreg>
- [41] R. Kolde, S. Laur, P. Adler, and J. Vilo, “Robust rank aggregation for gene list integration and meta-analysis,” *Bioinformatics*, vol. 28, no. 4, pp. 573–580, 2012.
- [42] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, C. J. C. Burges, L. Bottou, Z. Ghahramani, and K. Q. Weinberger, Eds., 2013, pp. 3111–3119. [Online]. Available: <https://proceedings.neurips.cc/paper/2013/hash/9aa42b31882ec039965f3c4923ce901b-Abstract.html>

- [43] J. Pennington, R. Socher, and C. D. Manning, “Glove: Global vectors for word representation,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, A. Moschitti, B. Pang, and W. Daelemans, Eds. ACL, 2014, pp. 1532–1543. [Online]. Available: <https://doi.org/10.3115/v1/d14-1162>
- [44] M. Faruqui and C. Dyer, “Community evaluation and exchange of word vectors at wordvectors.org,” in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, System Demonstrations*. The Association for Computer Linguistics, 2014, pp. 19–24. [Online]. Available: <https://doi.org/10.3115/v1/p14-5004>
- [45] N. Reimers and I. Gurevych, “Sentence-bert: Sentence embeddings using siamese bert-networks,” in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, K. Inui, J. Jiang, V. Ng, and X. Wan, Eds. Association for Computational Linguistics, 2019, pp. 3980–3990. [Online]. Available: <https://doi.org/10.18653/v1/D19-1410>
- [46] F. D. Richard, C. F. Bond Jr, and J. J. Stokes-Zoota, “One hundred years of social psychology quantitatively described,” *Review of general psychology*, vol. 7, no. 4, pp. 331–363, 2003.
- [47] C. Kern, S. Eckman, J. Beck, R. Chew, B. Ma, and F. Kreuter, “Annotation sensitivity: Training data collection methods affect model performance,” *arXiv preprint arXiv:2311.14212*, 2023.
- [48] O. Inel, T. Draws, and L. Aroyo, “Collect, measure, repeat: Reliability factors for responsible ai data collection,” in *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, vol. 11, no. 1, 2023, pp. 51–64.
- [49] C. Homan, T. C. Weerasooriya, L. Aroyo, and C. Welty, “Annotator response distributions as a sampling frame,” in *Proceedings of the 1st Workshop on Perspectivist Approaches to NLP@ LREC2022*, 2022, pp. 56–65.
- [50] V. Prabhakaran, C. Homan, L. Aroyo, A. Parrish, A. Taylor, M. Díaz, and D. Wang, “A framework to assess (dis) agreement among diverse rater groups,” *arXiv preprint arXiv:2311.05074*, 2023.

AUDIO CONDITIONING FOR MUSIC GENERATION VIA DISCRETE BOTTLENECK FEATURES

Simon Rouard^{1,2} Yossi Adi^{1,3} Jade Copet¹ Axel Roebel² Alexandre Défossez⁴

¹ FAIR Meta ² IRCAM - Sorbonne Université ³ Hebrew University of Jerusalem ⁴ Kyutai

srouard@meta.com, alex@kyutai.org

ABSTRACT

While most music generation models use textual or parametric conditioning (e.g. tempo, harmony, musical genre), we propose to condition a language model based music generation system with audio input. Our exploration involves two distinct strategies. The first strategy, termed textual inversion, leverages a pre-trained text-to-music model to map audio input to corresponding "pseudowords" in the textual embedding space. For the second model we train a music language model from scratch jointly with a text conditioner and a quantized audio feature extractor. At inference time, we can mix textual and audio conditioning and balance them thanks to a novel double classifier free guidance method. We conduct automatic and human studies that validates our approach. We will release the code and we provide music samples on musicgenstyle.github.io in order to show the quality of our model.

1. INTRODUCTION

In the field of music generation, prior research has predominantly focused on producing brief musical segments [1,2], MIDI generation [3], while generating long and coherent waveforms (around 30 seconds) has only recently been tackled [4–6]. Specifically, most of these recent models have been designed to perform text-to-music generation, providing a fascinating tool for creators. Other types of high-level conditioning have been used in previous work such as tempo, harmony [7]. For lower-level and aligned conditioning, the authors of [5] use melody, while [8] uses chords, piano rolls, or the drum stem. However, music is hard to describe textually and the scarcity of text-music pair datasets makes it challenging to generate music in the style of a specific artist or song, since the artist is probably not represented in the training dataset. Then a common use case would be to generate music in the style of a reference segment. This gives more control to the user since they do not have to find a textual prompt that describes the music they want to generate.

In the computer vision domain, the authors of [9] introduced textual inversion to extract visual concepts that can then be used to generate new images with a text-to-image model. Given a few images (3-5) of a concept or object, one sets them as outputs of a frozen text-to-image model with a randomly initialized learnable text embedding. Backpropagating the generative model loss on the text allows to learn new "pseudowords" in the textual embedding space of the model that match the common concept depicted on the images. One can then compose this learnt pseudoword S^* in a textual prompt to generate an image of the learnt concept (for instance "a painting of S^* in the style of Picasso").

We first adapted this method by using the text-to-music model MusicGen [5], using crops of a song to depict a concept, and optimizing the cross-entropy loss of the music language model. This approach does not need to retrain a model from scratch. However, its inference is very slow since it requires hundreds of optimization steps of the textual prompt, including gradient computation through the language model, before generating music.

To tackle this issue, we present another method where we design a style conditioner module that we jointly train with a text-to-music MusicGen model [5]. This style conditioner takes a few seconds of audio and extracts features out of it. As a result this new model can generate music using two modalities as input: waveforms and textual descriptions. Our conditioning is high level even if it can retain some lower level content such as melodic patterns or rhythm. Designing this style conditioner is challenging as we need to extract enough features to have a meaningful conditioning but not too much, to prevent the generative model to copy and loop the conditioning audio. We thus need to introduce and tune information bottlenecks in our conditioning module. Our contributions are the following:

1) We adapt the textual inversion method of [9] to a pretrained text-to-music MusicGen model. This allows to perform audio conditioning for music generation without training a model from scratch.

2) We present our style conditioner method which is based on a frozen audio feature extractor (Encodec [10], MERT [11] or MusicFM [12]) followed by a transformer encoder [13], Residual Vector Quantizer (RVQ) [14] and temporal downsampling. The number of residual streams used by RVQ is adjustable at inference time which gives the user the ability to change the strength of the style conditioning. To our knowledge, we are the first to explore



this approach for music generation.

3) Since the model is trained with both textual and audio conditioning inputs, we can combine both to generate music. However, audio contains much more information, so that text is ignored by the model at inference. We propose to balance them with a new double classifier free guidance [15] which is a general method for merging conditions with various degrees of information.

4) We introduce novel objective metrics for style conditioning, based on nearest neighbors search in the latent space, validated with human evaluations.

We compare our method to baselines which are: a MusicGen trained with CLAP embeddings [16] as conditioning, a text-to-music MusicGen used with text prompts, and a MusicGen model without conditioning used in continuation mode. We perform as well some ablation studies in order to justify the architecture of our style encoder. Based on results, we show the practicality of our methods and the musical quality of the generated music.

2. RELATED WORK

2.1 Generative models for music

Music generation models can be categorized into two types: autoregressive models and non autoregressive ones. **Autoregressive** ones are motivated by the successful work done in natural language modeling. Recent successful models use a compression model taking the form of a multi stream quantized autoencoder [10, 14] in order to convert audio into K parallel discrete streams of tokens. The K streams are obtained by performing Residual Vector Quantization (RVQ) [14] on the latent space of an autoencoder, making the first stream contain coarse information and following ones refine the approximation of the latent space. Then, an autoregressive transformer [13] is used to model these audio tokens. MusicLM [4] and MusicGen [5] are built on this principle. MusicLM uses a multi-stage approach with different models to predict the K streams, while MusicGen models them in parallel using a delay pattern [5, 17].

Non-autoregressive models such as AudioLDM2 [18], MusicLDM [19], and Stable Audio [6], are latent diffusion models operating in the latent space of a continuous variational autoencoder. Some other models use cascaded diffusion such as Noise2Music [20] to progressively increase the sampling rate of the audio. Moïsaï [21] uses a first diffusion model to compress the music and a second one to generate music from this representation and textual descriptions. MusTango [7] uses a latent diffusion model conditioned on textual description, chord, beat, tempo and key. Jen-1 [22] combines a diffusion model and a masked autoencoder trained with multi-tasks objectives. It can perform music generation, continuation and inpainting. A second version [23] uses source separation [24] over their dataset to allow the user to generate and edit music stem by stem. VampNet [25] is a masked modeling approach to music synthesis that uses masking at training and inference time in order to generate discrete audio tokens.

MAGNeT [26] is based on the same masking principle. It can also combine autoregressive and masking to reach the same quality as the autoregressive baseline (MusicGen) but with a 7x faster inference. In MeLoDy [27], a language model is used to model coarse semantic tokens and a dual path diffusion model is then used for acoustic modeling. The authors claim faster than real time generation.

2.2 Jointly trained conditioners for music generative models

Regarding the conditioning, most of the models focused on text-to-music [4, 5, 19–22]. Since pairs of text-music data are rare, most models use a pre-trained contrastive text-music model such as CLAP [16] or MuLan [28], to condition their text-to-music models. Then, massive amount of non-annotated audio data can be used at training time and text is used at inference time. However, these text-to-music models never exploit the fact that audio can be used as conditioning. For other types of conditioning, MusTango [7] is trained with text, beat tempo, key and chords as conditioning, StableAudio [6] takes timing embeddings to control the length and structure of the generated music. Some models generate stems while being conditioned on other stems. For instance, SingSong [29] generates musical accompaniments from singing and Jen-1 Composer [23] handles multi-track music generation on 4 different stems (bass, drums, instrument and melody). MusicGen [5] and Music ControlNet [30] can handle melody as conditioning and the latter can also use dynamics and rhythm. Both papers use chromagrams extraction for melody conditioning.

2.3 Conditioning a pretrained generative model

With finetuning: In Coco-Mulla [8], the authors use parameter-efficient fine-tuning (PEFT) to specialize a text-to-music MusicGen model on chords and rhythm. They finetune on a number of parameter that is 4% the amount of parameters of the original network with only 300 songs. Music ControlNet [30] is a finetuned text-to-music diffusion model that operates in the spectral domain. The finetuning strategy comes from the text-to-image method ControlNet [31] and allows to handle melody, dynamics and rhythm conditioning. The pixel-level control that allows ControlNet on images gives a pixel-level control on the mel-spectrogram.

Without finetuning: In [32], the authors use AudioLDM [18] as a backbone to perform textual inversion [9]. For each textual inversion they use a group of 5 excerpts of 10 seconds. They also try an experiment where they optimize the pseudoword S^* as well as the diffusion neural network which gives them better results. In [33], the authors use a diffusion model trained on musical data with no conditioning and perform various interactive tasks at inference which are infilling, continuation, transition (smooth a transition between two songs) and guidance. The one that is the most similar to our audio conditioning is the guidance where the diffusion model is guided by the PaSST classifier [34] embedding of an audio prompt. However the model only generates 5 seconds excerpts of music. Some

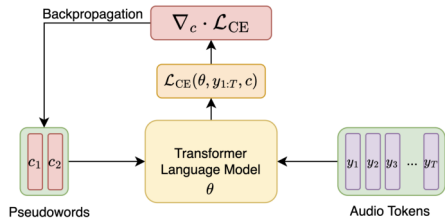


Figure 1. An overview of the Textual Inversion method based on the pretrained text-to-music MusicGen

other papers involve new control with no finetuning such as in [35] or DITTO [36] where the authors use a pre-trained text-to-music diffusion model and control its inference by optimizing the initial noise latent. In SMITIN [37], the authors control a pretrained MusicGen model by steering the attention heads in the direction that maximizes the probability of some features.

3. TEXTUAL INVERSION METHOD

We first present our textual inversion method in the case of autoregressive modeling (see Fig. 1). It is based on previous work in the image domain [9] with diffusion models.

Autoregressive modeling aims to estimate the conditional distribution of the next token y_t given the preceding tokens $y_{<t}$ and a conditioning context c , such as a textual embedding. In the framework of transformer decoder neural networks parameterized by θ , denoted as p_θ , this conditional distribution is typically modeled as a product of individual probabilities:

$$p_\theta(y_{1:T}|c) = \prod_{t=1}^T p_\theta(y_t|y_{<t}, c) \quad (1)$$

Here, $y_{1:T}$ represents the sequence of tokens, and $p_\theta(y_t|y_{<t}, c)$ denotes the probability of observing token y_t given the preceding tokens and the conditioning context. During training, with a given sequence $y_{1:T}$ and its associated textual description c , we compute the cross-entropy loss:

$$\mathcal{L}_{CE}(\theta, y_{1:T}, c) = - \sum_{t=1}^T \log p_\theta(y_t|y_{<t}, c) \quad (2)$$

It is minimized by taking a gradient descent step on $\nabla_\theta \mathcal{L}_{CE}(\theta, y_{1:T}, c)$. This loss quantifies the dissimilarity between the predicted conditional distribution and the true distribution of the next token, serving as the optimization objective for training autoregressive models.

For the textual inversion method, we take a pretrained text-to-music MusicGen for the transformer decoder. We initialize the textual embedding (for instance with the textual embedding of the word "music") c . Given a song Y , we cut it into random chunks $\{y_{1:T}^i\}_i$ and optimize the textual embedding c by taking successive gradient steps on $\nabla_c \mathcal{L}_{CE}(\theta, y_{1:T}^i, c)$. After a few hundreds iterations the learnt c is fed into the text-to-music MusicGen model to generate a song in the style of Y .

4. STYLE CONDITIONING METHOD

4.1 General Architecture

The general architecture, depicted on the left of Fig. 2, is based on the text-to-music model MusicGen [5] with the addition of a style conditioner that is jointly trained with the language model. At train time, a 30 seconds music excerpt paired with a textual description is input to the model. The textual description is fed into a frozen T5 tokenizer and transformer encoder [38]. The style encoder takes a random subsample (between 1.5 and 4.5 seconds) of the input audio and encodes it. The text and style latent representations are both projected with linear layers to have the same dimension as the transformer language model, and provided as prefix to the sequence to model.

The input audio is encoded by a pretrained EnCodec [10] model and the language model is trained in an autoregressive manner with a cross-entropy loss. In addition, the tokens that correspond to the random subsample fed into the style encoder are masked in the loss, as we noticed this reduces the tendency of the model to just copy the style audio input. At inference time, we can use text or/and a short excerpt of music as a conditioning to generate music.

4.2 Architecture of the Style Conditioner

Our style conditioner is designed with bottlenecks (RVQ [14] and downsampling) to prevent transmitting all the information of the conditioning audio excerpt to the model. Without these bottlenecks, the generative models retrieves easily the excerpt and copies it (see the ablation study in Sec. 5.5). The style conditioner depicted on the right of Fig. 2 takes an audio input of length 1.5 to 4.5 seconds, passes it through a frozen feature extractor followed by a trainable transformer encoder and a residual vector quantization (RVQ) module with 6 codebooks. After quantization, we downsample on the temporal axis to obtain a conditioning with a 5Hz frame rate which gives a similar length as a text description (8 to 25 tokens). Finally a linear layer outputs the same dimension as the language model.

The candidates for the audio encoder are an Encodec followed by trainable embeddings for each codebook that are summed, a transformer based music foundation model from [12] (we now name it MusicFM for the rest of the paper) where the authors claim state of the art on several downstream tasks specific to music information retrieval and a MERT model [11], a transformer based music model trained in a self-supervised manner. The first one has a frame rate of 50Hz and 60M parameters, the second one has a frame rate of 25Hz and 620M parameters and the third one has a frame rate of 75Hz and 95M parameters

At training time, we use dropout on the conditioning, keeping both conditions 25% of time, one of the two conditions 25% of time for each (no text or no style) or no condition 25% of time. There is also a dropout on the number of the codebooks used by the RVQ of the style conditioner: at each step of the training, the number of used codebooks is uniformly sampled between 1 and 6. Then, at inference time, we can control the bottleneck of the style conditioner.

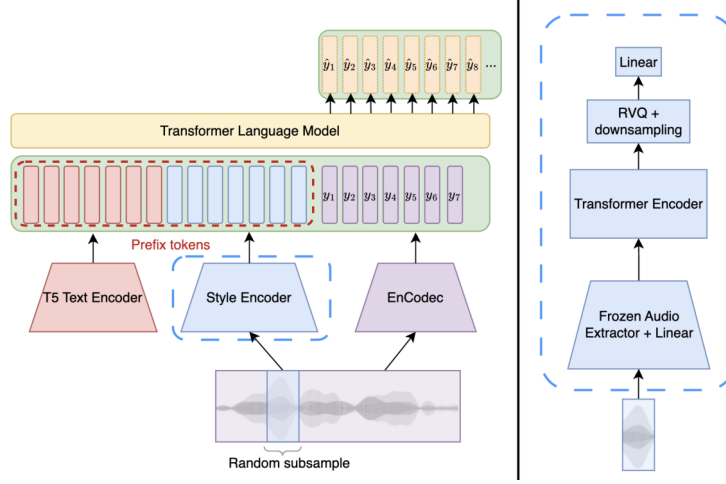


Figure 2. An overview of the general architecture. Text conditioning and style conditioning are provided to the model as a prefix. On the right we present the style conditioner.

Setting the number of codebooks to 1 gives more flexibility to the generative model whereas using 6 levels of quantization constrains it more. In practice, this means that music generated with 6 streams of quantization will sound more similar to the input condition compared to music generated with 1 stream of quantization.

4.3 Double Classifier Free Guidance

When doing next token prediction, let’s denote $l_{\text{style, text}}$ the logits of the model conditioned on style and textual description. Classifier free guidance [15] consists of pushing the logits in the direction predicted with the conditioning, to increase its importance:

$$l_{\text{CFG}} = l_{\emptyset} + \alpha(l_{\text{style, text}} - l_{\emptyset}), \text{ with } \alpha > 1, \quad (3)$$

typically, $\alpha = 3$ is used in previous work [5].

When generating music with a textual description that contradicts the audio of the style conditioning, we observe that the description is ignored by the model. This is explained by the fact that audio is more informative conditioning compared with the text, so that the model weights it more during training. To counteract this effect, we introduce a *double classifier free guidance* in which we iterate the CFG formula: we first push from style only to style and text and we then push these logits a second time from no conditioning.

$$l_{\text{double CFG}} = l_{\emptyset} + \alpha[l_{\text{style}} + \beta(l_{\text{text, style}} - l_{\text{style}}) - l_{\emptyset}] \quad (4)$$

We retrieve the simple CFG with $\beta = 1$. For $\beta > 1$, we boost the importance of the text conditioning (see Sec. 5.6).

4.4 Objective Metrics

The difficulty with generating samples in the same style of a song is that we want to generate something that is similar enough but not too close. This is something that can be subjectively evaluated. For easing the comparison of various approaches and hyper parameters, we also introduce a novel set of objective metrics.

Nearest Neighbours in Common: Let’s note $x_C \in \mathbb{R}^{D \times T}$ ($D = 1$ for mono music) the audio that we input in the style conditioner and $x_G \in \mathbb{R}^{D \times T'}$ the generated sequence. We use an encoder $E : \mathbb{R}^{D \times T} \rightarrow \mathbb{R}^N$ which outputs a single vector whatever the input length T is. In practice, this is done by taking a MusicFM model and averaging on the time dimension. Then, for each song of our valid and test sets, we cut it into chunks of 30 seconds and store the embeddings $\{E_{i,j}\}$, i being the index of the song and j the chunk number. For $E_C = E(x_C)$, we compute the cosine similarities $\cos(E_C, E_{i,j}), \forall i, j$ and the set of its K nearest neighbors: $\{i_1^C, \dots, i_K^C\}$. We do the same for $E_G = E(x_G)$ and obtain a set of K values $\{i_1^G, \dots, i_K^G\}$. We then have found the nearest songs in the dataset. We define our metric $\text{KNN}_{\text{common}}(x_C, x_G)$ for a song x_G that has been generated after being conditioned by x_C :

$$\text{KNN}_{\text{common}}(x_C, x_G) = \frac{|\{i_1^C, \dots, i_K^C\} \cap \{i_1^G, \dots, i_K^G\}|}{K} \in [0, 1]. \quad (5)$$

The intuition behind this metric is that a model performs well at recreating a song in the style of another if the generated song and its conditioning audio have embeddings that are close enough to share neighbors in the dataset. However, if a model copies the conditioning (i.e. $x_G \approx x_C$) the metric will tend to 1, we thus need a second metric to avoid x_G and x_C being too similar.

G is the Nearest Neighbor of C: We want E_G and E_C to be close while being different. One way to be sure that the corresponding audios are not too similar is to check that if we add E_G to the set of embeddings $\{E_{i,j}\}$, E_G is not the nearest neighbor of E_C . Assuring that another song from the dataset is closer to the conditioning means that the model is creative enough and not just copying its input. Formally, denoting $\{E_{\cup}\} = \{E_{i,j}\} \cup \{E_G\}$, we define

$$\text{KNN}_{\text{overfit}}(x_C, x_G) = \begin{cases} 1 & \text{if } \operatorname{argmax}_{E \in \{E_{\cup}\}} [\cos(E_C, E)] = E_G \\ 0 & \text{otherwise.} \end{cases} \quad (6)$$

Model	FAD _{vgg} ↓	KL ↓	CLAP ↑	KNN _{common} ↑	KNN _{overfit} ↓	OVL ↑	SIM ↑	VAR ↓
Textual Inversion	6.07	0.55	0.20	0.20	0.14	78.11 ± 0.93	61.78 ± 1.06	69.53 ± 1.44
MusicGen Continuation	1.22	0.51	0.30	0.26	0.17	83.95 ± 0.83	73.38 ± 0.97	77.24 ± 1.29
MusicGen w. audio CLAP	0.96	0.43	0.31	0.09	0.02	84.76 ± 0.93	62.37 ± 1.04	68.58 ± 1.42
Our Model w. EnCodec, 2 RVQ	0.85	0.49	0.29	0.23	0.12	83.41 ± 1.04	72.16 ± 0.93	72.39 ± 1.33

Table 1. Comparison with baselines. The KNN* metrics, introduced in Sec. 4.4, measure how close the generation is from the style condition, yet different from the matching ground truth. Those are completed with the subjective evaluations from Sec. 4.5. While using MusicGen for continuation matches well to the style audio, it has limited variation. Using a CLAP audio encoder as conditioning does the opposite, while using our style encoder gets the right balance between the two.

For our evaluations, we take 1000 samples of 3 seconds x_C from our test set, generate the corresponding x_G and average the two KNN metrics. Intuitively, the two metrics are positively correlated, but for a similar value for KNN_{common} we will favor the model that minimizes KNN_{overfit}.

Other Objective Metrics To evaluate the quality of the generated music, we also use the official implementation of the Fréchet Audio Distance defined in [39] that uses a VG-Gish model, the KL-divergence based metric introduced in [5] that computes the KL-divergence on the probabilities of the labels of a pretrained audio classifier between the conditioning and the generated music. We noticed that a high FAD (> 2) often indicates a poor quality of the generated samples. The CLAP score [5, 16] computes the cosine similarity between the description and the audio embeddings obtained with the CLAP model. A higher score indicates that the generated audio aligns well with the textual description of the conditioning audio.

4.5 Human studies metrics

We follow a similar protocol as in [5] for the human studies. We ask human raters to evaluate three different aspects of the generated audio: (1) How would you rate the overall quality of this excerpt [OVL]? (2) Without considering audio quality, how similar are these two excerpts in terms of style [SIM]? (3) Without considering audio quality, how likely do you think these two excerpts are from the same song [VAR]?

We believe that the SIM and VAR scores are the subjective versions of KNN_{common} and KNN_{overfit}.

5. EXPERIMENTAL RESULTS

5.1 Hyperparameters for the textual inversion

For the textual inversion method we test different parameters sets and retain these ones: we use a 12 tokens sentence for initialization, a batch size of 8 with 5 seconds segments randomly sampled from a 30 second excerpt with 200 optimization steps, a learning rate of 0.025 with a vanilla Adam optimizer. Finally the main issue that we encounter with this method is its instability. It is hard to find a set of hyperparameters that works well for any song. Some songs seem to be easier to invert for different sets of hyperparameters. For some song, we never achieve to obtain hearable music as the result suffers from glitches, and tempo instabilities. Finally, it seems beneficial to augment the length of the text embedding, as well as performing the inversion

over chunks taken from a 30 seconds excerpt rather than the entire song. The results are shown in Tab. 1.

5.2 Hyperparameters for the style conditioner

All the models that we train are medium size (1.5B parameters) MusicGen models built on top of the 4 stream 32kHz music version of EnCodec [10]. All models are trained for 400K steps on 64 V100 GPUs with the AdamW optimizer using $\beta_1 = 0.9$, $\beta_2 = 0.95$, a batch size of 192, and music sequences of 30 seconds. For the style conditioner, excerpts between 1.5 and 4.5 seconds are subsampled from the original sequence, the transformer encoder has 8 layers, 8 heads, a dimension of 512 and is non-causal, the residual vector quantizer has a codebook size of 1024, 6 streams and a variable number of streams is sampled at each training step, hence allowing the language model to train on all the levels of quantization. The style tokens are downsampled to 5Hz. All our evaluations are done on 1000 samples of the test set. Similarly to the MusicGen Melody model, both the textual description and the style condition are provided as prefix to the language model.

5.3 Datasets

We use 20K hours of licensed music as in [5]. The training dataset is composed of 25K and 365K songs from the Shutterstock and Pond5 music data collections, as well as 10k tracks of an internal dataset. Each song comes with textual description, and is downsampled to 32kHz mono.

5.4 Comparison with baselines and model selection

Apart from the closed-source model udio [40], there is no other audio conditioned music generative model. We use as a baseline a MusicGen model in the continuation setting: given 3 seconds of music, we ask MusicGen to continue the music with no textual prompt. For the second one we train a MusicGen model with a pretrained CLAP audio encoder [16] as conditioning, also taking 3 seconds of audio as input. In Tab. 1, we compare these two baselines with our model with the EnCodec feature extractor for the style conditioner, a quantization level of 2 and with a textual inversion model. We notice that the FAD correlates well with the quality metric (OVL) since the textual inversion model has the worst OVL and FAD scores. Thus excluding this approach, we observe that the KNN_{common} and the SIM metrics ranks the models in the same orders as well as the KNN_{overfit} and VAR metrics.

Feat. Ext.	Quant.	FAD _{vgg} ↓	KL ↓	CLAP ↑	KNN _{common} ↑	KNN _{overfit} ↓	OVL ↑	SIM ↑	VAR ↓
MERT	1	0.78	0.50	0.29	0.19	0.06	84.07 ± 0.93	70.27 ± 1.22	69.69 ± 1.31
MERT	2	0.75	0.47	0.30	0.24	0.13	84.14 ± 0.96	72.53 ± 1.05	72.81 ± 1.21
MERT	4	0.75	0.45	0.31	0.29	0.18	84.32 ± 1.04	74.15 ± 0.96	75.12 ± 1.35
EnCodec	2	0.85	0.49	0.29	0.23	0.12	84.02 ± 0.89	72.69 ± 0.91	72.47 ± 1.28
MusicFM	2	0.70	0.45	0.31	0.28	0.16	84.45 ± 1.09	73.01 ± 0.95	74.01 ± 1.36

Table 2. Comparison between the 3 feature extractors. The human studies correlate well with the KNN metrics. As expected, using coarser quantization of the style features leads to more variations in the generated audio. Self-supervised encoder like MERT and MusicFM outperforms low level acoustic models like EnCodec.

Model	FAD _{vgg} ↓	KL ↓	CLAP ↑	KNN _{common} ↑	KNN _{overfit} ↓
Our Model	0.75	0.45	0.31	0.29	0.18
Smaller Transformer	0.98	0.48	0.29	0.24	0.13
No Transformer	2.92	0.96	0.13	0.01	0.0
No Masking of the loss	1.11	0.53	0.29	0.22	0.30

Table 3. Ablation Study on our model with a MERT feature extractor with 4 quantization streams.

Regarding the baselines, the textual inversion method provides results of poor quality (FAD). The continuation method provides music that has a high similarity to the conditioning (high KNN_{common} and SIM) but that is too similar to it (high KNN_{overfit} and VAR). However, the CLAP conditioning captures a more vague style of the conditioning and generates music that is too far from it (low KNN_{common}, KNN_{overfit}, SIM and VAR). Our model with the EnCodec feature extractor and 2 levels of quantization strikes the right balance between these two baselines.

In order to strengthen our claim that our KNN metrics correlates well with human perception of closeness between musical excerpts, we showcase a second study in Tab. 2. In this study we compare the metrics of the MERT feature extractor with 3 quantization levels 1, 2, 4 (we recall that the models can go up to 6) as well as the EnCodec and MusicFM feature extractors with a quantization level of 2. All models generate music of similar quality (FAD and OVL). We notice that when the bottleneck is larger (i.e. when the quantization level is higher), the KNN_{common} augments. This follows the intuition that if the conditioner transmits more information to the language model, the generated music will be closer to the input condition. The models follows similar orders for KNN_{common} and SIM as well as for KNN_{overfit} and VAR.

5.5 Ablation Study

We perform an ablation study in Tab. 3 on the components of the style conditioner with MERT as a feature extractor, and 4 RVQ streams. When reducing the size of the transformer encoder from 8 layers and 512 dimensions to 4 layers and 256 dimensions, the quality of the generated audio is worse. When removing the transformer encoder, the model generates audio that is far from music (high FAD). When we don't mask the music that is input to the style conditioner in the cross-entropy loss at training time, the audio quality is slightly worse and the model generates music that is too close to the conditioning and tends to loop. The very high KNN_{overfit} indicates it since for a KNN_{common} lower than the best model the KNN_{overfit} is twice its value.

Type	α	β	FAD _{vgg} ↓	CLAP ↑	KNN _{common} ↑
No CFG	X	X	1.54	0.25	0.088
simple	3	X	0.92	0.28	0.162
double	3	3	0.80	0.35	0.123
double	3	4	0.78	0.37	0.104
double	3	5	0.84	0.37	0.095
double	3	6	0.97	0.38	0.081

Table 4. Classifier Free Guidance parameters tuning. Larger β from (4) leads to increasing the importance of the text conditioning (given by the CLAP score), and decreasing the similarity to the style audio, given by KNN_{common}.

5.6 Tuning the Classifier Free Guidance

When style and text conditioning are both used and are not consistent, it is necessary to use double CFG instead of simple CFG so that the text is not ignored. To tune the parameters α, β of the double classifier free guidance given by (4), we rely on the following protocol. For 1000 samples of our test set, we randomly shuffle text descriptions and generate music while conditioning both on text and music. We track the FAD [39], the KNN_{common} and the CLAP score. In Tab 4 we observe the intuitive fact that the KNN_{common} and CLAP score are negatively correlated: if the balancing favors the text condition the CLAP score is higher, if it favors the audio condition the KNN_{common} is higher. The double CFG thus works as expected.

6. CONCLUSION

In this paper we introduced style conditioning for language model based music generative models: given a few seconds of a musical excerpt, one can generate music in the same style using our proposed audio encoder with an information bottleneck. We introduced new metrics to assess the equilibrium between generating music that maintains a similar style to the condition while also being different. We validated those with human studies. Finally, we can also mix this style conditioning with inconsistent textual description and balance them thanks to a new double classifier free guidance method. This method could be applied in other generative models with multiple conditions.

Ethical statement: Improving music generation brings ethical challenges. Through carefully chosen bottlenecks in our style extractor (RVQ, downsampling) we aim for the right balance between increasing the model controllability and possible creative use while ensuring the model does not copy existing works, and provided new metrics to measure this. Finally, we only used music we licensed.

7. REFERENCES

- [1] J. Engel, K. K. Agrawal, S. Chen, I. Gulrajani, C. Donahue, and A. Roberts, “Gansynth: Adversarial neural audio synthesis,” in *ICLR*, 2019.
- [2] S. Rouard and G. Hadjeres, “Crash: Raw audio score-based generative modeling for controllable high-resolution drum sound synthesis,” in *ISMIR*, 2021.
- [3] L.-C. Yang, S.-Y. Chou, and Y.-H. Yang, “Midinet: A convolutional generative adversarial network for symbolic-domain music generation,” in *ISMIR*, 2017.
- [4] A. Agostinelli, T. I. Denk, Z. Borsos, J. Engel, M. Verzetti, A. Caillon, Q. Huang, A. Jansen, A. Roberts, M. Tagliasacchi, M. Sharifi, N. Zeghidour, and C. Frank, “Musiclm: Generating music from text,” *ArXiv*, 2023.
- [5] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Défossez, “Simple and controllable music generation,” in *Neurips*, 2023.
- [6] Z. Evans, C. Carr, J. Taylor, S. H. Hawley, and J. Pons, “Fast timing-conditioned latent audio diffusion,” *ArXiv*, 2024.
- [7] J. Melechovsky, Z. Guo, D. Ghosal, N. Majumder, D. Herremans, and S. Poria, “Mustango: Toward controllable text-to-music generation,” *ArXiv*, 2023.
- [8] L. Lin, G. Xia, J. Jiang, and Y. Zhang, “Content-based controls for music large language modeling,” *ArXiv*, 2023.
- [9] R. Gal, Y. Alaluf, Y. Atzmon, O. Patashnik, A. H. Bermano, G. Chechik, and D. Cohen-Or, “An image is worth one word: Personalizing text-to-image generation using textual inversion,” in *ICLR*, 2023.
- [10] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, “High fidelity neural audio compression,” *TMLR*, 2022.
- [11] Y. Li, R. Yuan, G. Zhang, Y. Ma, X. Chen, H. Yin, C. Lin, A. Ragni, E. Benetos, N. Gyenge, R. Dannenberg, R. Liu, W. Chen, G. Xia, Y. Shi, W. Huang, Y. Guo, and J. Fu, “Mert: Acoustic music understanding model with large-scale self-supervised training,” *ArXiv*, 2023.
- [12] M. Won, Y.-N. Hung, and D. Le, “A foundation model for music informatics,” *ICASSP 24*, 2024.
- [13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in Neural Information Processing Systems*, 2017.
- [14] N. Zeghidour, A. Luebs, A. Omran, J. Skoglund, and M. Tagliasacchi, “Soundstream: An end-to-end neural audio codec,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, 2022.
- [15] J. Ho and T. Salimans, “Classifier-free diffusion guidance,” in *NeurIPS Workshop on Deep Generative Models and Downstream Applications*, 2021.
- [16] Y. Wu, K. Chen, T. Zhang, Y. Hui, T. Berg-Kirkpatrick, and S. Dubnov, “Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation,” *ArXiv*, 2023.
- [17] E. Kharitonov, A. Lee, A. Polyak, Y. Adi, J. Copet, K. Lakhotia, T. A. Nguyen, M. Riviere, A. Mohamed, E. Dupoux, and W.-N. Hsu, “Text-free prosody-aware generative spoken language modeling,” in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, S. Muresan, P. Nakov, and A. Villavicencio, Eds. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 8666–8681.
- [18] H. Liu, Q. Tian, Y. Yuan, X. Liu, X. Mei, Q. Kong, Y. Wang, W. Wang, Y. Wang, and M. Plumbley, “Audioldm 2: Learning holistic audio generation with self-supervised pretraining,” *ArXiv*, 2023.
- [19] K. Chen, Y. Wu, H. Liu, M. Nezhurina, T. Berg-Kirkpatrick, and S. Dubnov, “Musicldm: Enhancing novelty in text-to-music generation using beat-synchronous mixup strategies,” *ArXiv*, 2023.
- [20] Q. Huang, D. S. Park, T. Wang, T. I. Denk, A. Ly, N. Chen, Z. Zhang, Z. Zhang, J. Yu, C. Frank, J. Engel, Q. V. Le, W. Chan, Z. Chen, and W. Han, “Noise2music: Text-conditioned music generation with diffusion models,” *ArXiv*, 2023.
- [21] F. Schneider, O. Kamal, Z. Jin, and B. Schölkopf, “Moûsai: Text-to-music generation with long-context latent diffusion,” *ArXiv*, 2023.
- [22] P. Li, B. Chen, Y. Yao, Y. Wang, A. Wang, and A. Wang, “Jen-1: Text-guided universal music generation with omnidirectional diffusion models,” *ArXiv*, 2023.
- [23] Y. Yao, P. Li, B. Chen, and A. Wang, “Jen-1 composer: A unified framework for high-fidelity multi-track music generation,” *ArXiv*, 2023.
- [24] S. Rouard, F. Massa, and A. Défossez, “Hybrid transformers for music source separation,” in *ICASSP 23*, 2023.
- [25] H. F. Garcia, P. Seetharaman, R. Kumar, and B. Pardo, “Vampnet: Music generation via masked acoustic token modeling,” *ArXiv*, 2023.
- [26] A. Ziv, I. Gat, G. L. Lan, T. Remez, F. Kreuk, A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, “Masked audio generation using a single non-autoregressive transformer,” in *ICLR*, 2024.

- [27] M. W. Y. Lam, Q. Tian, T. Li, Z. Yin, S. Feng, M. Tu, Y. Ji, R. Xia, M. Ma, X. Song, J. Chen, Y. Wang, and Y. Wang, “Efficient neural music generation,” in *Neurips*, 2023.
- [28] Q. Huang, A. Jansen, J. Lee, R. Ganti, J. Y. Li, and D. P. W. Ellis, “Mulan: A joint embedding of music audio and natural language,” in *ISMIR*, 2022.
- [29] C. Donahue, A. Caillon, A. Roberts, E. Manilow, P. Esling, A. Agostinelli, M. Verzetti, I. Simon, O. Pietquin, N. Zeghidour, and J. Engel, “Singsong: Generating musical accompaniments from singing,” *ArXiv*, 2023.
- [30] S.-L. Wu, C. Donahue, S. Watanabe, and N. J. Bryan, “Music controlnet: Multiple time-varying controls for music generation,” *ArXiv*, 2023.
- [31] L. Zhang, A. Rao, and M. Agrawala, “Adding conditional control to text-to-image diffusion models,” in *ICCV*, 2023.
- [32] M. Plitsis, T. Kouzelis, G. Paraskevopoulos, V. Katsouros, and Y. Panagakis, “Investigating personalization methods in text to music generation,” *ArXiv*, 2023.
- [33] M. Levy, B. D. Giorgi, F. Weers, A. Katharopoulos, and T. Nickson, “Controllable music production with diffusion models and guidance gradients,” *ArXiv*, 2023.
- [34] K. Koutini, J. Schlüter, H. Eghbal-zadeh, and G. Widmer, “Efficient training of audio transformers with patchout,” in *Interspeech 2022*. ISCA, Sep. 2022.
- [35] H. Manor and T. Michaeli, “Zero-shot unsupervised and text-based audio editing using ddpm inversion,” in *ICML*, 2024.
- [36] Z. Novack, J. McAuley, T. Berg-Kirkpatrick, and N. J. Bryan, “Ditto: Diffusion inference-time t-optimization for music generation,” *ArXiv*, 2024.
- [37] J. Koo, G. Wichern, F. G. Germain, S. Khurana, and J. L. Roux, “Smitin: Self-monitored inference-time intervention for generative music transformers,” 2024. [Online]. Available: <https://arxiv.org/abs/2404.02252>
- [38] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, “Exploring the limits of transfer learning with a unified text-to-text transformer,” *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, 2020.
- [39] K. Kilgour, M. Zuluaga, D. Roblek, and M. Sharifi, “Fréchet audio distance: A metric for evaluating music enhancement algorithms,” *ArXiv*, 2019.
- [40] Udio. [Online]. Available: <https://www.udio.com>

VARIATION TRANSFORMER: NEW DATASETS, MODELS, AND COMPARATIVE EVALUATION FOR SYMBOLIC MUSIC VARIATION GENERATION

Chenyu Gao¹ Federico Reuben¹ Tom Collins^{2,3}

¹ School of Arts and Creative Technologies, University of York, UK

² Frost School of Music, University of Miami, FL, USA

³ MAIA, Inc., Davis, CA, USA

{chenyu.gao, federico.reuben}@york.ac.uk, tomthecollins@gmail.com

ABSTRACT

Variation in music is defined as repetition of a theme, but with various modifications, playing an important role in many musical genres in developing core music ideas into longer passages. Existing research on variation in music is mostly confined to datasets consisting of classical theme-and-variation pieces, and generative models limited to melody-only representations. In this paper, to address the problem of the lack of datasets, we propose an algorithm to extract theme-and-variation pairs automatically, and use it to annotate two datasets called POP909-TVar (2,871 theme-and-variation pairs) and VGMIDI-TVar (7,830 theme-and-variation pairs). We propose both non-deep learning and deep learning based symbolic music variation generation models, and report the results of a listening study and feature-based evaluation for these models. One of our two newly proposed models, called Variation Transformer, outperforms all other models that listeners evaluated for “variation success”, including non-deep learning and deep learning based approaches. An implication of this work for the wider field of music making is that we now have a model that can generate material with stronger and perceivably more successful relationships to some given prompt or theme.¹

1. INTRODUCTION

The term variation refers to “a form founded on repetition, and as such an outgrowth of a fundamental musical and rhetorical principle, in which a discrete theme is repeated several or many times with various modifications” [1]. In western music, variation is a technique in which the theme is repeated but in an alternate form with various modifications in one or more aspects of melody, rhythm, harmony,

¹Demos: <https://variation-transformer.glitch.me>.
Code and datasets: <https://github.com/ChenyuGAO-CS/Variation-Transformer>.



© C. Gao, F. Reuben, and T. Collins. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** C. Gao, F. Reuben, and T. Collins, “Variation Transformer: New datasets, models, and comparative evaluation for symbolic music variation generation”, in *Proc. of the 25th Int. Society for Music Information Retrieval Conf.*, San Francisco, United States, 2024.

texture, instrumentation, etc. An example from game music is provided in Figures 1(a) and (b), where the top and bottom staves in (b) are embellished compared to (a), but the melodic and harmonic structure is largely the same.

In recent years, a number of music generation algorithms and commercialised artificial intelligence (AI) music generation systems have emerged [2–9], but there are only a few studies focusing on symbolic music variation generation [10–14]. Although some infilling systems claim that they have the potential to generate variations [7, 8, 15, 16], an infilling system may sometimes work to continue writing [17] rather than always varying the theme (i.e., generate content with a strong relationship to the given prompt). Accepting a musical input prompt but destroying the original music idea is a “lack of control” issue, and could frustrate composers [18, 19]. It also leaves the presence and perception of rhetorical or narrative content to serendipity (chance), which goes against the rhetorical principle of the definition of musical variation.

Existing music variation research is mostly confined to datasets consisting of classical theme-and-variation pieces [20] or monophonic folk music [21], and most of existing music variation generation models are also limited to varying melody only [11, 12, 14].

To address the issues above, in this paper we develop both new datasets and models for symbolic polyphonic music variation generation. For data annotation, we develop an algorithm for theme-and-variation extraction, and apply it to annotate two datasets: POP909 [22], and VGMIDI [23]. For model design, we propose both deep and non-deep learning-based models, as another shortcoming of recent research is that evaluations ignore models published prior to c. 2015 – assuming, rather than actually testing whether, deep learning approaches are superior for music generation [13, 14, 24]. Three research questions are addressed: **RQ1:** To what extent can AI models generate successful music variations? **RQ2:** Can deep learning approaches outperform non-deep learning approaches on music variation generation? **RQ3:** Would variation generation tools be useful? We conduct a listening study and feature-based evaluation to address these research questions, and finish by discussing the implications of the study’s findings for music generation and the field of MIR.

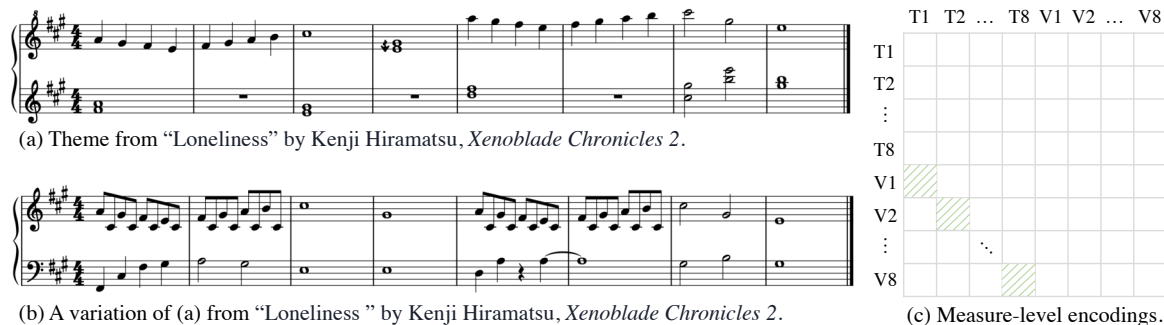


Figure 1. (a) Theme from “Loneliness” by Kenji Hiramatsu, *Xenoblade Chronicles 2*; (b) Variation from the same piece; (c) Measure-level encodings based on this theme-variation example, discussed in a subsequent section. T_m denotes the encoding of the m th measure of the theme, while V_m denotes the encoding of the m th measure of the variation. The shaded areas are filled with 1s, while the blank areas are filled with 0s, indicating how we condition the model to attend to specific parts of the theme when generating a variation.

2. RELATED WORK

2.1 Symbolic music generation approaches

Before deep learning models became a popular approach for music generation, many models were based on Markov chains [25–28]. Markov models assume that the current state prediction depends on one (first-order Markov model) or more (second-, third-order models) previous states. They have been used to generate music in many styles, and recent work [29] finds evidence that ratings of the stylistic success of their outputs are on a par with deep learning models such as Music Transformer [30].

Among a large number of deep learning approaches to symbolic music generation [30–40], the most popular architectures are the generative adversarial network (GAN) [41], variational autoencoder (VAE) [31, 42, 43], and transformer [44]. More recently, there are also some attempts to adopt the diffusion model [45, 46] to generate music [39, 40]. But to date, we observed these diffusion-based algorithms suffer from the problem of a lack of structure in long-term music generation.

2.2 Symbolic music variation generation

As a sub-task of symbolic music generation, symbolic music *variation* generation takes a prompt/theme as input, and aims to generate variations where the new material is different to the theme but remains musically relatable. Some variation generation approaches are based on genetic algorithms [10, 11], but drawbacks are that they have only been applied to sequential representations of monophonic music and are reliant on manually designed rules [47].

There are also variation generation methods based on probabilistic methods. For example, an algorithm mentioned in [48] starts with the same first beat as the theme, and subsequent beats are generated by a Markov model. To ensure the generated variation begins and ends somewhere “sensible”, the Markov process can be run forwards and backwards from such start and end points, with a join in the middle of a generated phrase that may break the Markov property [26, 48]. Compared to variation generation, these two methods are more like style-composing. In

contrast, an idea in [49] is to decide if each of the states in an existing sequence (such as a theme) should be replaced by another state according to a corresponding probability distribution. However, this approach has only ever been applied to monophonic variation generation.

Compared to non-deep learning approaches, an advantage of deep learning for music generation is that it places less emphasis on the domain knowledge/expertise of the programmer – the trained network weights *should* take on responsibility for generalizing the style/structure of the training data.

There are only a few studies for music variation generation that have adopted deep learning methods. This is because most deep learning approaches are data-driven, and existing theme-and-variation datasets are either relatively small classical music datasets (e.g., the TAVERN dataset [20] with 17 works by Beethoven and 10 by Mozart for a total of 281 variations) or monophonic folk music datasets [21], which restricts the development of deep methods for music variation generation. Although Music Transformer is adopted in [13] for jazz variation generation, the JAZZVAR dataset (502 theme-variation pairs) proposed in this study is still relatively small for deep learning model training. Besides, the lack of listening studies and comparative evaluation makes it difficult to conclude to what extent these models are effective in generating musical variations [13, 14, 24].

3. DATASET

In this section, we introduce our algorithm for automatic extraction of variations from a collection containing annotated themes. We use it to extract theme-and-variations (TVar) pairs from the POP909 [22] and VGMIDI [23] datasets. As a result, two new datasets (POP909-TVar, and VGMIDI-TVar) are constructed.

3.1 Construction of the POP909-TVar dataset

The POP909 dataset [22] contains piano arrangements of 909 Chinese popular songs in MIDI format. We use 809 pieces ($\sim 90\%$) for training, and 100 ($\sim 10\%$) for testing.

As repetitive phrase annotations are provided in the POP909 dataset [50], we use these to estimate the lower and upper bounds of the similarity between human-composed themes and variations by utilizing a symbolic fingerprinting-based similarity calculation [51, 52]. The first occurrence of each repetitive pattern is regarded as the theme, and the following occurrences are regarded as variations. For each theme, we record the minimum and maximum similarity scores between it and its variations. The similarity lower bound is the average of the per-theme minimum scores, and the upper bound is defined correspondingly, with values of 53.03 and 70.95, respectively.

Algorithm 1 TVar extraction on the POP909 dataset

Input: Repetitive pattern labels (\mathbf{P}) and MIDIs (\mathbf{M}) of the dataset, similarity upper bound u and lower bound l

Output: TVar pairs

```

1: for  $p \in \mathbf{P}$  do
2:   Separate the first occurrence of  $p$  as the theme  $t$ , and
   the subsequent occurrences as an array  $\mathbf{V}_{\text{Rep}}$ 
3:    $\mathbf{V}_{\text{Match}} \leftarrow \text{match\_occ}(t, \mathbf{M}, u, l)$ 
4:   Push  $\mathbf{V}_{\text{Match}}$  into  $\mathbf{V}_{\text{Rep}}$ 
5:   for  $v \in \mathbf{V}_{\text{Rep}}$  do
6:     if  $\text{similarity}(t, v) > u$  then
7:       Filter out  $v$ 
8:     else
9:       if Similarity score between  $v$  and the previous
       occurrence  $> u$  then
10:        Filter out  $v$ 
11:       else
12:        Push  $v$  into  $\mathbf{V}_{\text{Out}}$ 
13:       if Occurrence count of  $\mathbf{V}_{\text{Out}} \geq 1$  then
14:         return  $t, \mathbf{V}_{\text{Out}}$ 

```

The pseudocode for TVar extraction is given in Algorithm 1. We take the first occurrence of repetitive patterns as themes when applying our algorithm to POP909 (line 2).² Variations are extracted from both human-annotated patterns (line 2) and the whole dataset (line 3). When extracting variations from human annotations, we exclude variations whose similarity score is larger than the similarity upper-bound (lines 6-7), since we aim to train models to generate variations where there is new but theme-relatable material. When extracting variations of a theme on the whole dataset, we run a symbolic fingerprinting-based pattern-matching approach [51–53] using the same lower and upper bounds mentioned previously to retain variations (line 3). We also filter out variations that are too similar to existing variations (lines 9-12).

The POP909-TVar dataset is constructed by applying our TVar extraction algorithm to POP909, giving 2,609 TVar pairs in the training set, and 262 TVar pairs in the test set.

² First occurrences are not always the *archetypal occurrence*, but it is a reasonable assumption [17].

3.2 Construction of the VGMIDI-TVar dataset

The VGMIDI dataset [23] contains piano arrangements of game music in MIDI format recorded by human performers.³ There are three subsets in VGMIDI: the largest has 2,520 MIDI files for music generation model training, the second one has 136 MIDI files with emotion labels, and the third one (272 MIDI files) is for music discriminator training, which involves both human-composed music and fake data. Here, we merged the largest subset and the subset with emotional labels and adhered to the original train-test split, obtaining 2,301 MIDI files for training and 355 for testing.

Compared to popular music, we infer there could be greater scope for new material in variations in game music, so we reduce the similarity lower bound to 30 but keep the similarity upper bound as 70.95. Also, we restrict the extracted variation and the theme to come from the same song. Then, we follow the steps as in Section 3.1 to obtain variations. In contrast to the POP909 dataset, repetitive patterns are not annotated in the VGMIDI dataset, so we run a slice window with size = 8 measures and step = 4 measures from the beginning to the end of the song to extract theme samples. The similarity between each new theme and previous themes is calculated to filter out theme samples that are too similar (similarity score > upper-bound) to existing themes. The variation extraction function `match_occ()` is applied to each of the theme samples, and then the matched occurrences will be filtered by the same processes as that in Algorithm 1 (lines 5-12). Only theme samples with more than one variation will be retained (lines 13-14 in Algorithm 1).

The VGMIDI-TVar dataset is constructed by applying the above steps to VGMIDI, giving 6,790 TVar pairs in the training set, and 1,040 in the test set.

4. MUSIC VARIATION GENERATION MODELS

In this section, we introduce two new music variation generation models: one is a deep-learning model called Variation Transformer, and the other acts as a non-deep learning baseline called Variation Markov.

4.1 Variation Transformer

Variation Transformer builds on Music Transformer [30], utilizing the REMI representation [54] to encode incoming MIDI files. The design of the relative positional self-attention [55] alleviates the problems of the regular self-attention that attends only locally or at the beginning for a sequence [56] – Music Transformer is used for jazz variation generation in [13]. However, while developing and testing these models, we observed that Music Transformer’s ability to understand the measure-wise relationship between theme and variation was not strong enough. For example, when generating a variation of an 8-measure theme, Music Transformer might generate something new in the first 2 measures, but copy large sections of the theme

³ Sources for this dataset are <https://www.vgmusic.com> and <https://www.ninsheetmusic.org>.

in the following measures [52, 57], showing the failure of the Music Transformer model to learn the theme-and-variation relationship. When a human composer creates a variation, commonly each bar of the variation is related to the corresponding measure of the theme (recall Figures 1(a) and (b)). Thus, in this study, we propose the measure-level encodings (Figure 1 (c)) and **theme-and-variation Attention** (tvAttn) to force the transformer architecture to take into account more information about a specific measure of an existing theme when generating the corresponding measure of a new variation, calling our new model the Variation Transformer.

Figure 1(c) shows the measure-level encoding (which we will notate E_{bar}) to capture the relationship between corresponding measures of theme and variation, with a size of $N \times N$, where N is the length of the encoding of theme concatenated with variation. The formula for tvAttn is then

$$\text{tvAttn} = \text{Softmax} \left((1 + \mathbf{w}E_{\text{bar}}) \frac{QK^\top + S^{\text{rel}}}{\sqrt{D_h}} \right) V, \quad (1)$$

where \mathbf{w} is a learnable parameter, and E_{bar} is the measure-level encodings. Q represents the queries, K is the set of keys, V is the set of values, $1/\sqrt{D_h}$ is a scale factor, and S^{rel} is to encode the relative positional information between each pair of tokens in a sequence.

4.2 Variation Markov

Based on [26, 58] and inspired by [12, 48], we propose a non-deep learning music variation generation strategy based on Markov models. Polyphonic MIDI inputs are represented as states in a state space consisting of beat in the measure and MIDI note numbers relative to estimated tonal center. The transitions between states observed across our training data are stored in a directed graph.

When generating variations, we extract the beginning and end states of each measure of the theme, and run a “scenic pathfinding algorithm” to find replacement states. This algorithm is adapted from Dijkstra’s shortest path algorithm [59]. When finding the shortest path between connected vertices u and v in a graph G , Dijkstra’s method always updates the distance from the starting vertex u to other vertices with shorter distances. In our scenic version, we insert an extra piece of logic to determine whether the distance from the starting vertex u to another vertex will be updated to a shorter distance with probability $p = .5$. In this way, more varied musical content is generated, because the path connecting u to v that results on each occasion is not necessarily the shortest. We replace each measure from the theme with the scenic path alternative with probability $q = .5$, and if u and v are not connected (due to not being observed in a training data sequence), then we retain the original measure from the theme.

5. EVALUATION

5.1 Experimental design

We conduct a listening study and feature-based evaluation on both POP909-TVar and VGMIDI-TVar datasets. The

variation generation ability of three transformer-type models (TTMs) – fast-Transformer (FaTr) [37, 60, 61], Music Transformer (MuTr) [13, 30], and Variation Transformer (VaTr) – and Variation Markov (VaMa) is compared.

For fair comparison, we use the REMI representation [54] to represent MIDI files for all three TTMs, which were trained on A40 GPUs with a batch size of 16 for 10 epochs on each of the two training sets. The learning rate is set as 1×10^{-4} for the first 5 epochs, then decreased to 5×10^{-5} for the last 5 epochs. For model training, we concatenate each theme-and-variation pair, with a [Separate] token inserted between the theme and variation. Ten variations were generated by each algorithm with using each theme in the test set as an initial prompt to provide the pool of stimuli for evaluation.

For hypothesis testing, we utilize a Bayes factor analysis (BFA, [62]), where the ratio of the marginal likelihoods of the alternative hypothesis H_1 to the null hypothesis H_0 is calculated, and notated BF_{10} . A large value of BF_{10} suggests there is strong evidence for H_1 . Conversely, a small BF_{10} suggests strong evidence for H_0 . A table for interpreting BF_{10} values is provided in [63]. BFA is superior to classical (frequentist) hypothesis testing, because of this ability to find evidence in favor of the null, which in (computational) systems testing corresponds to a meaningful non-difference between systems.

5.2 Listening study

Our listening study is approved by the Ethics Committee of the School of Arts and Creative Technologies at the University of York. Our overall design builds on previous listening studies in this domain (e.g., [49]), and our hypotheses are as follows:

1. In terms of variation success, we predict the following ordering of systems: VaTr > MuTr > FaTr > VaMa.
2. TTMs (VaTr, MuTr, and FaTr) achieve better music quality than VaMa.

5.2.1 Participants

We aim to recruit participants with a relatively high level of music knowledge, using student email lists at the University of York, and the C4DM group of the Queen Mary University of London.⁴ Participants are compensated £10 Amazon vouchers for the 30 mins it takes to complete the study. After removing responses that were unfinished or submitted too quickly to fully listen to the music, there are 25 responses under analysis.

Participants’ mean age is 25 years old, and their mean years of formal musical training is 10 years. Over 90% of participants listen to music daily, and 80% of participants play music/sing at least weekly.

5.2.2 Stimuli

For the listening study, 15 groups of stimuli were picked randomly from POP909-TVar generated outputs, and 15

⁴ We follow the consensual assessment technique [49,64] to design our study, which requires participants be experienced in the relevant domain.

from VGMIDI-TVar outputs. In each group, there are one theme and five variations, in which one is composed by a human, and the other four are generated by the models (VaTr, MuTr, FaTr, and VaMa). Each music excerpt is about 30-sec rendered using a piano sound. Each participant listens to 3 groups of music.

5.2.3 Procedure

After informed consent, instructions, and TVar examples, participants listen to a theme and then each variation, rating the musical dimensions of *variation success*, *stylistic consistency*, *similarity*, *creativity*, and *musical quality* on a 1–7 Likert scale, as well as two additional questions – *willingness to use a system that generates this variation (willingness)*, and *the extent to which this variation sounds like it is composed by a human (is human)*.⁵ An optional free text box for any comments follows the rating scales. After completing the evaluation of all 3 groups of materials, the extent to which the participant finds an algorithmic variation generation tool useful for their creative practice is rated (same scale), and a final optional free text box for any comments is provided. Given the participant and stimulus numbers, each TVar stimulus group was heard by approximately 3 participants, and all presentation orders were randomized to mitigate ordering and fatigue effects.

5.2.4 Results

Participants’ ratings for the features mentioned in Section 5.2.2 are shown as violin plots in Figure 2. For the BFA addressing our hypotheses at the top of Section 5.2, results for Hyp. 1 demonstrate that VaTr outperforms all three other algorithms (MuTr, FaTr, and VaMa) on *variation success ratings*, and MuTr outperforms FaTr and VaMa. But there is no difference between FaTr and VaMa. Results for Hyp. 2 indicate that TTMs (VaTr, MuTr, and FaTr) perform better than VaMa on *musical quality ratings*.

In terms of observations of results not tied to particular hypotheses, human-composed variations (Hu) appear to outperform algorithms on all metrics. In addition to *variation success* mentioned above, VaTr achieves higher ratings than other algorithms for *willingness* on both POP909-TVar and VGMIDI-TVar. The TTMs have higher ratings for *stylistic consistency*, *musical quality*, and *is human* than VaMa, but VaMa shows potential for generating creative variations. For POP909-TVar, VaMa and VaTr receive similar *creativity*, which is higher than that of MuTr and FaTr. For VGMIDI-TVar, although VaMa gets lower *creativity* than VaTr, it is still on par with MuTr and FaTr.

Approximately 100 comments are provided explaining the reasons for ratings, from which we find that participants usually consider the success of a variation according to the musical dimensions of pitch, rhythm, structure, dynamics, key signature, and texture, as well as the four more holistic dimensions mentioned in Section 5.2.2 (*stylistic consistency*, *similarity*, *musical quality*, and *is human*). As

⁵ The *variation success* is mainly to address **RQ1** and **RQ2**. Following existing research [37, 40, 65], we also include other music dimension metrics and *is human*. The *willingness* metric is to address **RQ3**.

such, deviations in these musical dimensions (e.g., dissonance, discordant dynamics, confusing structure) during the generation process could lead to unsatisfactory results.

Usually, a lack of stylistic consistency or being too similar/different to the theme will also result in an unsuccessful variation, but sometimes slight alterations (P11) or varying a lot from the theme (P21) can still lead to high ratings.

When considering whether a variation is written by a human composer or generated by AI, participants usually evaluate it in terms of the musical dimensions of rhythmic repetition, and appearance of dissonance, as well as overall musical quality. Lower-quality music seems to be associated with thoughts of being created by machines. But sometimes, even if the variation is recognised as AI-generated, participants are still receptive to it if the creativity and/or quality of the variation is good (P14 and P21).

The distribution of the extent to which participants find an algorithmic variation generation tool useful for their creative practice is: lower quartile = 3, median = 4, upper quartile = 5 on a 1–7 Likert scale. Corresponding comments comprise the following categories: i) benefit of music variation generation AI (MVG-AI) [18, 19, 66], with 8 out of 25 participants mentioning MVG-AI could be beneficial especially for inspiration; ii) concerns about MVG-AI [18, 19, 66], such as the quality and consistency not being sufficient to replace human composers (P7); iii) the clash between “creative ego” and MVG-AI [19], where for example P1 considers composing as creating art that is meaningful to the individual, which should not be done by AI instead. Similarly, P9 and P15 demonstrate wariness of the implications of AI and reluctance to use generative AI [18]; iv) further support/functionality required [18], such as P14 expecting MVG-AI to be able to produce variations that reflect a composers’ own style, and P24 thinking composers may have extra requirements for the MVG-AI in terms of emotional or style targets.

5.3 Feature-based evaluation

We use the whole pool of evaluation materials here, in which ten variations were generated by each algorithm for each theme drawn from the test sets. Three musical features are extracted and evaluated at the measure level:

Similarity score (SS) [67] gives the similarity between each measure of the generated variation and the corresponding measure of the theme in terms of pitch and rhythm.

Translational coefficient consistency (TC) [68] estimates the complexity or music-repetitive structure of an excerpt. A lower TC value means a music excerpt is highly repetitive, and vice versa. Here we calculate the absolute difference between the TC of each measure of the generated variation and the theme.

Key signature consistency (KSC) [69] captures the percentage of measures of the generated variation that have the same estimated key as the theme.

The evaluation results are shown in Table 1. We found that VaMa has the highest *SS* and *KSC* for both datasets. Among the TTMs, VaTr has higher values than MuTr and

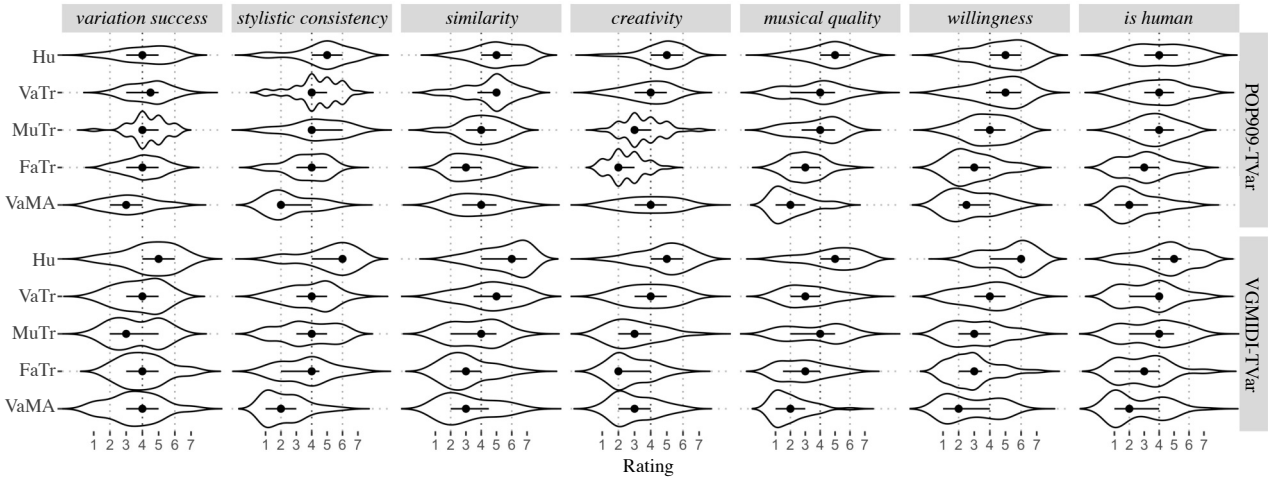


Figure 2. Violin plots for rating (1–7) distributions of seven dimensions on two datasets, where the envelope represents the distribution of responses; the lines indicate rating scales; the horizontal line goes from the lower quartile through the median (point) to the upper quartile.

POP909-TVar					
Feat.	Hu	VaTr	MuTr	FaTr	VaMa
SS	26.1 (20.6)	19.0 (18.0)	12.0 (15.2)	6.8 (6.6)	15.8 (9.5)
TC	0.10 (0.09)	0.11 (0.10)	0.12 (0.10)	0.12 (0.09)	0.15 (0.11)
KSC	51.7	41.0	29.0	22.4	52.6
VGMIDI-TVar					
Feat.	Hu	VaTr	MuTr	FaTr	VaMa
SS	25.2 (19.1)	9.7 (16.7)	7.2 (12.9)	4.2 (8.0)	17.4 (14.6)
TC	0.12 (0.13)	0.16 (0.15)	0.17 (0.15)	0.14 (0.13)	0.14 (0.13)
KSC	32.8	22.4	19.7	19.5	52.0

Table 1. The feature evaluation results, with mean and standard deviation (in brackets) for each feature.

FaTr on all three metrics for POP909-TVar, and higher than MuTr and FaTr on SS and KSC for VGMIDI-TVar. But, VaMa is outperformed by TMMs in most of metrics in the listening study (Section 5.2.4), reflecting that feature-based metrics alone cannot evaluate the performance of models from the human-aesthetic perception of music [70].

6. DISCUSSION

In this paper, we propose datasets and models for symbolic variation generation. To address our research questions, we run a listening study and feature-based evaluation for both deep and non-deep learning models, as most recent music generation research only compares deep learning approaches. According to our listening study results, human-composed variations outperform algorithms on all metrics, indicating that there is still a gap between human-composed variations and those generated by our proposed algorithms (RQ1). One of our proposed models (VaTr) is the strongest for variation generation, which demonstrates the superiority of a deep learning over a non-deep learning approach when the task is as specific as “generate a successful variation of this theme”. But our experiment results also show that not all deep learning approaches outperform the non-deep learning approach, especially in creativity (RQ2). And so for the less specific task of “gen-

erate music in a target style”, more research and comparative evaluation is required to establish the superiority of deep learning over alternative music generation methods. We hope that our study encourages researchers to revisit non-deep learning approaches, as well as to test experimentally whether deep learning methods are broadly superior to non-deep learning methods for music-generative tasks. To address RQ3, we further explore the extent to which participants in our listening study find MVG-AI useful for their creative practice, with an average rating of 4 on a 1–7 Likert scale, and some of the comments suggest that MVG-AI could lead to powerful tools for inspiration. One of our proposed models VaTr achieves the higher ratings for *willingness* than other models, and a comparable rating for *willingness* as that for human-composed variations on POP909-TVar (Figure 2).

Although the results are promising, there is still plenty of work to do in order to bridge technology and musical creativity. To increase the willingness of users to adopt MVG-AI, it is necessary to improve the quality of music generation and to consider the expectations of users. For example, to mitigate deviations in musical dimensions like dissonance, which lead to unsatisfactory results, adding a post-processing stage could be useful. Some participants mentioned their expectations about personalized AI in our study as well, as in [18, 19]. Using low-rank adaptation techniques [71, 72] to fine-tune a pre-trained model could be a strategy to explore in future. Another topic for future work entails further investigation of the quality of the provided datasets, to validate the reliability of the extracted theme-variation pairs.

Future applications of this work include: being integrated into AI music making systems to enable these systems to generate music with a stronger relationship to the user’s music prompt; being used in video game music domain, either as a tool to provide inspirations for composers, or for in-game generation to reduce listener fatigue [73, 74]; and structured music generation [12, 24, 75].

7. ETHICS STATEMENT

The listening study in this paper is approved by the Ethics Committee of The School of Arts and Creative Technology, University of York. A Participant Consent Form and a Participant Project Information Sheet is included prior to the start of the questionnaire to inform participants of the project and obtain their consent. Participants have the right to withdraw at any time. Each participant's data is protected by anonymization. The data collected involves ratings and comments as described in Section 5.2.2. The demographic information collected only involves participants' age in years, years of formal music training, regularity of playing music or signing, and regularity of listening to music, which are not sufficiently detailed for participants to be identified. No other identifying data are collected. Researchers shuffle the order of their responses, and then record these responses and use anonymized new IDs, which are person 1, person 2, etc. This way, even the researchers will not be able to identify the person after the survey.

Previous work demonstrates that some deep learning approaches that generate music from scratch tend to copy large sections from the training set with a high risk of copyright infringement [52]. In order to mitigate this issue, our models vary the input prompt. Moreover, future work includes further experiments regarding originality of the generation results. Although the training materials come from open-source datasets (POP909 [22], and VGMIDI [23]), it does not mean all the contents are copyright free. There is a possibility of our models to output copyrighted music. Therefore, our models and data are used for academic research only, not for commercially usages.

8. ACKNOWLEDGEMENT

The Viking cluster was used during this project, which is a high performance compute facility provided by the University of York. We are grateful for computational support from the University of York, IT Services and the Research IT team. Chenyu Gao is a PhD student supported jointly by the China Scholarship Council and the University of York.

9. REFERENCES

- [1] E. Sisman, "Variations. The New Grove Dictionary of Music and Musicians, edited by Stanley Sadie and John Tyrrell," 2001.
- [2] "AIVA," 2016. [Online]. Available: <https://www.aiva.ai/>
- [3] A. Roberts, J. Engel, Y. Mann, J. Gillick, C. Kayacik, S. Nørly, M. Dinculescu, C. Radebaugh, C. Hawthorne, and D. Eck, "Magenta studio: Augmenting creativity with deep learning in ableton live," in *Proceedings of the 7th International Workshop on Musical Metacreation*, 2019.
- [4] "Stable Audio," 2023. [Online]. Available: <https://www.stableaudio.com/>
- [5] "Suno," 2023. [Online]. Available: <https://www.suno.ai/>
- [6] "Staccato," 2023. [Online]. Available: <https://staccato.ai/>
- [7] R. B. Tchemeube, J. Ens, C. Plut, P. Pasquier, M. Safi, Y. Grabit, and J.-B. Rolland, "Evaluating human-AI interaction via usability, user experience and acceptance measures for MMM-C: A creative AI system for music composition," in *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, 2023, pp. 5769–5778.
- [8] M. E. Malandro, "Composer's Assistant: Interactive transformers for multi-track MIDI infilling," in *Proceedings of the 24th International Society for Music Information Retrieval Conference*, 2023.
- [9] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Défossez, "Simple and controllable music generation," *Advances in Neural Information Processing Systems*, vol. 36, 2024.
- [10] K. Ishizuka and T. Onisawa, "Generation of variations on theme music based on impressions of story scenes considering human's feeling of music and stories," *International Journal of Computer Games Technology*, vol. 2008, 2008.
- [11] V. Arutyunov and A. Averkin, "Genetic algorithms for music variation on genom platform," *Procedia computer science*, vol. 120, pp. 317–324, 2017.
- [12] F. Pachet, A. Papadopoulos, and P. Roy, "Sampling variations of sequences for structured music generation." in *Proceedings of 18th International Conference on Music Information Retrieval*, 2017, pp. 167–173.
- [13] E. Row, J. Tang, and G. Fazekas, "JAZZVAR: A dataset of variations found within solo piano performances of jazz standards for music overpainting," in *Proceedings of the 16th International Symposium on Computer Music Multidisciplinary Research*, 2023.
- [14] B. Banar, N. Bryan-Kinns, and S. Colton, "A tool for generating controllable variations of musical themes using variational autoencoders with latent space regularisation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 13, 2023, pp. 16401–16403.
- [15] J. Ens and P. Pasquier, "MMM: Exploring conditional multi-track music generation with the transformer," *arXiv preprint arXiv:2008.06048*, 2020.
- [16] R. Guo, I. Simpson, C. Kiefer, T. Magnusson, and D. Herremans, "MusIAC: An extensible generative framework for music infilling applications with multi-level control," in *International Conference on Computational Intelligence in Music, Sound, Art and Design (Part of EvoStar)*. Springer, 2022, pp. 341–356.

- [17] W. E. Caplin, *Classical form: A theory of formal functions for the instrumental music of Haydn, Mozart, and Beethoven*. Oxford University Press, 1998.
- [18] M. Newman, L. Morris, and J. H. Lee, “Human-AI music creation: Understanding the perceptions and experiences of music creators for ethical and productive collaboration,” in *Proceedings of 24th International Conference on Music Information Retrieval*, 2023.
- [19] K. Worrall and T. Collins, “Considerations and concerns of professional game composers regarding artificially intelligent music technology,” *IEEE Transactions on Games*, 2023.
- [20] J. Devaney, C. Arthur, N. Condit-Schultz, and K. Nisula, “Theme and variation encodings with roman numerals (TAVERN): A new data set for symbolic music analysis,” in *Proceedings of 16th International Conference on Music Information Retrieval*, 2015.
- [21] B. L. Sturm, J. F. Santos, O. Ben-Tal, and I. Korshunova, “Music transcription modelling and composition using deep learning,” in *1st Conference on Computer Simulation of Musical Creativity*, 2016.
- [22] Z. Wang, K. Chen, J. Jiang, Y. Zhang, M. Xu, S. Dai, G. Bin, and G. Xia, “POP909: A pop-song dataset for music arrangement generation,” in *Proceedings of 21st International Conference on Music Information Retrieval*, 2020.
- [23] L. N. Ferreira and J. Whitehead, “Learning to generate music with sentiment,” in *Proceedings of 22st International Conference on Music Information Retrieval*, 2021.
- [24] Y.-J. Shih, S.-L. Wu, F. Zalkow, M. Muller, and Y.-H. Yang, “Theme transformer: Symbolic music generation with theme-conditioned transformer,” *IEEE Transactions on Multimedia*, 2022.
- [25] T. Collins, R. Laney, A. Willis, and P. H. Garthwaite, “Developing and evaluating computational models of musical style,” *AI EDAM*, vol. 30, no. 1, pp. 16–43, 2016.
- [26] T. Collins and R. Laney, “Computer-generated stylistic compositions with long-term repetitive and phrasal structure,” *Journal of Creative Music Systems*, vol. 1, no. 2, 2017.
- [27] A. Eigenfeldt and P. Pasquier, “Realtime generation of harmonic progressions using controlled markov selection,” in *Proceedings of ICCX-Computational Creativity Conference*, 2010, pp. 16–25.
- [28] F. Pachet, P. Roy, and G. Barbieri, “Finite-length markov processes with constraints,” in *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.
- [29] Z. Yin, F. Reuben, S. Stepney, and T. Collins, “Deep learning’s shallow gains: A comparative evaluation of algorithms for automatic music generation,” *Machine Learning*, vol. 112, no. 5, pp. 1785–1822, 2023.
- [30] C.-Z. A. Huang, A. Vaswani, J. Uszkoreit, N. Shazeer, I. Simon, C. Hawthorne, A. M. Dai, M. D. Hoffman, M. Dinculescu, and D. Eck, “Music transformer: Generating music with long-term structure,” in *International Conference on Learning Representations*, 2019.
- [31] A. Roberts, J. Engel, C. Raffel, C. Hawthorne, and D. Eck, “A hierarchical latent vector model for learning long-term structure in music,” in *International Conference on Machine Learning*, 2018, pp. 4364–4373.
- [32] C. Payne, “Musenet,” 2019. [Online]. Available: <https://openai.com/research/musenet>
- [33] N. Zhang, “Learning adversarial transformer for symbolic music generation,” *IEEE Transactions on Neural Networks and Learning Systems*, 2020.
- [34] J. Jiang, G. G. Xia, D. B. Carlton, C. N. Anderson, and R. H. Miyakawa, “Transformer VAE: A hierarchical model for structure-aware and interpretable music representation learning,” in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2020, pp. 516–520.
- [35] H. Zhang, L. Xie, and K. Qi, “Implement music generation with GAN: A systematic review,” in *2021 International Conference on Computer Engineering and Application*, 2021, pp. 352–355.
- [36] H.-T. Hung, J. Ching, S. Doh, N. Kim, J. Nam, and Y.-H. Yang, “EMOPIA: A multi-modal pop piano dataset for emotion recognition and emotion-based music generation,” in *Proceedings of 22nd International Conference on Music Information Retrieval*, 2021.
- [37] L. N. Ferreira, L. Mou, J. Whitehead, and L. H. Lelis, “Controlling perceived emotion in symbolic music generation with monte carlo tree search,” in *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, vol. 18, no. 1, 2022, pp. 163–170.
- [38] S.-L. Wu and Y.-H. Yang, “MuseMorphose: Full-song and fine-grained piano music style transfer with one transformer VAE,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 1953–1967, 2023.
- [39] G. Mittal, J. Engel, C. Hawthorne, and I. Simon, “Symbolic music generation with diffusion models,” in *Proceedings of the 22nd International Society for Music Information Retrieval Conference*, 2021.
- [40] L. Min, J. Jiang, G. Xia, and J. Zhao, “Polyffusion: A diffusion model for polyphonic score generation with internal and external controls,” in *Proceedings of 24th International Conference on Music Information Retrieval*, 2023.

- [41] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” *Advances in Neural Information Processing Systems*, vol. 27, 2014.
- [42] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” in *International Conference on Learning Representations*, 2014.
- [43] Z. Wang, Y. Zhang, Y. Zhang, J. Jiang, R. Yang, J. Zhao, and G. Xia, “Pianotree VAE: Structured representation learning for polyphonic music,” in *Proceedings of 21st International Conference on Music Information Retrieval*, 2020.
- [44] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” *Advances in Neural Information Processing Systems*, vol. 30, 2017.
- [45] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, “Deep unsupervised learning using nonequilibrium thermodynamics,” in *International Conference on Machine Learning*, 2015, pp. 2256–2265.
- [46] J. Ho, A. Jain, and P. Abbeel, “Denoising diffusion probabilistic models,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 6840–6851, 2020.
- [47] P. Doornbusch, “Algorithmic composition: Paradigms of automated music generation,” *Computer Music Journal*, vol. 34, no. 3, pp. 70–74, 2010.
- [48] D. Cope, *Computer models of musical creativity*. Cambridge: MIT Press Cambridge, US, 2005.
- [49] M. T. Pearce and G. A. Wiggins, “Evaluating cognitive models of musical composition,” in *Proceedings of the 4th International Joint Workshop on Computational Creativity*, 2007, pp. 73–80.
- [50] S. Dai, H. Zhang, and R. B. Dannenberg, “Automatic analysis and influence of hierarchical structure on melody, rhythm and harmony in popular music,” in *Proceedings of the Joint Conference on AI Music Creativity*, 2020.
- [51] A. Arzt, S. Böck, and G. Widmer, “Fast identification of piece and score position via symbolic fingerprinting,” in *Proceedings of 13rd International Conference on Music Information Retrieval*, 2012, pp. 433–438.
- [52] Z. Yin, F. Reuben, S. Stepney, and T. Collins, “Measuring when a music generation algorithm copies too much: The originality report, cardinality score, and symbolic fingerprinting by geometric hashing,” *SN Computer Science*, vol. 3, no. 5, p. 340, 2022.
- [53] T. Collins, A. Arzt, S. Flossmann, and G. Widmer, “SIARCT-CFP: Improving precision and the discovery of inexact musical patterns in point-set representations,” in *Proceedings of 14th International Conference on Music Information Retrieval*, 2013, pp. 549–554.
- [54] Y.-S. Huang and Y.-H. Yang, “Pop music transformer: Beat-based modeling and generation of expressive pop piano compositions,” in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 1180–1188.
- [55] P. Shaw, J. Uszkoreit, and A. Vaswani, “Self-attention with relative position representations,” in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, vol. 2, 2018, pp. 464–468.
- [56] A. Huang, M. Dinculescu, A. Vaswani, and D. Eck, “Visualizing music self-attention,” in *Proceedings of NeurIPS Workshop on Interpretability and Robustness in Audio, Speech, and Language*, vol. 1, 2018, p. 4.
- [57] R. Batlle-Roca, E. Gómez, W. Liao, X. Serra, and Y. Mitsufuji, “Transparency in music-generative AI: A systematic literature review,” 2023.
- [58] T. Collins, R. Laney, A. Willis, and P. H. Garthwaite, “Chopin, mazurkas and Markov: Making music in style with statistics,” *Significance*, vol. 8, no. 4, pp. 154–159, 2011.
- [59] E. W. Dijkstra, “A note on two problems in connexion with graphs,” *Numerische Mathematik*, vol. 50, pp. 269–271, 1959.
- [60] A. Katharopoulos, A. Vyas, N. Pappas, and F. Fleuret, “Transformers are RNNs: Fast autoregressive transformers with linear attention,” in *Proceedings of the International Conference on Machine Learning*, 2020.
- [61] A. Vyas, A. Katharopoulos, and F. Fleuret, “Fast transformers with clustered attention,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 21 665–21 674, 2020.
- [62] Z. Dienes, “Using Bayes to get the most out of non-significant results,” *Frontiers in psychology*, vol. 5, p. 85883, 2014.
- [63] M. D. Lee and E.-J. Wagenmakers, *Bayesian cognitive modeling: A practical course*. Cambridge University Press, 2014.
- [64] T. M. Amabile, *Creativity in context*. Westview Press, Boulder, Colorado, 1996.
- [65] S. Ji, X. Yang, and J. Luo, “A survey on deep learning for symbolic music generation: Representations, algorithms, evaluations, and challenges,” *ACM Computing Surveys*, vol. 56, no. 1, pp. 1–39, 2023.
- [66] R. Louie, A. Coenen, C. Z. Huang, M. Terry, and C. J. Cai, “Novice-AI music co-creation via AI-steering tools for deep generative models,” in *Proceedings of the 2020 CHI conference on human factors in computing systems*, 2020, pp. 1–13.

- [67] Z. Yin, “New evaluation methods for automatic music generation,” Ph.D. dissertation, University of York, 2022.
- [68] K. Foubert, T. Collins, and J. De Backer, “Impaired maintenance of interpersonal synchronization in musical improvisations of patients with borderline personality disorder,” *Frontiers in Psychology*, vol. 8, p. 537, 2017.
- [69] C. S. Sapp, “Visual hierarchical key analysis,” *Computers in Entertainment*, vol. 3, no. 4, pp. 1–19, 2005.
- [70] L.-C. Yang and A. Lerch, “On the evaluation of generative models in music,” *Neural Computing and Applications*, vol. 32, no. 9, pp. 4773–4784, 2020.
- [71] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, “LoRA: Low-rank adaptation of large language models,” in *International Conference on Learning Representations*, 2022.
- [72] R. Zhang, J. Han, C. Liu, A. Zhou, P. Lu, H. Li, P. Gao, and Y. Qiao, “LLaMA-Adapter: Efficient fine-tuning of large language models with zero-initialized attention,” in *International Conference on Learning Representations*, 2023.
- [73] K. Collins *et al.*, “An introduction to the participatory and non-linear aspects of video games audio,” *Essays on sound and vision*, pp. 263–298, 2007.
- [74] W. Phillips, *A composer’s guide to game music*. MIT Press, 2014.
- [75] S. Dai, H. Yu, and R. B. Dannenberg, “What is missing in deep music generation? A study of repetition and structure in popular music,” in *Proceedings of 23rd International Conference on Music Information Retrieval*, 2022.

AUTOMATIC DETECTION OF MORAL VALUES IN MUSIC LYRICS

Vjosa Preniqi¹ Iacopo Ghinassi¹ Julia Ive¹
Kyriaki Kalimeri² Charalampos Saitis¹

¹ Centre for Digital Music, Queen Mary University of London, London, UK

² ISI Foundation, Turin, Italy

{v.preniqi, i.ghinassi, j.ive, c.saitis}@qmul.ac.uk, kyriaki.kalimeri@isi.it

ABSTRACT

Moral values play a fundamental role in how we evaluate information, make decisions, and form judgements around important social issues. The possibility to extract morality rapidly from lyrics enables a deeper understanding of our music-listening behaviours. Building on the Moral Foundations Theory (MFT), we tasked a set of transformer-based language models (BERT) fine-tuned on 2,721 synthetic lyrics generated by a large language model (GPT-4) to detect moral values in 200 real music lyrics annotated by two experts. We evaluate their predictive capabilities against a series of baselines including out-of-domain (BERT fine-tuned on MFT-annotated social media texts) and zero-shot (GPT-4) classification. The proposed models yielded the best accuracy across experiments, with an average F1 weighted score of 0.8. This performance is, on average, 5% higher than out-of-domain and zero-shot models. When examining precision in binary classification, the proposed models perform on average 12% higher than the baselines. Our approach contributes to annotation-free and effective lyrics morality learning, and provides useful insights into the knowledge distillation of LLMs regarding moral expression in music, and the potential impact of these technologies on the creative industries and musical culture.

1. INTRODUCTION

Lyrics play a crucial role in how we experience music, affecting our emotions and actions. Positive lyrics can motivate and elevate listeners, whereas negative or aggressive content in songs may negatively impact mood and behaviour [1]. Social, political, and cultural issues, such as racial inequality and gender discrimination, are often reflected in the music lyrics of their time [2, 3]. Songs that feature in successful campaigns typically include uplifting melodies and lyrics that reflect the ideals of a nation, representing values of optimism and progress towards a better future [4]. Moral rhetoric in lyrics has been used to

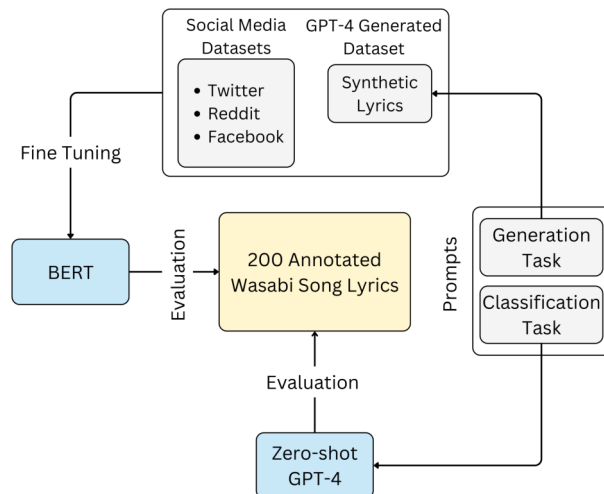


Figure 1. Model Structure for predicting Moral Foundations (MFT) in Lyrics, fine-tuned on out-of-domain social media data, and synthetically generated lyrics with GPT-4.

advocate for what is perceived to be a necessary societal change [5], promote peace and unity [6], and raise awareness for marginalised groups [7]. These narratives are closely related to moral judgements and beliefs, yet their relationship to music listening behaviors has received limited attention by music scientists.

In the field of Music Information Retrieval (MIR), lyrical content analysis has focused primarily on genre classification [8], mood prediction [9], emotion dynamics [10], and lyrics-to-audio alignment [11, 12]. Recent works have elaborated on less attended psychological characteristics of music lyrics, including moral valence. For example, insights into personal values and personality traits derived from lyrics can enhance various MIR tasks, including genre classification, audio tagging, and music recommendations [13]. Preniqi and colleagues [14] showed that moral valence extracted from lyrics can to some extent predict listeners’ moral values, in some cases more accurately than audio features. The possibility to extract morality rapidly from lyrics can enable a deeper understanding of our music listening behaviours.

Inferring moral values from song lyrics is a complex natural language processing (NLP) task from the start due to the subjectivity of our perceptions and interpretations. The progress is further hindered by the lack of an-



notated lyrics for training new or fine-tuning pre-trained models, and for benchmarking. Using models fine-tuned with out-of-domain annotated texts (e.g., from social media [15, 16]) to predict moral values in music lyrics faces significant challenges due to the unique structure of lyrics compared to other textual forms (e.g., greater use of repetition, metaphor, imagery, and other poetic devices).

In light of the above, we investigate the novel task of automatic detection of moral values in music lyrics using an integrated approach that leverages the strengths of two distinct NLP technologies. Specifically, we leverage the generative capabilities of GPT-4 (Generative Pre-trained Transformer) to create morally nuanced synthetic lyrics—a process required only once—and employ BERT (Bidirectional Encoder Representations from Transformers), which demands fewer computational resources, to learn from the synthetic data structure.

Following recent related work [14, 15, 17], we operationalize morality drawing on Haidt and Graham’s Moral Foundations Theory [18], which outlines five core moral traits, or foundations, divided into “virtue” and “vice” based on moral polarity: *Care* and *Harm*, *Fairness* and *Cheating*, *Loyalty* and *Betrayal*, *Authority* and *Subversion*, *Purity* and *Degradation*. We developed a corresponding set of 10 single-label classification models, each customized to predict the presence or absence of one moral value in lyrical text. MFT is a straightforward yet comprehensive model for understanding moral values, uniquely characterized by well-developed term dictionaries [19].

We present a dataset of 200 real song lyrics human-annotated with MFT. To the best of our knowledge, this is the first such dataset. It serves as the basis for evaluating our proposed method. We make the real and synthetic lyrics datasets, and the paper code fully available via a GitHub repository.¹

We report a comprehensive comparison of the proposed models against BERT fine-tuned with out-of-domain human-annotated moral text data and zero-shot classification with GPT-4. Figure 1 summarizes the overall pipeline of this work. The proposed models yielded the best accuracy across experiments, with an average F1 weighted score of 0.8. This performance is, on average, 5% higher than out-of-domain and zero-shot models. When examining precision in binary classification, the proposed models perform on average 12% higher than the baselines. Our approach contributes to annotation-free lyrics morality learning, and provides useful insights into the knowledge distillation of large language models such as GPT-4 regarding moral expression in music.

2. RELATED WORK

The field of music and moral expression has received limited attention. However, recent studies have shown a link between an individual’s moral values and their preferences for lyrics and music, suggesting significant implications for tailoring personalisation in streaming services

[14, 17, 20]. Further research has delved into how moral values and lyrical preferences manifest within specific music communities. For example, Messick and Aranda [21] demonstrated that moral values could explain a unique and significant portion of the variance in lyrical preferences among fans of different metal music sub-genres.

Given the understanding that verbal expressions more effectively convey morality than non-verbal forms [17, 22], initial studies introduced lexicons [23, 24] as an extension of Moral Foundations Dictionary (MFD) [25] for identifying words and lemmas that accurately depict moral foundations. More recent studies focused on examining moral values in texts using human-annotated social media datasets [26–28], and introducing more advanced Natural Language Processing (NLP) approaches to detect moral dimensions in textual content [15, 16]. Trager et al. [27] introduced baseline models for predicting moral values, employing a pre-trained BERT model fine-tuned on the Moral Foundation Reddit Corpus. Guo et al. [16] proposed a multi-label model for predicting moral values with Twitter and news data, incorporating the domain adversarial training framework suggested by Ganin et al. [29] to align multiple datasets and generalise for out-of-domain predictions. A similar approach was taken by Preniqi et al. [15] in predicting moral values in different social media domains.

However, a main challenge that persists is the ability of these models to generalise across various domains. Lisco and colleagues [30] demonstrated that text classifiers perform better when domains are similar. This poses a major obstacle when predicting morality in lyrics because there is no prior study that has presented an annotated lyrics dataset with moral values. Further, manually annotating extensive text demands substantial time, resources, and deep understanding of Moral Foundations Theory (MFT).

To overcome these limitations, we employ GPT-4, an advanced LLM, to generate lyrics infused with various moral undertones, which helps in fine-tuning a moral classifier. This minimises the need for laborious manual annotation of extensive lyric databases, enabling us to utilise a smaller, human-annotated dataset to validate the effectiveness of knowledge distilled from GPT-4. The capacities of LLMs for music tasks are being actively explored for the moment. Doh et al. [31] similarly employed a large language model such as GPT-3 for generating pseudo captions from tags to mitigate the problem of data scarcity in the field of automatic music captioning. While Zhang et al [32] evaluated the quality and correctness of generated music lyrics via GPT-3. Sawicki et al. [33] investigated the possibility of using GPT-3 models to generate high-quality poems in a specific author’s style while suggesting that GPT-3 can be a useful tool in assisting authors.

3. METHOD

3.1 Human-Annotated Lyrics

For this work, we annotated 200 song lyrics, categorising them into 10 different moral foundations. This annotation process was conducted by two skilled annotators: the

¹ <https://github.com/vjosapreniqi/ismir-mft-values>

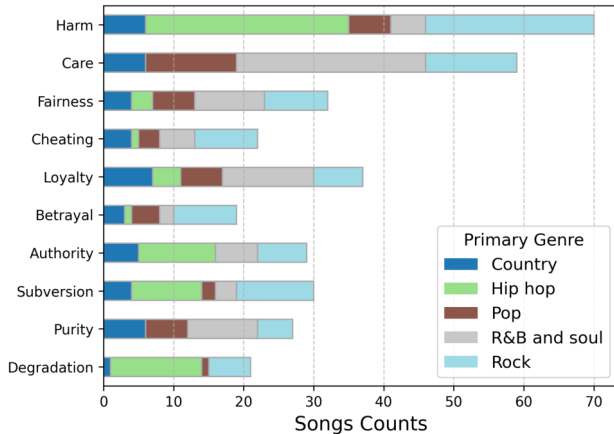


Figure 2. Distribution of Moral Foundations in 200 song lyrics dataset annotated by human annotators with genre proportions for each moral foundation.

lead author of this study and an external researcher with a background in music and sound design, both of whom agreed to contribute. Before starting, the annotators were informed about their participation rights, including the option to discontinue their involvement at any point. Each annotator was assigned with 125 songs for annotation. To evaluate the agreement between annotators, 50 songs were annotated by both annotators. The inner-annotator agreement was assessed using Cohen’s kappa coefficient for each moral label. This resulted in an almost perfect agreement [34] with an average score of 0.86 across all moral categories identified within the lyrics of the chosen songs. We selected the songs for the moral values annotation from the Wasabi Dataset [35], known for its extensive collection of 2 million songs including lyrics, artist gender, and musical genre among other data. This dataset spans over five decades, enabling the selection of songs from various eras. The process of selecting the songs involved a semi-random approach, with efforts made to retain the distribution of genres, and the timeline of song releases as found in the original dataset. Among the 200 songs annotated for moral values, 18 were from the 60s and 70s, 78 from the 80s and 90s, and 116 from the post-2000 era. The chosen songs represented a balanced mix of genres including Rock, Pop, Hip-Hop, R&B, Soul, and Country. Figure 2 depict the distribution of Moral values in the human-annotated song lyrics with the proportion of genre for each MFT value.

3.2 Predicting Morality in Lyrics with Domain Adaptation

Initially, we tried to predict moral values in lyrics by fine-tuning a BERT model with out-of-domain social media data, following the approach used by Preniqi et al. [15]. We utilised 20,628 tweets from the Moral Foundation Twitter Corpus (MFTC) [26]; 13,995 posts from the Moral Foundations Reddit Corpus (MFRC) [27]; 1,510 posts from Facebook vaccination dataset [28]. Preniqi’s and other work have demonstrated that predicting moral values using a single-label approach—predicting one MFT value at

a time—results in higher accuracy [15, 27]. Informed by these findings, we developed a set of single-label classification models tailored to predict individual moral foundations in lyrics.

As a baseline model, we apply a similar approach to the MoralBERT [15]. We identify the polarities (virtues and vices) of moral foundations, as opposed to just identifying the mere presence or absence of moral values. We incorporate the domain adversarial method aiming to improve the models’ ability to generalise effectively in predicting moral values in lyrics [15, 16]. Adopting this model, we start by deriving a domain invariant representation h from the BERT CLS embedding e :

$$h = W_{inv}e$$

where $W_{inv} \in \mathcal{R}^{768 \times 768}$ is a learnable matrix. Next, we calculate moral values predictions \hat{y}_m using:

$$\hat{y}_m = \text{Softmax}(W_1(\text{ReLU}(W_2h)))$$

with $W_1 \in \mathcal{R}^{768 \times 768}$, $W_2 \in \mathcal{R}^{768 \times c}$ representing 2 learnable matrices, c being the number of classes, ReLU is the rectified linear unit activation function and Softmax is the normalised exponential function. A domain classification head is also included for obtaining domain predictions \hat{y}_d :

$$\hat{y}_d = \text{Softmax}(W_3(\text{ReLU}(W_4h)))$$

with $W_3 \in \mathcal{R}^{768 \times 768}$, and $W_4 \in \mathcal{R}^{768 \times d}$ learnable matrices and with d being the number of domains in the training set. The main rationale of the adversarial network is increasing the loss from the domain head while minimising the loss from the moral values prediction. Hence, the model is “forced” to learn domain-invariant representations. This is achieved by integrating a gradient reversal layer before the domain classification head, while using standard training for minimising moral prediction loss. Cross-entropy (CE) loss is used for both the moral and domain classification heads. The final loss is expressed as:

$$L = CE(\hat{y}_m, Y_m) - CE(\hat{y}_d, Y_d) + L_{norm} + L_{rec}$$

with Y_m and Y_d as the ground truth for moral values and domain, respectively. Two regularisation terms from [16] are added: L2 norm regularisation and reconstruction loss:

$$L_{norm} = \|W_{inv}h - I\|^2, \quad L_{rec} = \|W_{rec}h - e\|^2$$

similar to W_{inv} (defined above), $W_{rec} \in \mathcal{R}^{768 \times 768}$ is also a learnable matrix and I is the identity matrix. These regularization losses are combined with moral and domain classification losses. The regularization terms are not applied when training MoralBERT on a single domain (e.g., when trained on just synthetic lyrics).

The binary setting we use implies the model should learn from highly unbalanced datasets, where the neutral label (negative class) is far more represented than the single moral value to be predicted in each instance (positive

class). To address the class imbalance, we employed two methods. First, weights are assigned to classes [36]:

$$weight_c = \frac{N - N_c}{N}$$

where N is the total training samples and N_c is the count of samples per class c . Second, similar to [37], we employed a separate threshold θ_v for each moral value v , so that we use \hat{y}_m to obtain the final prediction \hat{m} :

$$\hat{m} = \begin{cases} 1 & \text{if } \hat{y}_m > \theta_v \\ 0 & \text{otherwise} \end{cases}$$

with $\hat{m} = 1$ indicating the moral value is present in the lyrics and $\hat{m} = 0$ indicating it is not. The optimal value θ_v for each moral value v was found by optimizing for binary F1 during training, searching in the search space 0.05 to 0.95 with a step of 0.05. The models were trained for 20 epochs using a single Nvidia T4 GPU, a learning rate of $5e-5$, and the Adam optimiser for all MoralBERT experiments.

3.3 Synthetic Lyrics Generation for Moral Assessment

There is a growing interest in knowledge distillation from large pre-trained language models via synthetic text generation [38]. Here we apply a similar knowledge distillation approach by utilising GPT-4 for synthetic lyrics generation. This method eliminates the need to collect real-life data, which is often difficult to gather for a specific NLP task and with a specific input distribution [39]. Initially, we assessed GPT-4’s familiarity with Moral Foundation Theory [25], confirming its fundamental understanding of moral values. We tasked GPT-4 with generating lyrics by formulating a prompt, as follows:

Prompt: *You are an assistant to a songwriter; you need to assist in writing lyrics related to the Moral foundations described in the Moral Foundation Theory. Given the {Moral Foundations Tags}, which represent {Description Tags}, write original lyrics of a song expressing these moral foundations. DO NOT directly mention these moral foundations. DO NOT explicitly talk about morality. Write it in the style of {Artist Tags}.*

We assigned a “role” (songwriter assistant) for the model and provided three types of “input tags”. The {Moral Foundations Tags} comprise any of the 10 moral values. The resulting lyrics can represent 1, 2, or 3 moral values. We determined this based on the moral combinations observed in our human-annotated lyrics dataset. The {Description Tags} represent fundamental concepts of each moral value. The {Artist Tags} represent the names of artists whose styles we employ to diversify the lyrics. Initially, we intended to commence the lyrics generation task solely using moral categories and genres as tags. However, we observed that the lyrics were more uniform and generic compared to when we incorporated the artist’s style. To tailor the lyrical style using various artists, we employed MusicOSet [40], a collection of musical elements (e.g., music, albums, artists, genres and

popularity) suitable for music data mining. To capture the nuances of different genres, we organized the artists according to their popularity and grouped them into prevalent genres like Rock, Pop, Country, Hip Hop, R&B, Soul, Folk, Blues, and Jazz. These genres align very closely with those in the song lyrics we selected for human annotations. We chose to utilise this dataset because it offers detailed data on artist genres and sub-genres, as well as an artist popularity metric that we employ in developing lyric styles. We acquired a dataset comprising 2,721 artificially generated lyrics, each aligned with moral categories similar to our human-annotated lyrics dataset. On average, the generated lyrics had 146 words, with a total of 10,305 unique words across the synthetic lyrics dataset.

3.4 GPT-4 in Moral Classification Task

In addition, we wanted to assess the capability of the 0-shot GPT-4 model in classifying morality in actual song lyrics while comparing it to our proposed model. To do so, we prompted the task as follows:

Prompt: *You will be provided with song lyrics. The song lyrics will be delimited with #### characters. Classify each lyric into 10 Possible Moral Foundations as defined in Moral Foundation Theory The available Moral Foundations are: {Moral Foundations Tags}. The explanation of the moral foundations is as follows: {Description Tags}. This is a multi-label classification problem: where it’s possible to assign one or multiple categories simultaneously. Report the results in JSON format such that the keys of the correct moral values are reported in a list.*

The song lyrics utilised for the GPT-4 model classification are the same as the ones annotated by human annotators. In this way, we can compare the human annotations with those of the model while assessing the general performance of GPT-4 for the classification task.

4. EXPERIMENTS

We started by analysing the MoralBERT technique [20] and fine-tuned models using social media data from Twitter, Reddit, and Facebook. The total number of text records was 35,887. We found that 51% of the texts were neutral and 49% of them were labeled with one or more moral values. This indicated a significant skew towards neutral texts, which we addressed by adding the class weighting technique. After that, we evaluated the BERT models fine-tuned with only GPT-4 generated lyrics. We call these models “BERT SL”. We also fine-tuned the models with a combination of out-of-domain social media data and the generated lyrics data which we call “MoralBERT SL”. We used the Domain Adversarial module only when fine-tuning BERT with multiple domain data, including synthetic lyrics. When fine-tuning solely with synthetic lyrics, this module was not utilized. Lastly, we evaluated GPT-4’s zero-shot classification capabilities against our models on the manually annotated song lyrics.

The results show that the models achieving the highest

	F1 Scores Weighted Average				F1 Scores Binary			
	MoralBERT	GPT-4	BERT SL	MoralBERT SL	MoralBERT	GPT-4	BERT SL	MoralBERT SL
Care	.80 ± .03	.68 ± .03	.81 ± .03	.83 ± .03	.68 ± .05	.64 ± .04	.68 ± .05	.75 ± .04
Harm	.68 ± .03	.75 ± .03	.71 ± .03	.70 ± .03	.62 ± .05	.71 ± .04	.63 ± .05	.69 ± .04
Fairness	.55 ± .03	.73 ± .03	.73 ± .03	.74 ± .03	.30 ± .05	.39 ± .06	.41 ± .06	.38 ± .06
Cheating	.84 ± .03	.80 ± .03	.86 ± .02	.69 ± .03	.27 ± .09	.16 ± .07	.52 ± .08	.32 ± .06
Loyalty	.69 ± .03	.67 ± .03	.77 ± .04	.79 ± .04	.38 ± .06	.34 ± .06	.21 ± .08	.27 ± .09
Betrayal	.81 ± .02	.72 ± .03	.89 ± .02	.84 ± .02	.34 ± .07	.31 ± .06	.40 ± .11	.37 ± .08
Authority	.77 ± .03	.75 ± .03	.77 ± .03	.84 ± .03	.45 ± .06	.42 ± .06	.35 ± .07	.39 ± .09
Subversion	.80 ± .03	.72 ± .03	.80 ± .03	.71 ± .03	.44 ± .07	.39 ± .06	.40 ± .07	.43 ± .06
Purity	.77 ± .03	.86 ± .02	.89 ± .02	.90 ± .02	.41 ± .06	.56 ± .07	.55 ± .08	.63 ± .08
Degradation	.74 ± .03	.81 ± .03	.81 ± .03	.86 ± .03	.34 ± .06	.40 ± .07	.30 ± .07	.32 ± .10
Average	.75 ± .03	.75 ± .03	.80 ± .03	.80 ± .03	.42 ± .06	.43 ± .06	.45 ± .07	.46 ± .07

Table 1. F1 scores of prediction models with standard deviation estimated via 1,000 bootstraps. Weighted average scores account for both moral and non-moral (neutral) classes, while binary scores only for moral classes. SL = Synthetic Lyrics.

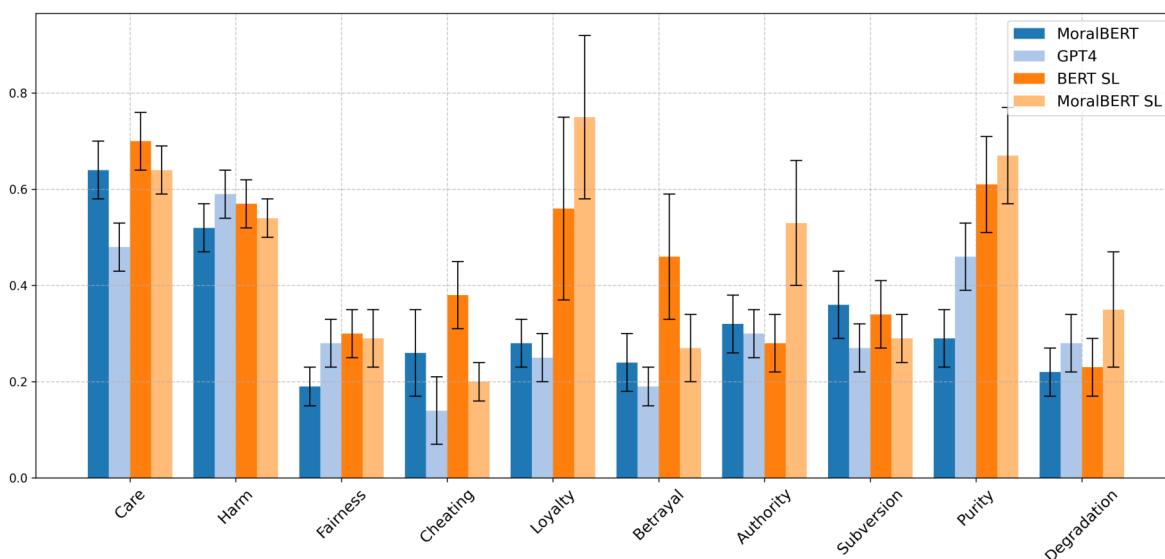


Figure 3. Precision scores for binary classification with standard deviation estimated via 1,000 bootstraps.

performance were BERT SL and MoralBERT SL. These models performed on average 5% better across all moral values in terms of F1 weighted score which accounts for both moral and non-moral prediction classes. While for the binary F1, these models were marginally better than GPT-4. For harm foundation, GPT-4 performed slightly better, possibly due to the synthetic lyrics’ lack of natural variability when expressing this foundation. The fact that MoralBERT SL and BERT SL performances are similar to the one from GPT-4 for binary F1 is expected as the same latent knowledge of GPT-4 has been distilled into BERT by using the generated lyrics. The improvements from MoralBERT SL and BERT SL are significant for what concerns weighted F1, suggesting that given the supervised setting of these models, they were also able to learn the higher prior probability of non-moral (e.g., neutral) instances, which generally outweigh moral instances. The same is evident if we look at Figure 3, which compares the binary Precision scores of the various models. From the figure, it is evident that MoralBERT SL and BERT SL exhibit significantly higher Precision surpassing GPT-4 and

MoralBERT by 12% on average. These models, then, are often correct when labelling lyrics with moral values (even though results vary according to which moral value), while being more cautious in assigning a moral value, given the preponderance of neutral cases. For the evaluation metrics, we report the standard deviation estimated via Bootstrapping which is a statistical resampling technique used to estimate the variability of the metrics. We used 1,000 bootstraps which is typically sufficient to achieve a reasonable approximation of the standard deviation.

Our findings show that BERT-based models are still comprehensible with larger models such as GPT-4, when fine-tuned properly they can excel in specified tasks. GPT-4 demonstrated a very good performance even without any fine-tuning (zero-shot approach) which was anticipated given its state-of-art performance in multiple tasks and its training on an extensive amount of data. These models have been trained on diverse text sources such as Wikipedia, GitHub, chat logs, books, and articles [41], enabling them to comprehend language across various domains [31]. The earlier model, GPT-3, contains 175 bil-

Song Name	Artist	Human Annotations	MoralBERT	GPT-4	BERT SL MoralBERT SL
“Take This Heart of Mine”	Foghat	Care, Purity	Care, Purity	Care, Loyalty	Care, Fairness, Purity
“Who’s Cheatin’ Who”	Charly McClain	Cheating, Betrayal	Cheating, Betrayal, Loyalty, Purity	Cheating, Betrayal	Cheating, Betrayal
“Samurai Showdown”	RZA	Harm, Authority	Harm, Betrayal, Authority, Purity	Harm, Loyalty, Authority	Harm, Authority
“Man In The Mirror”	Mark Chesnutt	Care, Fairness	Fairness, Loyalty, Authority	Care, Fairness, Loyalty, Authority	Care, Fairness

Table 2. Examples of moral values detected in song lyrics by human annotators and model predictions.

lion parameters, far exceeding BERT base model with 110 million parameters [42]. Such models demand significantly more computational resources than BERT models. In contrast, the BERT model is cost-free, easier to modify, and offers greater control over the models due to its open-source nature. On the other hand, BERT models need fine-tuning, which presents its own challenges due to the necessity for manual labelling and data annotation. Therefore, a hybrid approach like the one we suggest offers an optimised solution that combines the best of both worlds.

Table 2 presents four song examples annotated for moral values by both human annotators and prediction models. These examples show that MoralBERT SL and BERT SL (not shown in the table as it shares the same outcomes as MoralBERT SL for these instances) aligned most closely with human moral assessments. From a general observation of the song lyrics that were annotated by humans and tested with these models, it was noted that MoralBERT and GPT-4 tend to assign more moral attributes per song while increasing their chances of correctly guessing moral labels but also misclassifying neutral ones. In contrast, models trained with synthetic lyrics more accurately identified neutral (non-moral) lyrics, aligning with the quantitative observations of the F1 weighted score. Typically, human annotators did not assign more than three moral values per song. To control the number of assigned moral values per song, we adjusted the thresholds [37] for our prediction models, ensuring optimal accuracy. When lacking ground truth data, a post-processing can be applied for cutting moral labels with lower probabilities. Here we present only F1 and Precision scores. For further details, refer to the project’s results page on GitHub.²

5. CONCLUSION

In this paper, we presented an integrated approach for the automatic detection of moral values in lyrics. We created a synthetic lyrics dataset using GPT-4 which we used to fine-tune the BERT-base model alone (BERT SL) and in combination with out-of-domain social media corpora (MoralBERT SL). We introduced a dataset of 200 song lyrics

sourced from the WASABI dataset annotated for moral values by two experts, serving as the basis for evaluating our moral prediction models. We also assessed the performance of models trained with synthetic lyrics in comparison to those trained solely on social media data (MoralBERT) and a zero-shot GPT-4 classifier. We found that models trained with synthetic lyrics generally achieved significantly better binary Precision and higher weighted F1 scores compared to the GPT-4 classifier and MoralBERT, along with marginally better binary F1.

Our research has some limitations. To begin with, the synthetic lyrics is created via GPT-4, a powerful model but not an open-source, which limits our control of the model. We prompted GPT-4 to create unique lyrics in the style of various artists across different genres. Yet, adding musical composition details, lyrical themes [43], or visual images as descriptors [44], could enhance both the quality and diversity of the generated lyrics. However, we only employ this method for fine-tuning to make BERT models learn the structure and moral expressions in lyrics. The creation of truly creative lyrics for artistic purposes requires greater sophistication and rigorous human review [44]. Further, we analysed the overall moral expressions in the song lyrics without differentiating between structural elements such as verses, bridges, and choruses. Lastly, we focus on inferring moral values in English lyrics, which limits our ability to understand moral expressions in music lyrics from non-Western cultures.

Understanding how lyrics can convey moral values is important for the MIR field, as it can enhance how we experience and interact with music, including improving music tagging and recommendation systems [45]. Addressing challenges in automatic detection of moral values in lyrics can further push the boundaries of current technologies in natural language processing and machine learning applied to music and other creative tasks. Further, as lyrics often reflect societal values and cultural norms, tools for extracting morality rapidly from lyrical text enable researchers to gain insights into the prevailing moral attitudes of different times or cultures. This can be useful in sociological studies, helping scholars understand how music influences and is influenced by societal norms and changes.

² <https://github.com/vjosapreniqi/ismir-mft-values/tree/main/Results>

6. ETHICS STATEMENT

In this study, we employed large language models (LLMs) to generate synthetic lyrics. Given the vast amount of data on which these models are trained, there is a potential for bias transfer from the training datasets. Additionally, these models may inadvertently contain copyrighted literary works within their training data, necessitating meticulous steps to prevent plagiarism, particularly if the generated lyrics are utilised beyond fine-tuning for artistic and creative outputs [46,47].

We engaged two human annotators to label 200 songs with moral values based on the Moral Foundations Theory (MFT). These annotators signed a consent document that detailed the project’s objectives, their roles, and the nature of their tasks. They were informed of their right to withdraw from the study at any time without consequences. To protect their privacy, all data from the annotators were anonymised.

While powerful language models like BERT and GPT-4 offer significant potential to enhance communication and support social campaigns, they also pose risks if used for manipulative purposes. Our research is committed to advancing the understanding of moral expressions in music and fostering the responsible development and use of AI in creative contexts.

7. ACKNOWLEDGEMENTS

VP and IG are supported by PhD studentships from Queen Mary University of London’s Centre for Doctoral Training in Data-informed Audience-centric Media Engineering. KK acknowledges support from the Lagrange Project of the Institute for Scientific Interchange Foundation (ISI Foundation) which is funded by Fondazione Cassa di Risparmio di Torino (Fondazione CRT).

8. REFERENCES

- [1] M. E. Ballard and S. Coates, “The immediate effects of homicidal, suicidal, and nonviolent heavy metal and rap songs on the moods of college students,” *Youth & Society*, vol. 27, no. 2, pp. 148–168, 1995.
- [2] S. Frith, *Sound effects; youth, leisure, and the politics of rock’n’roll*. Pantheon Books, 1981.
- [3] L. Betti, C. Abrate, and A. Kaltenbrunner, “Large scale analysis of gender bias and sexism in song lyrics,” *EPJ Data Science*, vol. 12, no. 1, p. 10, 2023.
- [4] D. R. Dewberry and J. H. Millen, “Music as rhetoric: Popular music in presidential campaigns,” *Atlantic Journal of Communication*, vol. 22, no. 2, pp. 81–92, 2014.
- [5] E. J. Kizer, “Protest song lyrics as rhetoric,” *Popular Music & Society*, vol. 9, no. 1, pp. 3–11, 1983.
- [6] J. O. Adebayo, “Vote not Fight: Examining music’s role in fostering non-violent elections in Nigeria,” *African Journal on Conflict Resolution*, vol. 17, no. 1, pp. 55–77, 2017.
- [7] D. D. Sellnow, “Music as persuasion: Refuting hegemonic masculinity in “He Thinks He’ll Keep Her”,” *Women’s Studies in Communication*, vol. 22, no. 1, pp. 66–84, 1999.
- [8] R. Mayer and A. Rauber, “Musical genre classification by ensembles of audio and lyrics features,” in *Proceedings of the International Society for Music Information Retrieval (ISMIR)*, 2011, pp. 675–680.
- [9] R. Delbouys, R. Hennequin, F. Piccoli, J. Royo-Letelier, and M. Moussallam, “Music mood detection based on audio and lyrics with deep neural net,” in *Proceedings of the International Society for Music Information Retrieval (ISMIR)*, 2018, pp. 370–375.
- [10] Y. Song and D. Beck, “Modeling emotion dynamics in song lyrics with state space models,” *Transactions of the Association for Computational Linguistics*, vol. 11, pp. 157–175, 2023.
- [11] A. Vaglio, R. Hennequin, M. Moussallam, G. Richard, and F. d’Alché Buc, “Multilingual lyrics-to-audio alignment,” in *Proceedings of the International Society for Music Information Retrieval (ISMIR)*, 2020, pp. 512–519.
- [12] N. L. Masclef, A. Vaglio, and M. Moussallam, “User-centered evaluation of lyrics-to-audio alignment,” in *Proceedings of the International Society for Music Information Retrieval (ISMIR)*, 2021, pp. 420–427.
- [13] J. Kim, A. M. Demetriou, S. Manolios, M. S. Tavella, and C. C. Liem, “Butter lyrics over hominy grit: Comparing audio and psychology-based text features in mir tasks,” in *Proceedings of the International Society for Music Information Retrieval (ISMIR)*, 2020, pp. 861–868.
- [14] V. Preniqi, K. Kalimeri, and C. Saitis, “Soundscapes of morality: Linking music preferences and moral values through lyrics and audio,” *PLOS One*, 2023.
- [15] V. Preniqi, I. Ghinassi, C. Ive, Juliaand Saitis, and K. Kalimeri, “Moralbert: A fine-tuned language model for capturing moral values in social discussions,” in *ACM 4th International Conference on Information Technology for Social Good (GoodIT)*, 2024.
- [16] S. Guo, N. Mokhberian, and K. Lerman, “A data fusion framework for multi-domain morality learning,” in *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 17, 2023, pp. 281–291.
- [17] V. Preniqi, K. Kalimeri, and C. Saitis, ““More Than Words”: Linking music preferences and moral values through lyrics,” in *Proceedings of the International Society for Music Information Retrieval (ISMIR)*, 2022, pp. 797–805.

- [18] J. Haidt and J. Graham, “When morality opposes justice: Conservatives have moral intuitions that liberals may not recognize,” *Social Justice Research*, vol. 20, no. 1, pp. 98–116, 2007.
- [19] J. Hoover, K. Johnson, R. Boghrati, J. Graham, and M. Deghani, “Moral framing and charitable donation: Integrating exploratory social media analyses and confirmatory experimentation,” *Collabra: Psychology*, vol. 4, no. 1, 2018.
- [20] V. Preniqi, K. Kalimeri, and C. Saitis, “Modelling moral traits with music listening preferences and demographics,” *Music in the AI Era. CMMR 2021. Lecture Notes in Computer Science*, vol. 13770, pp. 183–194, 2021.
- [21] K. J. Messick and B. E. Aranda, “The role of moral reasoning & personality in explaining lyrical preferences,” *PLOS One*, vol. 15, no. 1, p. e0228057, 2020.
- [22] K. Kalimeri, M. G. Beiró, M. Delfino, R. Raleigh, and C. Cattuto, “Predicting demographics, moral foundations, and human values from digital behaviours,” *Computers in Human Behavior*, vol. 92, pp. 428–445, 2019.
- [23] O. Araque, L. Gatti, and K. Kalimeri, “MoralStrength: Exploiting a moral lexicon and embedding similarity for moral foundations prediction,” *Knowledge-Based Systems*, vol. 191, pp. 1–11, 2020.
- [24] F. R. Hopp, J. T. Fisher, D. Cornell, R. Huskey, and R. Weber, “The extended moral foundations dictionary (eMFD): Development and applications of a crowdsourced approach to extracting moral intuitions from text,” *Behavior Research Methods*, vol. 53, pp. 232–246, 2021.
- [25] J. Graham, J. Haidt, and B. A. Nosek, “Liberals and conservatives rely on different sets of moral foundations,” *Journal of Personality and Social Psychology*, vol. 96, no. 5, pp. 1029–1046, 2009.
- [26] J. Hoover, G. Portillo-Wightman, L. Yeh, S. Havaldar, A. M. Davani, Y. Lin, B. Kennedy, M. Atari, Z. Kamel, M. Mendlen *et al.*, “Moral foundations Twitter corpus: A collection of 35k tweets annotated for moral sentiment,” *Social Psychological and Personality Science*, vol. 11, no. 8, pp. 1057–1071, 2020.
- [27] J. Trager, A. S. Ziabari, A. M. Davani, P. Golazazian, F. Karimi-Malekabadi, A. Omrani, Z. Li, B. Kennedy, N. K. Reimer, M. Reyes *et al.*, “The moral foundations Reddit corpus,” *arXiv preprint arXiv:2208.05545*, 2022.
- [28] M. G. Beiró, J. D’Ignazi, V. Perez Bustos, M. F. Prado, and K. Kalimeri, “Moral narratives around the vaccination debate on Facebook,” in *Proceedings of the ACM Web Conference 2023*, 2023, pp. 4134–4141.
- [29] Y. Ganin and V. Lempitsky, “Unsupervised domain adaptation by backpropagation,” in *Proceedings of International Conference on Machine Learning*, 2015, pp. 1180–1189.
- [30] E. Liscio, O. Araque, L. Gatti, I. Constantinescu, C. Jonker, K. Kalimeri, and P. K. Murukannaiah, “What does a text classifier learn about morality? an explainable method for cross-domain comparison of moral rhetoric,” in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2023, pp. 14 113–14 132.
- [31] S. Doh, K. Choi, J. Lee, and J. Nam, “LP-MusicCaps: Llm-based pseudo music captioning,” in *Proceedings of the International Society for Music Information Retrieval (ISMIR)*, 2023.
- [32] Z. Zhang, K. Lasocki, Y. Yu, and A. Takasu, “Syllable-level lyrics generation from melody exploiting character-level language model,” in *Findings of the Association for Computational Linguistics: EACL 2024*, 2024, pp. 1336–1346.
- [33] P. Sawicki, M. Grzes, L. F. Góes, D. Brown, M. Peepkorn, A. Khatun, and S. Paraskevopoulou, “On the power of special-purpose GPT models to create and evaluate new poetry in old styles,” *University of Leicester*, 2023.
- [34] J. R. Landis and G. G. Koch, “The measurement of observer agreement for categorical data,” *Biometrics*, pp. 159–174, 1977.
- [35] G. Meseguer-Brocal, G. Peeters, G. Pellerin, M. Buffa, E. Cabrio, C. Faron Zucker, A. Giboin, I. Mirbel, R. Hennequin, M. Moussallam *et al.*, “WASABI: A two million song database project with audio and cultural metadata plus weaudio enhanced client applications,” *Web Audio Conference (WAC)*, 2017.
- [36] G. King and L. Zeng, “Logistic regression in rare events data,” *Political Analysis*, vol. 9, no. 2, pp. 137–163, 2001.
- [37] O. Koshorek, A. Cohen, N. Mor, M. Rotman, and J. Berant, “Text segmentation as a supervised learning task,” *arXiv preprint arXiv:1803.09337*, 2018.
- [38] J. Ye, J. Gao, Q. Li, H. Xu, J. Feng, Z. Wu, T. Yu, and L. Kong, “ZeroGen: Efficient zero-shot learning via dataset generation,” in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Y. Goldberg, Z. Kozareva, and Y. Zhang, Eds. Association for Computational Linguistics, Dec. 2022, pp. 11 653–11 669.
- [39] X. He, I. Nassar, J. Kiros, G. Haffari, and M. Norouzi, “Generate, annotate, and learn: Nlp with synthetic text,” *Transactions of the Association for Computational Linguistics*, vol. 10, pp. 826–842, 2022.

- [40] M. O. Silva, L. M. Rocha, and M. M. Moro, “MusicOSet: An enhanced open dataset for music data mining,” in *XXXII Simpósio Brasileiro de Banco de Dados: Dataset Showcase Workshop, SBBD*, 2019, pp. 8–17, Accessed online: 2024-02-05.
- [41] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, “Language models are few-shot learners,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 1877–1901, 2020.
- [42] M. Bosley, M. Jacobs-Harukawa, H. Licht, and A. Hoyle, “Do we still need BERT in the age of GPT? comparing the benefits of domain-adaptation and in-context-learning approaches to using llms for political science research,” *University of Michigan*, 2023.
- [43] K. Watanabe, Y. Matsubayashi, K. Inui, T. Nakano, S. Fukayama, and M. Goto, “Lyrisys: An interactive support system for writing lyrics based on topic transition,” in *Proceedings of the 22nd International Conference on Intelligent User Interfaces*, 2017, pp. 559–563.
- [44] K. Watanabe and M. Goto, “Text-to-lyrics generation with image-based semantics and reduced risk of plagiarism,” in *Proceedings of the International Society for Music Information Retrieval (ISMIR)*, 2023.
- [45] A. Laplante, “Improving music recommender systems: What can we learn from research on music tastes?” in *Proceedings of the International Society for Music Information Retrieval (ISMIR)*, 2014, pp. 451–456.
- [46] J. Barnett, “The ethical implications of generative audio models: A systematic literature review,” in *Proceedings of the 2023 AAAI/ACM Conference on AI, Ethics, and Society*, 2023, pp. 146–161.
- [47] F. Morreale, M. Sharma, I. Wei *et al.*, “Data collection in music generation training sets: A critical analysis,” in *Proceedings of the International Society for Music Information Retrieval (ISMIR)*, 2023.

SEMI-SUPERVISED PIANO TRANSCRIPTION USING PSEUDO-LABELING TECHNIQUES

Sebastian Strahl, Meinard Müller

International Audio Laboratories Erlangen, Germany

{sebastian.strahl,meinard.mueller}@audiolabs-erlangen.de

ABSTRACT

Automatic piano transcription (APT) transforms piano recordings into symbolic note events. In recent years, APT has relied on supervised deep learning, which demands a large amount of labeled data that is often limited. This paper introduces a semi-supervised approach to APT, leveraging unlabeled data with techniques originally introduced in computer vision (CV): pseudo-labeling, consistency regularization, and distribution matching. The idea of pseudo-labeling is to use the current model for producing artificial labels for unlabeled data, and consistency regularization makes the model’s predictions for unlabeled data robust to augmentations. Finally, distribution matching ensures that the pseudo-labels follow the same marginal distribution as the reference labels, adding an extra layer of robustness. Our method, tested on three piano datasets, shows improvements over purely supervised methods and performs comparably to existing semi-supervised approaches. Conceptually, this work illustrates that semi-supervised learning techniques from CV can be effectively transferred to the music domain, considerably reducing the dependence on large annotated datasets.

1. INTRODUCTION

Automatic music transcription (AMT) converts polyphonic music recordings into symbolic representations that encode which notes are played [1, 2]. The AMT output may be a MIDI-like transcription, containing for every note event information about the instrument, onset time, duration, and velocity. AMT is considered as one of the fundamental problems in music information retrieval (MIR) because its symbolic output can be used for subsequent tasks such as music synchronization, structure analysis, or cover song detection [3]. AMT is challenging since multiple instruments may be active at the same time, due to possible polyphonic activity per instrument, and because sound events may have overlapping harmonics [2].

Early approaches to AMT rely, e.g., on non-negative matrix factorization [4, 5], while most recent approaches

use deep learning-based models [6–13]. The limiting factor in training neural networks for AMT, however, is the scarcity of labeled data. Creating such datasets typically requires manual labeling of each note present in a recording, which can be time-consuming, or relies on music synchronization techniques to align score information with recordings [11, 14]. The latter approach, however, may result in inaccurate labels due to issues such as playing errors or synchronization inaccuracies. Alternatively, one can create datasets with highly precise labels by utilizing instruments that allow automated playback or recording note activity. For instance, several piano datasets were automatically created using a Disklavier, which can synthesize MIDI files or log key activity during performance [15–17]. Since these piano datasets exist, many works [6–9, 12] focus on the special case of automatic piano transcription (APT). Still, it was observed that APT methods cannot generalize well across datasets due to overfitting [18].

In this work, we aim to improve model generalization of APT in scenarios with little labeled data by using semi-supervised learning (SSL), where the idea is to leverage unlabeled data during training. Unlabeled data can be obtained in large amounts as it does not depend on a labeling process. SSL has seen limited application in AMT, with Cheuk et al. [19] among the few to investigate this path. However, we argue that its full potential remains to be realized, especially when considering the significant achievements of SSL in computer vision (CV) [20, 21]. As our main contribution, we adapt techniques originally introduced in CV [22, 23] to APT. More specifically, our method makes use of pseudo-labeling, consistency regularization, and distribution matching as outlined in the following.

In our approach, we use the extended Onsets and Frames model [7, 16], which jointly predicts onsets, offsets, frame activity, and velocities. The raw model outputs for onsets, offsets, and frames are each a piano roll-like representation that can be interpreted as probabilities per time–pitch bin. Initially, we pre-train this model in a supervised fashion using the available labeled data. Thereafter, the model is used to produce binary pseudo-labels for unlabeled data. Only sufficiently confident predictions are converted into pseudo-labels, i.e., those below the lower threshold are set to zero and those above the upper threshold are set to one, while the remaining predictions are considered as unreliable. Next, the model makes predictions for an augmented version of the same recording, where augmentation involves frequency masking [24] and addi-



tion of noise to the data. The predictions made for the augmented data are then used in combination with the pseudo-labels derived from the clean data to compute an additional unsupervised loss. Using an augmented version instead of a clean one encourages the model to produce consistent predictions under these kinds of augmentations and is thus called consistency regularization. As a third technique, we apply distribution matching, which ensures that the pseudo-labels follow the same marginal distribution as the reference labels, preventing the model from collapsing. To achieve this goal, we use an undersampling strategy. For reproducibility, we will provide our code ¹.

The rest of this paper is structured as follows: In Section 2, we give an overview of related work on AMT, SSL, and distribution matching in the context of pseudo-labeling. In Section 3, we describe all steps of the proposed approach. Section 4 describes our experimental setup as well as the experimental results. We conclude the paper in Section 5 with possible future research directions.

2. BACKGROUND AND RELATED WORK

2.1 Automatic Music Transcription

Most research on AMT is based on supervised learning. Sigtia et al. [6] proposed the the first end-to-end approach to APT. Hawthorne et al. [7] emphasized the importance of explicitly predicting onsets alongside frame activity, later extending their model in [16] to include explicit prediction of offsets. In [8], onset and offset estimation is formulated as a regression problem, which yields note predictions with improved temporal resolution. The attention-based Transformer architecture is used for APT [9, 12, 25] and multi-instrument AMT [10]. In [13], the Perceiver architecture is employed for multi-instrument AMT. Recently, AMT has been formulated as a conditional generative task: In [26], a diffusion model is trained to generate realistic piano rolls, being conditioned on the corresponding spectrograms.

Weakly supervised methods are proposed in [11], where unaligned pairs of scores and recordings are used for training, and in [27], where cross-version targets are used to replace pitch labels. Cheuk et al. [19] propose a semi-supervised approach to AMT, utilizing unlabeled data via virtual adversarial training (VAT). VAT [28] perturbs input data to induce substantial changes in the model’s predictions and then encourages the model to produce consistent predictions under these perturbations. In [29], a fully self-supervised method is proposed for frame-level transcription. Their method encourages the concentration of energy around fundamental frequency candidates, invariance to timbral transformations, and equivariance to input translations in both time and frequency.

2.2 Semi-Supervised Learning

In SSL, the idea is to jointly learn from labeled and unlabeled data, and SSL is thus located between supervised and unsupervised learning [30,31]. The objective is to train a model that performs better than a reference model only

trained on the labeled data using supervised learning. SSL has been successfully used in combination with deep learning, e. g., in CV [20, 21], for text classification [32], and also in MIR [33, 34]. For an overview of deep learning-based SSL methods, we refer to [20, 35]. Two important SSL paradigms relevant to this paper are pseudo-labeling and consistency regularization.

Pseudo-labeling, introduced in [36], uses the current classification model to produce artificial labels for unlabeled data. Continuing training with pseudo-labeled data encourages the model to make confident predictions for that data, effectively pushing decision boundaries away from the data points [35]. Maman and Bermano [11] already combined pseudo-labeling and weak supervision for AMT, but the pseudo-labels were updated only at the beginning of every expectation maximization iteration rather than being calculated on-the-fly as in [36].

Consistency regularization methods [37, 38] encourage that the model’s predictions do not change if augmentations (e. g., random translation and addition of noise in the case of image classification [37, 38]) are applied to the unlabeled input data. In [37], this is achieved by adding a consistency loss term which penalizes disagreement in the predictions made for two augmented versions of the data.

The image classification method FixMatch [22] combines both pseudo-labeling and consistency regularization by using the current model to produce artificial labels given a weakly augmented input (e. g., horizontally flipped) to supervise the predictions made for a strongly augmented input (e. g., Cutout [39], where a randomly selected rectangular region is masked). In [40, 41], FixMatch proved to be effective for audio classification as well, where weak and strong augmentations were applied to spectrograms. FixMatch was also adapted to pixel-wise classification problems such as semantic image segmentation [42], which is similar to AMT from a technical point of view.

2.3 Distribution Matching

It is well-known that training classification models on class-imbalanced data is challenging because the models tend to be biased towards the majority classes [43]. Biased model predictions which do not follow a similar distribution as the reference labels are problematic for pseudo-labeling because the model may suffer from confirmation bias [44], where wrong predictions are reinforced. To avoid that problem, several approaches were proposed to match the class distribution of pseudo-labels with that of reference labels. Berthelot et al. [23] rescale the predicted class probabilities for unlabeled data in such a way that their marginal distribution is close to the marginal distribution of reference labels. Kim et al. [45] refine pseudo-labels by solving a convex optimization problem that aims to minimize the distance between pseudo-label distribution and reference label distribution while trying to preserve most information in the pseudo-labels. While Maman and Bermano [11] do not explicitly perform distribution matching for AMT, they set asymmetric thresholds for selecting pseudo-labels, increasing the impact of the minority class.

¹ https://github.com/groupmm/onsets_frames_semisup

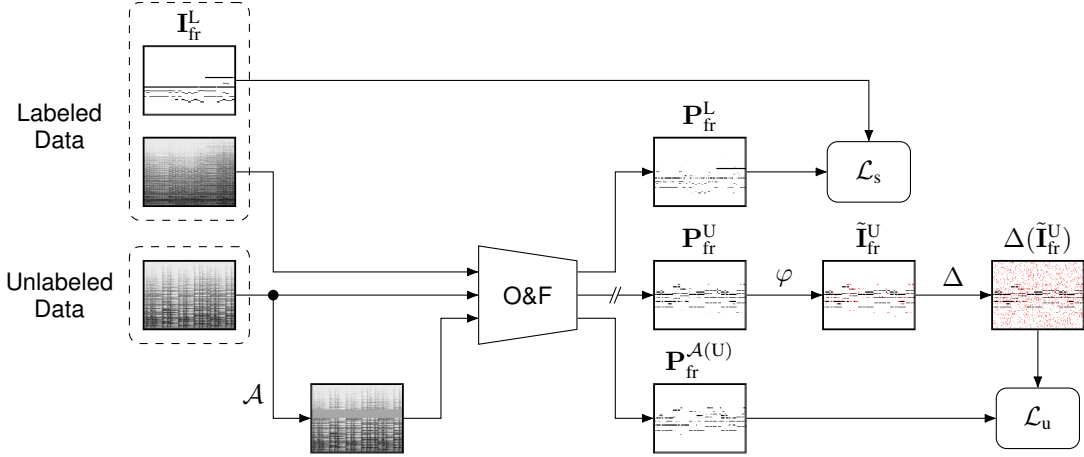


Figure 1: Detailed overview of our semi-supervised approach. The Onsets and Frames transcription model (O&F) [7, 16] is trained using both a supervised (upper branch) and an unsupervised loss (lower branches). Our method uses a clean version of unlabeled data to produce predictions, which, after thresholding (φ), are considered as pseudo-labels. Distribution matching (Δ) ensures that pseudo-labels and reference labels are similarly distributed. The pseudo-labels are used to supervise predictions made for an augmented (\mathcal{A}) version of the same data. The “interrupted” connection to the predictions made for the clean unlabeled input indicates that gradients are not backpropagated in this branch. For a better overview, we only show predictions, labels, and pseudo-labels for frame activity. Red color is used to represent NaN entries.

3. METHOD

In this section, we describe our proposed semi-supervised approach for learning APT. We first describe in Section 3.1 how the transcription model is trained in a supervised fashion. In Section 3.2, we explain how pseudo-labeling and consistency regularization can be used for semi-supervised training, and in Section 3.3, we explain the additional step of matching the pseudo-label distribution with the reference label distribution.

3.1 Supervised APT Baseline

We use the modified Onsets and Frames model [7, 16] and train our supervised APT baseline models similar to the original methodology. This model takes as input a log mel-scaled spectrogram with F frequency bins and T frames, and outputs onset, offset, frame activity, and velocity estimates. In this work, we focus on the involved classification problems and ignore velocity estimation for simplicity. Velocity estimation can be omitted without further consequences, as it is performed by an independent part of the model. We briefly explain how supervised learning is done using labeled data. The model outputs matrices $\mathbf{P}_{\text{on}}^L, \mathbf{P}_{\text{off}}^L, \mathbf{P}_{\text{fr}}^L \in [0, 1]^{P \times T}$ for onset, offset, and frame activity, respectively. In this notation, P denotes the number of MIDI pitches considered, and the entries of the matrices represent probabilities of activities for all time–pitch bins. For instance, $\mathbf{P}_{\text{on}}^L(p, t)$ denotes the predicted probability of an onset with pitch p in frame t . The reference MIDI annotations with continuous-time note events are temporally quantized to match the input frame rate and converted into binary labels $\mathbf{I}_{\text{on}}^L, \mathbf{I}_{\text{off}}^L, \mathbf{I}_{\text{fr}}^L \in \{0, 1\}^{P \times T}$, indicating bin-wise activities as described in [7, 16]. The supervised loss comprises three terms,

$$\mathcal{L}_s = \lambda_{\text{on}}^L \mathcal{L}_{\text{on}}^L + \lambda_{\text{off}}^L \mathcal{L}_{\text{off}}^L + \lambda_{\text{fr}}^L \mathcal{L}_{\text{fr}}^L, \quad (1)$$

with the frame activity loss

$$\mathcal{L}_{\text{fr}}^L = \frac{1}{PT} \sum_{p=1}^P \sum_{t=1}^T \ell_{\text{BCE}}(\mathbf{I}_{\text{fr}}^L(p, t), \mathbf{P}_{\text{fr}}^L(p, t)), \quad (2)$$

where ℓ_{BCE} denotes the binary cross entropy function and $\lambda_{\text{on}}^L, \lambda_{\text{off}}^L, \lambda_{\text{fr}}^L \in [0, 1]$ are suitable loss weights. Onset and offset loss terms are defined analogously. Note that, in contrast to [7], we leave out the weighting of individual frames within the frame activity loss in Equation (2) for simplicity.

3.2 Pseudo-Labeling and Consistency Regularization

We now describe how our approach leverages unlabeled data, which is illustrated in Figure 1. Our method is mainly inspired by FixMatch [22], with the difference that we do not apply weak augmentations to produce pseudo-labels. Instead, we produce pseudo-labels using the unmodified, clean data, which has been found to yield nearly the same results in audio classification [40].

To obtain pseudo-labels for unlabeled data, we first compute the current model’s predictions, $\mathbf{P}_{\text{on}}^U, \mathbf{P}_{\text{off}}^U, \mathbf{P}_{\text{fr}}^U \in [0, 1]^{P \times T}$, given the clean version of the log mel-scaled spectrogram as input. For converting soft probabilities into binary pseudo-labels, we define a thresholding function

$$\varphi(x, \tau_{\text{lo}}, \tau_{\text{up}}) = \begin{cases} 1, & \text{if } x \geq \tau_{\text{up}}, \\ \text{NaN}, & \text{if } \tau_{\text{lo}} < x < \tau_{\text{up}}, \\ 0, & \text{if } x \leq \tau_{\text{lo}}, \end{cases} \quad (3)$$

where τ_{lo} and τ_{up} denote lower and upper threshold, respectively. We obtain the pseudo-labels $\tilde{\mathbf{I}}_{\text{on}}^U, \tilde{\mathbf{I}}_{\text{off}}^U$, and $\tilde{\mathbf{I}}_{\text{fr}}^U$ by elementwise application of the thresholding function to the model predictions, i. e.,

$$\tilde{\mathbf{I}}_{\text{fr}}^U(p, t) = \varphi(\mathbf{P}_{\text{fr}}^U(p, t), \tau_{\text{lo}}, \tau_{\text{up}}) \quad (4)$$

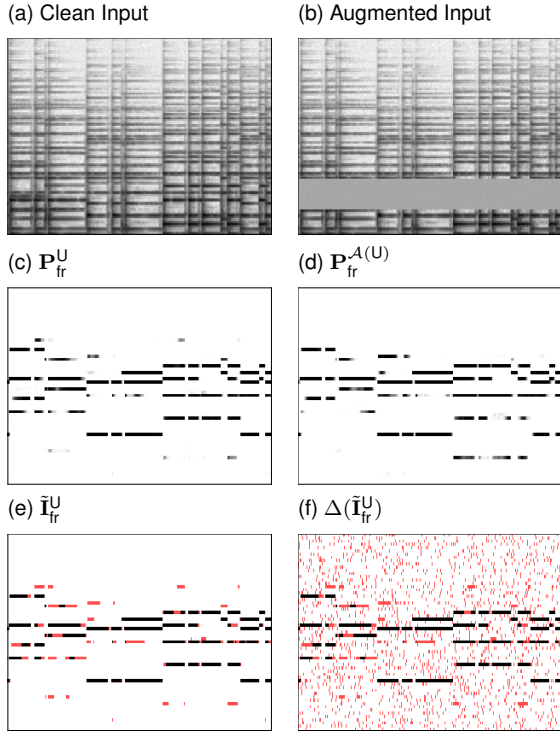


Figure 2: Examples of the representations involved in our semi-supervised method. Red color is used to represent NaN entries.

for $p \in [1 : P]$, $t \in [1 : T]$, and similarly for onsets and offsets. We use thresholds $\tau_{lo} = 0.05$ and $\tau_{up} = 0.95$ based on our observations in preliminary experiments, and we perform an ablation of this choice in Section 4. For illustration purposes, we refer to Figure 2, showing examples of clean model input, corresponding predictions \mathbf{P}_{fr}^U , and pseudo-labels $\tilde{\mathbf{I}}_{fr}^U$ in Figures 2a, 2c, and 2e, respectively, where NaN entries are represented by red color.

To perform consistency regularization, the pseudo-labels are used to supervise predictions made for an augmented version of the input. As in [40], we apply augmentations to the spectrograms. We opt for a simple augmentation pipeline which first applies frequency masking as described in [24], setting a randomly selected contiguous frequency band of up to 30 bins to the mean value of the spectrogram, and afterwards adds Gaussian noise with a standard deviation of 0.01 to the entire spectrogram. This choice of augmentation is inspired by the use of Cutout [39] in FixMatch [22] and the proposal of SpecAugment [24] as similar technique for spectrograms. We decided against temporal masking because this may completely remove information from the spectrogram regarding short events such as onsets. An example of such an augmented spectrogram is shown in Figure 2b. We denote the augmentation pipeline by \mathcal{A} , and the model’s predictions for the augmented input are denoted by $\mathbf{P}_{on}^{A(U)}$, $\mathbf{P}_{off}^{A(U)}$, $\mathbf{P}_{fr}^{A(U)} \in [0, 1]^{P \times T}$, respectively. An example of such predictions is shown in Figure 2d. Finally, the unsupervised loss is given by

$$\mathcal{L}_u = \lambda_{on}^U \mathcal{L}_{on}^U + \lambda_{off}^U \mathcal{L}_{off}^U + \lambda_{fr}^U \mathcal{L}_{fr}^U, \quad (5)$$

with the frame activity loss for unlabeled data,

$$\mathcal{L}_{fr}^U = \frac{1}{PT} \sum_{\substack{(p,t) \in [1:P] \times [1:T] : \\ \tilde{\mathbf{I}}_{fr}^U(p,t) \neq \text{NaN}}} \ell_{\text{BCE}}(\tilde{\mathbf{I}}_{fr}^U(p,t), \mathbf{P}_{fr}^{A(U)}(p,t)). \quad (6)$$

Onset and offset loss for unlabeled data are defined analogously. Only those time–pitch bins contribute to the loss, where the pseudo-labels have a value different from NaN. The loss is normalized by the total number of time–pitch bins for reducing the impact of the unsupervised loss if only a few predictions are confident. As for the supervised loss, we use suitable loss weights $\lambda_{on}^U, \lambda_{off}^U, \lambda_{fr}^U \in [0, 1]$. Note that the gradient of \mathcal{L}_u is not computed with respect to the predictions made for the clean version of the unlabeled input, which the “interrupted” connection in Figure 1 indicates. The overall loss function is obtained as the weighted sum of the supervised and the unsupervised loss,

$$\mathcal{L} = (1 - \lambda_u) \mathcal{L}_s + \lambda_u \mathcal{L}_u, \quad (7)$$

where $\lambda_u \in [0, 1]$ controls the relative weighting of both terms. Following [7], we weight the individual terms in the supervised loss equally, i. e., $\lambda_{on}^L = \lambda_{off}^L = \lambda_{fr}^L = 1$. However, preliminary experiments suggested that better results may be achieved if the unsupervised offset loss is not used. Hence, our default setting is $\lambda_{on}^U = \lambda_{fr}^U = 1$ and $\lambda_{off}^U = 0$. The overall weight of the unsupervised loss is set to $\lambda_u = 0.05$. We explore the impact of these hyperparameter choices through ablation studies in Section 4.

3.3 Distribution Matching

The classification problems involved in training transcription models are heavily imbalanced because the labels typically have only a few non-zero entries. For example, the training set of the MAPS dataset [15] has labels, where only about 0.3% of all entries are ones for both onsets and offsets, and about 3.4% of all entries are ones for frame activity. Hence, the transcription model may be biased towards predicting zeros. To avoid model collapse, we apply distribution matching to the pseudo-labels.

In this paper, we employ a simple method to match the marginal pseudo-label distribution per mini-batch with that of the reference labels. The marginal distribution of the reference labels is estimated by counting zeros and ones across all training examples. These counting operations are denoted by Γ_0 and Γ_1 . The following distribution matching method, explained using frame activity as an example, is similarly applied to onsets and offsets.

During training, we count the numbers of zeros and ones for every mini-batch of pseudo-labels, and will likely obtain a ratio $\Gamma_1(\tilde{\mathbf{I}}_{fr}^U)/\Gamma_0(\tilde{\mathbf{I}}_{fr}^U)$ that differs from the desired ratio $\Gamma_1(\mathbf{I}_{fr}^L)/\Gamma_0(\mathbf{I}_{fr}^L)$. The objective of the distribution matching operator, denoted by Δ , is to ensure that the ratio of zeros and ones is identical for reference labels and pseudo-labels, i. e.,

$$\frac{\Gamma_1(\mathbf{I}_{fr}^L)}{\Gamma_0(\mathbf{I}_{fr}^L)} = \frac{\Gamma_1(\Delta(\tilde{\mathbf{I}}_{fr}^U))}{\Gamma_0(\Delta(\tilde{\mathbf{I}}_{fr}^U))}. \quad (8)$$

	Thresholds		MAPS						MAESTRO						SMD					
			Note			Frame			Note			Frame			Note			Frame		
	τ_{on}	τ_{fr}	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Full																				
RV	0.50	0.50	80.9	70.6	75.1	85.9	72.0	77.9	-	-	-	-	-	-	-	-	-	-	-	-
OF	0.44	0.57	84.4	77.8	80.8	81.5	61.3	69.4	88.5	80.9	84.2	85.4	43.5	55.8	92.7	82.9	87.3	66.0	61.7	63.1
OF-SS4	0.35	0.34	84.7	79.6	81.9	78.3	67.5	72.0	93.3	82.7	87.5	84.5	53.2	63.5	94.7	85.5	89.7	63.1	69.0	65.2
Small																				
RV	0.50	0.50	86.2	57.1	68.2	90.0	43.9	58.2	-	-	-	-	-	-	-	-	-	-	-	-
OF	0.34	0.01	79.3	62.1	69.1	68.7	53.5	59.3	84.3	61.0	69.7	79.0	36.8	48.0	81.0	67.7	73.0	58.4	47.1	51.1
OF-SS4	0.05	0.01	78.2	75.9	76.7	62.3	69.9	65.0	93.8	78.4	85.0	73.6	56.3	61.8	93.6	80.2	85.9	52.3	68.1	58.2
One-Shot																				
RV	0.50	0.50	77.2	51.1	60.7	86.1	31.4	45.0	-	-	-	-	-	-	-	-	-	-	-	-
OF	0.02	0.01	66.5	56.0	60.2	67.4	35.3	45.2	76.5	50.9	59.9	76.7	23.3	34.0	69.0	57.9	62.1	56.3	32.3	39.8
OF-SS4	0.03	0.01	66.2	68.0	66.6	49.8	35.0	40.0	73.6	70.2	71.3	57.4	26.1	33.8	72.6	71.5	71.4	40.3	32.4	34.9
MB	0.50	0.50	88.2	86.5	87.3	84.4	76.7	79.6	92.6	87.2	89.7	77.4	76.1	76.0	-	-	-	-	-	-

Table 1: Performance metrics in percentages evaluated on the test sets of MAPS (ENSTDkAm and ENSTDkCl) and MAESTRO, and on the entire SMD dataset. Performance metrics are calculated per piece and then averaged over all pieces in the respective sets. As for the transcription models, RV is ReconVAT [19], OF is Onsets and Frames [16], OF-SS4 is our proposed semi-supervised method, and MB stands for Maman and Bermano [11]. Decision thresholds of OF and OF-SS4 are tuned using the group S_{ptkBGAm} of the MAPS dataset. F1 scores are highlighted in red for better readability.

To define Δ , we use undersampling as it is frequently used for class-imbalanced learning [46]. The distribution matching works as follows:

1. Determine whether the ratio $\Gamma_1(\tilde{\mathbf{I}}_{\text{fr}}^{\text{U}})/\Gamma_0(\tilde{\mathbf{I}}_{\text{fr}}^{\text{U}})$ is smaller or larger than the ratio $\Gamma_1(\mathbf{I}_{\text{on}}^{\text{L}})/\Gamma_0(\mathbf{I}_{\text{on}}^{\text{L}})$, i. e., whether there is an excess of zeros or ones, respectively, among the pseudo-labels.
2. Randomly select the required number of excess zeros or ones and convert them to NaN entries to obtain the desired ratio.

Distribution matching reduces the number of available pseudo-labels but ensures that the pseudo-labels within a mini-batch follow the same marginal distribution as the reference labels. An example of distribution-matched pseudo-labels is shown in Figure 2f.

4. EXPERIMENTS

4.1 Implementation Details

For our experiments, we use an open-source Pytorch implementation² of Onsets and Frames [7, 16]. Input representation and model architecture are unchanged compared to [7]. However, we do not ensure that input segments do not start in the middle of a note as it is done in [7]. We use a batch size of 8 each for labeled and unlabeled data and average losses across batches. We train our models using the Adam optimizer [47] with an initial learning rate of $6e-5$ and multiply the learning rate by a factor of 0.98 every 5k iterations. Also, we apply gradient clipping with norm 3. All audio recordings were downsampled to 16 kHz.

4.2 Datasets

We train and evaluate our models on three piano datasets: MAPS [15], MAESTRO V3.0.0 [16], and SMD [17].

² <https://github.com/jongwook/onsets-and-frames>

MAPS [15] contains isolated notes, chords, and complete piano pieces, but we only make use of the complete pieces. This dataset contains nine groups with 30 recordings each, where seven of the groups contain synthesized recordings, and the remaining two groups (ENSTDkAm and ENSTDkCl) contain real recordings which were automatically generated from MIDI files using a Disklavier. Following previous work [6, 7, 19], we use the groups with synthetic data as training data, and the real recordings as test data, and we remove the pieces from the training data which are also contained in the test data. This yields training and test sets of 139 and 60 recordings, respectively.

MAESTRO [16] and SMD [17] provide recordings together with the corresponding MIDI annotations automatically captured by a Disklavier. Both MAESTRO and SMD contain actual recordings of live performances, from the International Piano-e-Competition and played by music students, respectively. MAESTRO comprises 1276 performances, with the official data split assigning 962, 137, and 177 performances to the training, validation, and test set, respectively, and SMD comprises 50 performances.

4.3 Evaluation and Threshold Tuning

During inference, a decoding step is performed to obtain estimated note events from the network outputs [7, 16]. Two thresholds, τ_{on} and τ_{fr} , are applied to binarize onset and frame activity predictions. A note event is only recognized if an onset was detected, and the length of the note is determined based on the frame activity prediction. The offset prediction is not explicitly used during decoding.

Following existing literature, we evaluate model performance using note-based and frame-based metrics including precision (P), recall (R), and F1 score. Note-based metrics are computed using the *mir_eval* library [48], where a predicted note is considered as correct if its pitch matches that of a reference note and the onset is within ± 50 ms of that reference note’s onset.

Instead of using fixed thresholds τ_{on} and τ_{fr} , we tune these thresholds using a labeled validation set [27, 49]. We first determine an optimum τ_{on} via grid search so as to maximize the note F1 score, which does not depend on τ_{fr} . Since the frame-based metrics are computed based on the decoded note events, the frame F1 score is affected by both τ_{on} and τ_{fr} . We fix the previously found τ_{on} and determine the τ_{fr} that maximizes the frame F1 score.

4.4 Experimental Scenarios

To compare with [19], we adopt their three experimental scenarios which differ in the choice of the labeled data. The first scenario (*Full*) uses the full MAPS training set, the second scenario (*Small*) uses only the group `AkPnBcht` of the MAPS training set, which contains 23 non-overlapping piano pieces, and the third scenario (*One-Shot*) uses only a single recording (`chp_op31` from `AkPnBcht`) as labeled data. Note that for *One-Shot*, the batch size for labeled data needs to be reduced to 1. In all scenarios, the MAESTRO training set is used as unlabeled data. We use the group `SptkBGAm` of the MAPS training set as validation data—which overlaps with the labeled training data in the *Full* scenario.

In all scenarios, we start training the transcription model from scratch following the training strategy described in Section 3.1 for 50k iterations, using only the labeled data and supervised learning. After that pre-training stage, we train for another 50k iterations using our proposed semi-supervised method as described in Section 3.2. We refer to this model as `OF-SS4`. For a fair supervised baseline in each scenario, we also continue training the pre-trained model for another 50k iterations on only the labeled data, which we will refer to as `OF`.

4.5 Main Results

The main results of our experiments are provided in Table 1, where the models of all scenarios are evaluated on the test sets of MAPS and MAESTRO, and also on the independent SMD dataset. First, we can observe that `OF-SS4` achieves better F1 scores than `OF` almost in all scenarios and across all datasets, with the frame F1 score in the *One-Shot* scenario being the exception. Most notably, `OF-SS4` achieves a note F1 score of 85.0 on the MAESTRO test set in the scenario *Small*, which slightly exceeds the note F1 score 84.2 of `OF` in the scenario *Full*. This shows that our semi-supervised approach is indeed effective, reducing the number of labeled performances by more than 80% for achieving comparable performance in this case. We further note that the optimum decision thresholds of `OF` and `OF-SS4` are extremely low for the scenarios *Small* and *One-Shot*, indicating that threshold tuning is an important step if labeled training data is scarce.

For ReconVAT (RV) [19], we report for every scenario the performance of their semi-supervised method that achieved the highest note F1 score. Still, we observe that `OF-SS4` achieves higher note F1 scores than RV in all scenarios, e. g., 76.7 for `OF-SS4` compared to 68.2 for RV in the scenario *Small*. Regarding the frame F1 score, no clear

	τ_{lo}	τ_{up}	\mathcal{A}	Δ	λ_{off}^U	λ_u	N-F1	F-F1
<code>OF</code>	-	-	-	-	-	-	73.0	51.1
<code>OF-SS1</code>	0.05	0.95	-	-	0.0	0.05	0.1	3.0
<code>OF-SS2</code>	0.05	0.95	-	✓	0.0	0.05	82.4	9.4
<code>OF-SS3</code>	0.05	0.95	✓	-	0.0	0.05	82.7	57.6
<code>OF-SS4</code>	0.05	0.95	✓	✓	0.0	0.05	85.9	58.2
<code>OF-SS5</code>	0.25	0.75	✓	✓	0.0	0.05	74.6	51.6
<code>OF-SS6</code>	0.05	0.95	✓	✓	1.0	0.05	85.6	56.3
<code>OF-SS7</code>	0.05	0.95	✓	✓	0.0	0.01	72.8	51.5

Table 2: Results of an ablation study performed in the scenario *Small*, evaluated on the independent SMD dataset [17]. N-F1 and F-F1 are note F1 score and frame F1 score in percentage, respectively.

trend can be observed, with `OF-SS4` achieving a higher value for *Small*, but lower values for *Full* and *One-Shot*.

As another reference, we include the weakly-supervised method by Maman and Bermano (MB) [11], which also relies on the Onsets and Frames transcription model [7, 16] but benefits from training on much more data and across various instrumentations. Our method does not reach the performance of MB in any scenario, but the performance gap is reasonably small given the difference in amount of training data, e. g., a note F1 score of 85.0 for `OF-SS4` in scenario *Small* compared to 89.7 for MB on MAESTRO.

4.6 Ablation Study

We perform an ablation study to evaluate the efficacy of the individual components of our semi-supervised method. The results of this study are shown in Table 2. The method `OF-SS1` performs pseudo-labeling without consistency regularization and distribution matching, where the performance metrics indicate potential model collapse. Better results are achieved when additionally using either distribution matching (`OF-SS2`) or consistency regularization (`OF-SS3`), achieving already better note F1 scores than the supervised baseline `OF`. The performance is further improved by combining both techniques, which results in our proposed method `OF-SS4`. The remaining ablations change the hyperparameter setting of our method, where less restrictive thresholds for selecting pseudo-labels (`OF-SS5`), calculating the unsupervised loss also for offsets (`OF-SS6`), or a reduced overall weight of the unsupervised loss (`OF-SS7`) yield worse results.

5. CONCLUSION

In this paper, we successfully transferred SSL techniques from CV to the MIR domain. More specifically, we applied pseudo-labeling, consistency regularization, and distribution matching for the task of APT, enabling the option to leverage unlabeled data during training. Thereby, the dependence on large annotated datasets is considerably reduced. For instance, using our semi-supervised approach, we observed reductions in the required amount of labeled data by up to 80% for achieving similar performance as a purely supervised baseline.

In future work, we plan to investigate other augmentation strategies, e. g., musically meaningful augmentations as in [18], to perform consistency regularization, and the extension of the method to the multi-instrument setting.

Acknowledgements: This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Grant No.350953655 (MU 2686/11-2) and Grant No.500643750 (MU 2686/15-1). The authors are with the International Audio Laboratories Erlangen, a joint institution of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) and Fraunhofer Institute for Integrated Circuits IIS.

6. REFERENCES

- [1] C. Raphael, “Automatic transcription of piano music,” in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2002, pp. 15–19.
- [2] E. Benetos, S. Dixon, Z. Duan, and S. Ewert, “Automatic music transcription: An overview,” *IEEE Signal Processing Magazine*, vol. 36, no. 1, pp. 20–30, 2019.
- [3] M. Müller, *Fundamentals of Music Processing – Using Python and Jupyter Notebooks*, 2nd ed. Springer Verlag, 2021.
- [4] P. Smaragdis and J. C. Brown, “Non-negative matrix factorization for polyphonic music transcription,” in *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2003, pp. 177–180.
- [5] E. Vincent, N. Bertin, and R. Badeau, “Adaptive harmonic spectral decomposition for multiple pitch estimation,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 3, pp. 528–537, 2010.
- [6] S. Sigtia, E. Benetos, and S. Dixon, “An end-to-end neural network for polyphonic piano music transcription,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 5, pp. 927–939, 2016.
- [7] C. Hawthorne, E. Elsen, J. Song, A. Roberts, I. Simon, C. Raffel, J. H. Engel, S. Oore, and D. Eck, “Onsets and frames: Dual-objective piano transcription,” in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Paris, France, 2018, pp. 50–57.
- [8] Q. Kong, B. Li, X. Song, Y. Wan, and Y. Wang, “High-resolution piano transcription with pedals by regressing onset and offset times,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3707–3717, 2021.
- [9] C. Hawthorne, I. Simon, R. Swavely, E. Manilow, and J. H. Engel, “Sequence-to-sequence piano transcription with transformers,” in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Online, 2021, pp. 246–253.
- [10] J. Gardner, I. Simon, E. Manilow, C. Hawthorne, and J. H. Engel, “MT3: Multi-task multitrack music transcription,” in *Proceedings of the International Conference on Learning Representations (ICLR)*, Virtual, 2022.
- [11] B. Maman and A. H. Bermanno, “Unaligned supervision for automatic music transcription in the wild,” in *Proceedings of the International Conference on Machine Learning (ICML)*, Baltimore, Maryland, USA, 2022, pp. 14918–14934.
- [12] K. Toyama, T. Akama, Y. Ikemiya, Y. Takida, W. Liao, and Y. Mitsufuji, “Automatic piano transcription with hierarchical frequency-time transformer,” in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Milano, Italy, 2023, pp. 215–222.
- [13] W. T. Lu, J. Wang, and Y. Hung, “Multitrack music transcription with a time-frequency perceiver,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Rhodes Island, Greece, 2023.
- [14] C. Weiß, F. Zalkow, V. Arifi-Müller, M. Müller, H. V. Koops, A. Volk, and H. Grohgan, “Schubert Winterreise dataset: A multimodal scenario for music analysis,” *ACM Journal on Computing and Cultural Heritage (JOCCH)*, vol. 14, no. 2, pp. 25:1–18, 2021.
- [15] V. Emiya, R. Badeau, and B. David, “Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 6, pp. 1643–1654, 2010.
- [16] C. Hawthorne, A. Stasyuk, A. Roberts, I. Simon, C. A. Huang, S. Dieleman, E. Elsen, J. H. Engel, and D. Eck, “Enabling factorized piano music modeling and generation with the MAESTRO dataset,” in *Proceedings of the International Conference on Learning Representations (ICLR)*, New Orleans, Louisiana, USA, 2019.
- [17] M. Müller, V. Konz, W. Bogler, and V. Arifi-Müller, “Saarland music data (SMD),” in *Demos and Late Breaking News of the International Society for Music Information Retrieval Conference (ISMIR)*, Miami, Florida, USA, 2011.
- [18] D. Edwards, S. Dixon, E. Benetos, A. Maezawa, and Y. Kusaka, “A data-driven analysis of robust automatic piano transcription,” *IEEE Signal Processing Letters*, vol. 31, pp. 681–685, 2024.
- [19] K. W. Cheuk, D. Herremans, and L. Su, “Reconvat: A semi-supervised automatic music transcription framework for low-resource real-world data,” in *Proceedings of the ACM Multimedia Conference*, Virtual Event, China, 2021, pp. 3918–3926.

- [20] X. Yang, Z. Song, I. King, and Z. Xu, "A survey on deep semi-supervised learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 9, pp. 8934–8954, 2023.
- [21] A. Peláez-Vegas, P. Mesejo, and J. Luengo, "A survey on semi-supervised semantic segmentation," *CoRR*, vol. abs/2302.09899, 2023.
- [22] K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C. Raffel, E. D. Cubuk, A. Kurakin, and C. Li, "Fix-Match: Simplifying semi-supervised learning with consistency and confidence," in *Advances in Neural Information Processing Systems (NeurIPS)*, Virtual, 2020.
- [23] D. Berthelot, N. Carlini, E. D. Cubuk, A. Kurakin, K. Sohn, H. Zhang, and C. Raffel, "ReMixMatch: Semi-supervised learning with distribution matching and augmentation anchoring," in *Proceedings of the International Conference on Learning Representations (ICLR)*, Virtual, 2020.
- [24] D. S. Park, W. Chan, Y. Zhang, C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A simple data augmentation method for automatic speech recognition," in *Proceedings of the Annual Conference of the International Speech Communication Association (Interspeech)*, Graz, Austria, 2019, pp. 2613–2617.
- [25] L. Ou, Z. Guo, E. Benetos, J. Han, and Y. Wang, "Exploring transformer's potential on automatic piano transcription," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Virtual and Singapore, 2022, pp. 776–780.
- [26] K. W. Cheuk, R. Sawata, T. Uesaka, N. Murata, N. Takahashi, S. Takahashi, D. Herremans, and Y. Mitsufoji, "Diffroll: Diffusion-based generative music transcription with unsupervised pretraining capability," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Rhodes Island, Greece, 2023.
- [27] M. Krause, S. Strahl, and M. Müller, "Weakly supervised multi-pitch estimation using cross-version alignment," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Milano, Italy, 2023.
- [28] T. Miyato, S. Maeda, M. Koyama, and S. Ishii, "Virtual adversarial training: A regularization method for supervised and semi-supervised learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 8, pp. 1979–1993, 2019.
- [29] F. Cwitkowitz and Z. Duan, "Toward fully self-supervised multi-pitch estimation," *CoRR*, vol. abs/2402.15569, 2024.
- [30] O. Chapelle, B. Schölkopf, and A. Zien, Eds., *Semi-Supervised Learning*. The MIT Press, 2006.
- [31] J. E. van Engelen and H. H. Hoos, "A survey on semi-supervised learning," *Machine Learning*, vol. 109, no. 2, pp. 373–440, 2020.
- [32] J. M. Duarte and L. Berton, "A review of semi-supervised learning for text classification," *Artificial Intelligence Review*, vol. 56, no. 9, pp. 9401–9469, 2023.
- [33] S. Kum, J. Lin, L. Su, and J. Nam, "Semi-supervised learning using teacher-student models for vocal melody extraction," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Montréal, Canada, 2020, pp. 93–100.
- [34] M. Won, K. Choi, and X. Serra, "Semi-supervised music tagging transformer," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Online, 2021, pp. 769–776.
- [35] Y. Ouali, C. Hudelot, and M. Tami, "An overview of deep semi-supervised learning," *CoRR*, vol. abs/2006.05278, 2020.
- [36] D.-H. Lee, "Pseudo-Label: The simple and efficient semi-supervised learning method for deep neural networks," in *Proceedings of the ICML 2013 Workshop on Challenges in Representation Learning (WREPL)*, Atlanta, GA, USA, 2013.
- [37] S. Laine and T. Aila, "Temporal ensembling for semi-supervised learning," in *Proceedings of the International Conference on Learning Representations (ICLR)*, Toulon, France, 2017.
- [38] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *Advances in Neural Information Processing Systems (NeurIPS)*, Long Beach, CA, USA, 2017, pp. 1195–1204.
- [39] T. Devries and G. W. Taylor, "Improved regularization of convolutional neural networks with cutout," *CoRR*, vol. abs/1708.04552, 2017.
- [40] S. Grollmisch and E. Cano, "Improving semi-supervised learning for audio classification with Fix-Match," *Electronics*, vol. 10, no. 15, 2021.
- [41] L. Cances, E. Labbé, and T. Pellegrini, "Comparison of semi-supervised deep learning algorithms for audio classification," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2022, no. 1, p. 23, 2022.
- [42] M. M. i Rabadán, A. Pieropan, H. Azizpour, and A. Maki, "Dense FixMatch: A simple semi-supervised learning method for pixel-wise prediction tasks," in *Proceedings of the Northern Lights Deep Learning (NLDL) Workshop*, Tromsø, Norway, 2023.
- [43] B. Krawczyk, "Learning from imbalanced data: Open challenges and future directions," *Progress in Artificial Intelligence*, vol. 5, no. 4, pp. 221–232, 2016.

- [44] E. Arazo, D. Ortego, P. Albert, N. E. O'Connor, and K. McGuinness, "Pseudo-labeling and confirmation bias in deep semi-supervised learning," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*. Glasgow, United Kingdom: IEEE, 2020.
- [45] J. Kim, Y. Hur, S. Park, E. Yang, S. J. Hwang, and J. Shin, "Distribution aligning refinery of pseudo-label for imbalanced semi-supervised learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, Virtual, 2020.
- [46] M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *Neural Networks*, vol. 106, pp. 249–259, 2018.
- [47] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the International Conference for Learning Representations (ICLR)*, San Diego, California, USA, 2015.
- [48] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, and D. P. W. Ellis, "MIR_EVAL: A transparent implementation of common MIR metrics," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Taipei, Taiwan, 2014, pp. 367–372.
- [49] Y. Wu, B. Chen, and L. Su, "Polyphonic music transcription with semantic segmentation," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Brighton, UK, 2019, pp. 166–170.

NOTE-LEVEL TRANSCRIPTION OF CHORAL MUSIC

Huiran Yu

University of Rochester
hyu56@ur.rochester.edu

Zhiyao Duan

University of Rochester
zhiyao.duan@rochester.edu

ABSTRACT

Choral music is a musical activity with one of the largest participant bases, yet it has drawn little attention from automatic music transcription research. The main reasons we argue are due to the lack of data and technical difficulties arise from diverse acoustic conditions and unique properties of choral singing. To address these challenges, in this paper we propose a Transformer-based framework for note-level transcription of choral music. This framework bypasses the frame-level processing and directly produces a sequence of notes with associated timestamps. We also introduce YouChorale, a novel choral music dataset in a cappella setting curated from the Internet. YouChorale contains 452 real-world recordings in diverse acoustic configurations of choral music from over 100 composers as well as their MIDI scores. Trained on YouChorale, our proposed model achieves state-of-the-art performance in choral music transcription, marking a significant advancement in the field.

1. INTRODUCTION

Choral singing stands as one of the most widely engaged forms of musical expression, uniting voices in harmony across cultures and communities. Despite its profound presence in the musical landscape, choral singing has notably been overlooked in the field of Automatic Music Transcription (AMT), a domain predominantly oriented towards instrumental music [1–3], leaving choral singing with scant attention and few dedicated studies [4, 5]. This oversight not only highlights a gap in AMT research but also underscores the potential for significant advancements in the transcription of choral music, an area waiting for exploration and innovation.

The transcription of choral music introduces unique challenges compared with its instrumental counterparts. One of the main characteristics of choral singing is the soft onset of notes and smooth transitions between notes, resulting in indistinct boundaries and complicating the determination of note onsets. Additionally, the complex acoustic environment enriches choral music performances with reverberation, further complicates transcrip-

tion efforts. These factors combined present a formidable challenge in accurately capturing the note occurrences in choral music recordings, necessitating novel approaches to AMT that can handle these specific challenges.

Recent methodologies in AMT fall primarily into two categories: Onsets and Frames [1], which estimates frame-level pitch activation informed by note onset predictions, and then combines such results to note estimates; and the use of models like MT3 [2], which conceptualize transcription as a token prediction task. However, both approaches exhibit limitations in addressing the soft onset characteristic of choral singing. Onset and frame detection methods heavily rely on the successful identification of note onsets, a task made difficult by the blurry beginning of vocal notes. Conversely, models like MT3 predict notes as a series of tokens, which can complicate the aggregation of information pertaining to individual notes, thereby obscuring the cohesive representation of choral music.

Another critical hurdle in advancing choral music transcription is the availability of comprehensive and high-quality datasets. Existing resources include the Dagstuhl ChoirSet [6], which offers less than one hour of high-quality recording of two pieces and a set of systematic exercises. The Erkomaishvili Dataset [7] provides around seven hours of recordings, but the sound quality is poor for model training. The Bach Chorale¹ and Barbershop Quartet² datasets provide tracked recordings, but the music genre is limited in these datasets. Also, they only involve a small group of singers and a fixed recording environment. This dearth of datasets impedes field progress and highlights the need for more robust and accessible resources for choral music transcription.

In response to these challenges, this paper proposes a novel note-level transcription architecture inspired by advancements in object detection and sound event detection. Instead of predicting the frame-level activation or separated MIDI-like events, this model directly decodes the pitch, onset, and duration from a hidden embedding of each note. To take care of the sequential relationships between the notes, we integrate the Transformer model as the backbone of our network, leveraging its proven efficacy in capturing long-term dependencies. Experiment results show that this model has largely improved the frame-level recall of the transcription output, indicating that the proposed model makes better use of the entire process of note articulation. The proposed model has also shown



¹ <https://www.pgmusic.com/bachchorales.htm>

² <http://www.pgmusic.com/barbershopquartet.htm>

robustness against the distortion caused by reverberation in the recordings. To address the critical gap in available resources, we have curated a comprehensive dataset for choral music transcription, comprising 496 real-world recordings across a diverse array of acoustic environments and featuring compositions from over 100 composers, accompanied by their corresponding MIDI scores. This dataset not only facilitates the development of our proposed model but also provides a valuable resource for future research in choral music transcription. Through this work, we aim to bridge the existing gap in AMT research, offering novel insights and methodologies that enhance our understanding and capabilities in transcribing choral music.

The structure of this paper is as follows: Section 2 covers the related works of this study; Section 3 describes the transcription architecture we proposed for the choral singing task; Section 4 introduces the YouChorale dataset, the experimental settings and the results; finally, Section 5 concludes the paper.

2. RELATED WORK

Automatic Music Transcription (AMT) has been a largely investigated task in Music Information Retrieval (MIR), and people have proposed various methods to address this problem. Onsets and Frames [1] represents the start of a group of methods that uses Convolutional Neural Networks (CNN) to extract the onset activation and frame activation in the spectrogram based on which a final note prediction output is aggregated through post-processing. Many other methods have inherited this idea, and several methods have been proposed for piano [8] and multi-instrument transcription [3, 9]. To fully use the activation detection and produce holistic transcription results, Yan et al. [10] proposed a neural semi-CRF-based method that predicts the best interval combinations of the frame-level estimations.

Another choice is to use sequence-to-sequence models that transcribe tokens describing different aspects of the notes, such as note-on and note-off events, velocity, and time stamps [11]. MT3 [2] expanded this method to multi-instrument transcription, and Simon et al. [12] further augmented the training data of such model by mixing monophonic recordings. There also exist methods that use generative diffusion models [13] to perform transcription. However, the performance of this method is still not comparable with other works.

For choral music transcription, Schramm et al. [4] proposed a spectrogram factorization method to transcribe a cappella performances. McLeod et al. [5] proposed using extended probabilistic latent component analysis and music language model to improve the performance further. There is also literature on score transcription of choral music [14], but they focus on producing the music score instead of the precise physical timing of each note in the recordings.

We can view automatic music transcription as a special form of sound event detection, which aims to identify the

note entities in the audio recordings. The strong timing correlations between the notes drive us to detection methods with sequential modeling abilities. Carion et al. [15] proposed an end-to-end object detection architecture with Transformers, which uses the Transformer encoder and decoder to attend to the input image to detect sound events and their corresponding bounding boxes. Such an idea is also adapted in sound event detection, represented by works from Kong et al. [16].

3. METHOD

We demonstrate the architecture of the model in Figure 1. The input mel-spectrogram first goes through a pre-filtering CNN network. After adding the positional encoding, two multi-head attention and feed-forward encoder layers further aggregate information in the spectrogram. The processed spectrogram then goes into the transformer decoder to auto-regressively generate an array of note embeddings. Finally, we employ three Multi-Layer Perceptron (MLP) modules as the feed-forward network to predict the MIDI pitch, onset time, and duration from the embedding of each note.

Inspired by the Transformer-based object detection methods and sound event detection methods, we regard the onset and offset as the “bounding box” of each note. Like pitch, they are the note’s built-in attributes. Since Transformer models are well-known for their capability of learning long-term dependencies, we here let the encoder and decoder layers fully take care of the aggregation of information of each note to achieve end-to-end music transcription.

The model’s input is a batch of segmented spectrograms with the shape of (B, L, M) , where B is the batch size, L is the length of the segment, and M is the number of frequency bins. The model’s output is three parallel arrays of pitch, onset time, and duration, with the shapes of (B, N, K) , $(B, N, 1)$, $(B, N, 1)$, respectively. N is the length of the transcribed note sequence, and K is the number of possible pitch entries. To ensure the one-dimensional note sequence is unique for a polyphonic score, we serialize the notes first in chronological order from earliest to latest and second in pitch order from highest to lowest.

3.1 CNN Preprocessing

We use two layers of the 1D-Convolutional Neural Network (CNN) to preprocess the input mel-spectrogram. The network has a kernel size of 9 and is activated with the ReLU function, creating a receptive field of around 300 ms at each output frame. After the CNN, the shape of the output is (B, L, C) , where C is the size of the hidden dimension. Then, we add the output with positional encoding and feed it into the encoder layers.

3.2 Encoder and Decoder Layers

We inherit the encoder and decoder design in the original Transformer [17], which includes multi-head attention

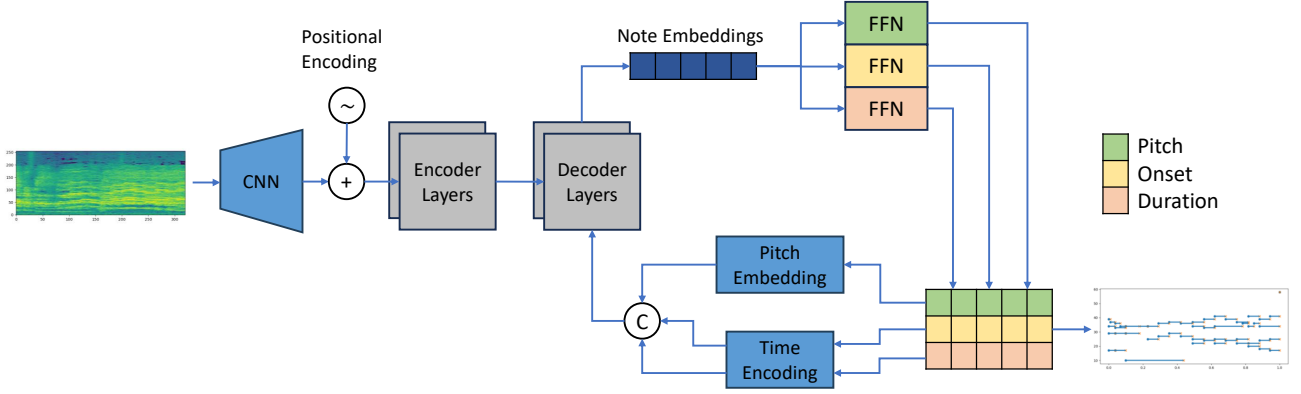


Figure 1. The overall architecture of the transcription model.

blocks and feed-forward layers. The encoder is conducting self-attention with the CNN-processed spectrogram; the decoder also attends to the output of the encoder after a self-attention layer. In our model, we use two layers of encoder layers and decoder layers.

During inference, the decoder performs auto-regressive decoding of the final note sequence. In model implementation, we normalize the time within one segment to $[0, 1]$ and calculate the onset time t_o and duration t_d accordingly to reduce the difficulties in training.

3.3 Positional Encoding

After the mel-spectrogram goes through the CNN filter banks, it will be added to a positional encoding to let the encoder layers learn the sequential relationship between the frames. For the positional encoding before the encoder layers, we adopt the original design in Transformer [17]. Given the frame from the processed spectrogram at pos out of the L possible positions and denote the dimensionality of the Transformer as d_{model} , we define the positional encoding PE at the $2i$ and $2i + 1$ dimension as

$$PE_{(pos,2i)} = \sin(pos/10000^{2i/d_{model}}), \quad (1)$$

$$PE_{(pos,2i+1)} = \cos(pos/10000^{2i/d_{model}}). \quad (2)$$

When we encode the continuous onset time and duration as the input of the decoder layers, we would like to align the decoder time encoding with the encoder’s positional encoding. An onset time at t_o will have the time embedding TE identical to the PE of the corresponding frame position:

$$TE_{(t_o,i)} = PE_{(t_o \times L, i)}. \quad (3)$$

Similarly, the duration of the note t_d is encoded with the same equation, replacing t_o with t_d in Equation (3). In this way, we can align the time in the spectrogram and the onset prediction of the model, which will help the model better find the relationship between the frames in the spectrogram and the time in the final transcription result. After we get the pitch embeddings and time encodings of the previously generated notes, we concatenate them together and send them into the decoder layers.

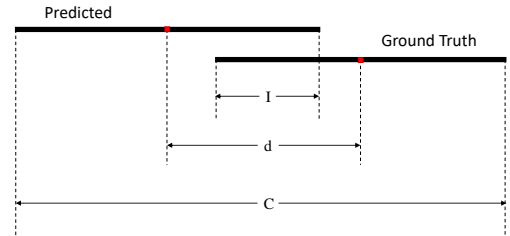


Figure 2. A demonstration of DIoU calculation.

3.4 Training Objectives

We optimize the loss of pitch estimation and timing estimation. For pitch estimation, we use the cross entropy loss \mathcal{L}_p . For time estimation, we first apply the L1 loss:

$$\mathcal{L}_{time} = \sum_{i=1}^N \|\hat{t}_o^{(i)} - t_o^{(i)}\|_1 + \sum_{i=1}^N \|\hat{t}_d^{(i)} - t_d^{(i)}\|_1, \quad (4)$$

where t_o is the ground-truth onset time, \hat{t}_o is the predicted onset time; t_d is the ground-truth duration, \hat{t}_d is the predicted duration.

We also adapt the DIoU (Distance-IoU, Intersection over Union) loss [18] from object detection to 1D scenario, as in Figure 2:

$$\mathcal{L}_{DIoU} = 1 - IoU + \frac{d^2}{C^2}. \quad (5)$$

Here, IoU is the ratio between the intersection and the union of the predicted time span and the ground-truth time span; d is the distance between the center of the prediction and the ground truth time span to add more penalties to the far away predictions; C is the length of the minimum bounding box that can cover both prediction and ground truth. Note that when there is an overlap between prediction and ground truth, $union = C$; when there is no overlapping between them, $IoU = 0$, we define union as the summation of the length of the two segments.

In the experiments, we trained two models with \mathcal{L}_{time} and \mathcal{L}_{DIoU} respectively and tested their performances.

	Singers in Each Part		Reverberation Time			Number of Parts						
	≤ 3	> 3	long	medium	short	2~3	4	5	6	7	8	≥ 9
Train	89	303	57	283	52	17	218	54	46	5	39	13
Validation	9	21	3	25	2	0	11	6	1	1	9	1
Test	10	20	5	23	2	0	11	9	4	0	5	1
Total	108	344	65	331	56	17	240	69	51	6	53	15

Table 1. Statistics of the YouChorale dataset.

4. EXPERIMENT

In this section, we describe experiments that evaluate our models against baselines.

4.1 YouChorale Dataset

In an effort to address the scarcity of resources for choral music transcription, we curated a dataset, YouChorale, from YouTube and a variety of MIDI archive sources^{3 4 5}, focusing exclusively on a cappella choral singing. With a total length of 22 hours 25 minutes, the YouChorale dataset contains 452 recordings of 261 compositions from 118 composers, representing a wide range of historical periods, styles, and complexities inherent to choral music. We have made the dataset publicly available at <https://github.com/ella-granger/YouChorale>, enriching a comprehensive resource of choral music for further exploration and development in the field of Automatic Music Transcription (AMT).

We split the dataset into train, validation and test set by the ratio of 392:30:30. To ensure the intonation of the evaluation recordings, we only selected performances by well-known choirs into the validation and the test sets. The detailed statistics of the dataset are shown in Table 1. The metric “singers in each part” indicates whether the performance is from a small a cappella group (less than or equals to three singers per part) or a larger ensemble (more than three singers per part). “Reverberation time” is an indication of the acoustic environment and how the signal is blurred or distorted. “Number of parts” indicates the complexity of the piece. Most of the pieces contain four to six parts, for example SATB or SSATTB, but there are also extreme cases where over nine parts appear in one composition.

We are also providing an aligned version of MIDI file along with the recordings. The alignment is achieved through the following steps: First, we adjust the key signature of the MIDI files to match the recording. Next, we render the waveform of the MIDI notation and align the Constant-Q Transform [19] feature of the synthesized audio and the performance recording by the soft-DTW algorithm [20]. Finally, we smooth the alignment curve to remove abrupt tempo changes in the aligned MIDI.

³ www.learnchoralmusic.co.uk

⁴ gasilvis.net

⁵ <http://www.maennerchor-sg.ch/midi/>

4.2 BachChorale Dataset

The accurately labeled BachChorale dataset also serves as a benchmark for evaluating the transcription performance of our model and the baselines. This two-volume dataset contains 53 four-part choral compositions by J.S. Bach, with a total length of two hours. Note that during the collection of YouChorale, all the Bach pieces we found are accompanied by organ or orchestra. Therefore, we excluded Bach pieces to keep the dataset in a cappella settings, which also means that using BachChorale as a test dataset does not introduce label leakage.

4.3 Training Settings

For the training data, we downsampled the audio to 16 kHz, and extracted the mel-spectrogram with $N_{FFT} = 2048$, hop length = 256, and number of frequency bins $M = 256$. The length of each segment $L = 320$, which corresponds to 5.12 seconds of audio. The hidden dimension of the model $C = 256$. During training, we set the batch size = 8, and use an Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.98$ and $\epsilon = 10^{-9}$. The warmup step is set to 12000. We use teacher forcing during the training phase, which provides the ground truth notes as the context and lets the model predict only the next note. We have released our code at <https://github.com/ella-granger/NoteTranscription>.

4.4 Results

We choose Schramm et al. [4], Onsets and Frames [1] and MT3 [2] as our baselines. For Schramm et al. [4], we list their reported frame-level multi-pitch estimation result which was also evaluated on the BachChorale dataset. For Onsets and Frames, we train a new model with the YouChorale training set from scratch; for MT3, we use the provided multi-instrument checkpoint. For the Onsets and Frames, MT3, and the proposed model, we evaluate the transcription result after they produce the final MIDI notes output.

We evaluated the frame-level activation detection and the note onset detection with a tolerance of 50 ms and 100 ms, respectively. The results on the BachChorale dataset are shown in Table 2, and the results on the YouChorale test set are shown in Table 3. We can see that compared with Onsets and Frames or MT3, our proposed model has a more balanced performance on precision and recall at the frame level, and produces the highest f1 score among the

Model	Frame			Onset (50ms)			Onset (100ms)		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Schramm et al. [4]	0.713	0.709	0.710	-	-	-	-	-	-
Onsets and Frames [1]	0.832	0.440	0.571	0.411	0.130	0.196	0.730	0.231	0.348
MT3 [2]	0.645	0.411	0.502	0.117	0.249	0.157	0.201	0.426	0.269
Proposed- \mathcal{L}_{time}	0.663	0.616	0.639	0.162	0.225	0.185	0.263	0.368	0.301
Proposed- \mathcal{L}_{DIOU}	0.611	0.639	0.624	0.189	0.182	0.183	0.284	0.274	0.275

Table 2. Model performances on BachChorale dataset.

Model	Frame			Onset (50ms)			Onset (100ms)		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Onsets and Frames [1]	0.806	0.326	0.428	0.450	0.178	0.242	0.688	0.248	0.344
MT3 [2]	0.590	0.243	0.344	0.117	0.148	0.127	0.200	0.255	0.217
Proposed- \mathcal{L}_{time}	0.670	0.596	0.631	0.181	0.221	0.192	0.284	0.339	0.299
Proposed- \mathcal{L}_{DIOU}	0.630	0.658	0.644	0.210	0.209	0.203	0.309	0.304	0.297

Table 3. Model performances on YouChorale test set.

Model	Frame			Δ F1	Onset (50ms)			Δ F1
	Precision	Recall	F1		Precision	Recall	F1	
Onsets and Frames [1]	0.604	0.313	0.406	-0.165 (-28.9%)	0.126	0.094	0.107	-0.089 (-45.4%)
MT3 [2]	0.553	0.398	0.463	-0.039 (-7.8%)	0.022	0.040	0.028	-0.129 (-82.2%)
Proposed- \mathcal{L}_{time}	0.518	0.518	0.518	-0.121 (-18.9%)	0.089	0.180	0.114	-0.069 (-37.7%)

Table 4. Model performance under reverb distortion on BachChorale dataset.

deep learning methods. Although the Onsets and Frames model still reaches a higher precision value on the onset time of the note, the significantly higher recall of our model at the frame level indicates that it places greater emphasis on the entire process of note articulation, not just the onset and offset of the notes, which achieves our goal with holistic note transcription. The deep-learning methods still have some room for improvement towards Schramm et al. [4] on the BachChorale dataset, however, since the dataset only have one singer for each part, Schramm et al. might have some advantage as it was trained on solo singing.

We would also like to compare the performance of the two loss function \mathcal{L}_{time} and \mathcal{L}_{DIOU} . From the results we can see that the \mathcal{L}_{time} trained model tends to have high precision and low recall at frame level and low precision and high recall on note onsets, while the \mathcal{L}_{DIOU} trained model has the opposite behavior. It indicates that the \mathcal{L}_{time} trained model usually extracts shorter fragments of the notes and the \mathcal{L}_{DIOU} trained model longer full notes. This is due to the property of the two loss functions: The L_1 based time loss function focuses more on the absolute distance between the boundary of the predicted notes and the ground-truth notes, while the L_{DIOU} based loss function puts more emphasis on the overall intersection of the prediction and ground truth, and will have the boundaries not as precise as the L_1 loss.

4.5 Performance Under Reverb Distortion

In real-world choral music performances, reverberation is an unignorable part of acoustic effects. For example, concert halls create reverb with a long reberveration time, which introduces distortions into the spectrogram. To evaluate the resilience of our model against common distortions encountered in live settings, we apply an artificial reverb⁶ to our test set to simulate the complex acoustic environmental characteristic of real-world choral performances. The performance of each model is shown in Table 4. The findings indicate that our proposed model still holds a relatively high performance, and the proposed model together with Onset and Frames trained on the YouChorale dataset, retains some ability to predict onset timing while the MT3 model nearly failed to predict any reasonable onset of the notes. This resilience highlights the importance of incorporating diverse, real-world data in training AMT models, ensuring their applicability and effectiveness in practical, everyday transcription scenarios.

Through cautious dataset curation and strategic model design, the experiments have shown our proposed model’s capabilities in the realm of choral music transcription. By directly addressing the nuanced challenges of this genre,

⁶ <https://ccrma.stanford.edu/jos/pasp/Freeverb.html>. The roomsize is set to one.

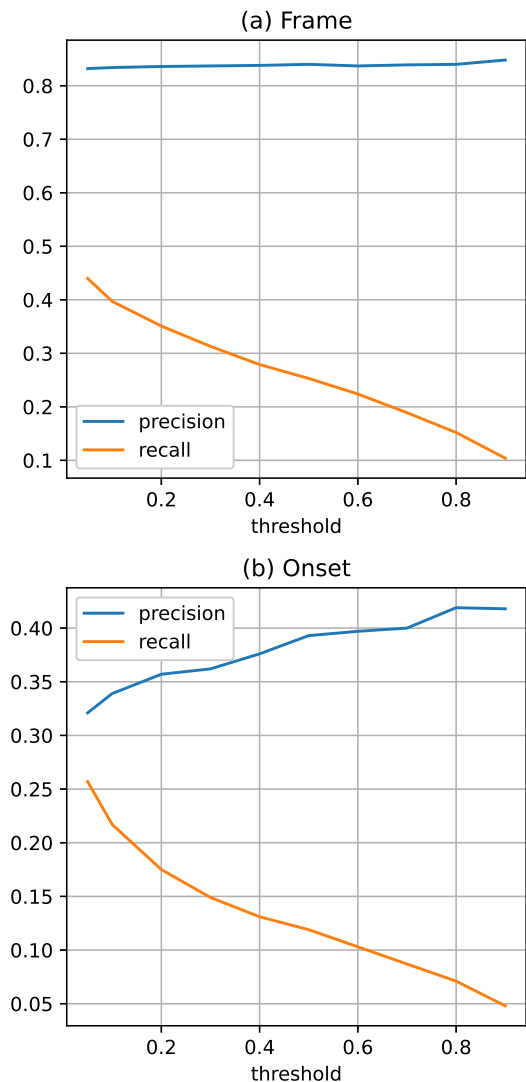


Figure 3. The precision/recall v.s. threshold curves of frame-level and note-level (onset) transcription from Onsets and Frames.

from soft note onsets to complex acoustic environments, we not only advance the state of AMT but also pave the way for future innovations in the transcription of polyphonic vocal music.

4.6 Limitations of Onsets and Frames Model

If we take a closer look at the result in Table 5, we may find that for Onsets and Frames model, there is a big gap between the frame-level precision and recall. After extracting the frame activation before post-processing and calculating its objectives, we get the result in Table 5. We can see that although post-processing improves prediction precision, it discards a large amount of true-positive frame activations.

Since the Onsets and Frames model will not transcribe any new note until it finds a new onset, the model’s capability of correctly predicting the onset significantly affects the overall performance. Figure 3 shows the precision and recall curve of frame and onset prediction with respect to

Model	Precision	Recall	F1
Onsets and Frames [1]	0.851	0.267	0.400
O&F (frame activation)	0.801	0.632	0.704

Table 5. Comparison between the final transcription result and frame-level activation of Onsets and Frames on Bach-Chorale dataset.

the onset decision threshold. The curves are unbalanced, and the reported result in Table 2 is at the threshold value of 0.05, which means we almost extract all the possible onsets as long as there is a trace amount of activation. All the evidence shows that the limitation of putting too much attention to onsets becomes especially pronounced in the context of choral music, where soft onsets and smooth transitions are prevalent. Instead, we should leverage the information contained in the frame activation and view each note as a whole, and let the model decide where to locate the notes, which is the design principle of our proposed method.

5. CONCLUSIONS

We proposed a novel transcription model architecture for choral music, which conducts holistic note transcription, addressing the soft onset and complex acoustic environment issues. We also introduced a newly curated a cappella dataset for the development of automatic music transcription. Tested on the BachChorale dataset, our model has shown competent performance on the choral music transcription task, particularly in its robustness against reverb. By addressing the noted limitations of existing models and contributing a valuable dataset to the research community, our work paves the way for future innovations in AMT, enhancing the accessibility and understanding of choral music through technology.

The next step of this work would be to distinguish and stream different parts in choral singing, and explore the potential of model architecture to general music transcription tasks.

6. ACKNOWLEDGEMENT

This work is supported in part by National Science Foundation (NSF) grants 1846184 and 2222129 and synergistic activities funded by NSF grant DGE-1922591.

7. REFERENCES

- [1] C. Hawthorne, E. Elsen, J. Song, A. Roberts, I. Simon, C. Raffel, J. Engel, S. Oore, and D. Eck, “Onsets and Frames: Dual-Objective Piano Transcription,” in *Proceedings of the 19th International Society for Music Information Retrieval Conference*. ISMIR, Nov. 2018, pp. 50–57.
- [2] J. P. Gardner, I. Simon, E. Manilow, C. Hawthorne, and J. Engel, “MT3: Multi-Task Multitrack Music Tran-

- scription,” in *International Conference on Learning Representations*. ICLR, 2021.
- [3] B. Maman and A. H. Bermann, “Unaligned Supervision for Automatic Music Transcription in The Wild,” in *Proceedings of the 39th International Conference on Machine Learning*, vol. 162. PMLR, 17–23 Jul 2022, pp. 14 918–14 934.
- [4] R. Schramm and E. Benetos, “Automatic Transcription of a Cappella recordings from Multiple Singers,” in *Audio Engineering Society Conference: 2017 AES International Conference on Semantic Audio*. Audio Engineering Society, 2017.
- [5] A. McLeod, R. Schramm, M. Steedman, and E. Benetos, “Automatic transcription of polyphonic vocal music,” *Applied Sciences*, vol. 7, no. 12, p. 1285, 2017.
- [6] S. Rosenzweig, H. Cuesta, C. Weiß, F. Scherbaum, E. Gómez, and M. Müller, “Dagstuhl ChoirSet: A Multitrack Dataset for MIR Research on Choral Singing,” *Transactions of the International Society for Music Information Retrieval*, Jul 2020.
- [7] S. Rosenzweig, F. Scherbaum, D. Shugliashvili, V. Arifi-Müller, and M. Müller, “Erkomaishvili Dataset: A Curated Corpus of Traditional Georgian Vocal Music for Computational Musicology,” *Transactions of the International Society for Music Information Retrieval*, Apr 2020.
- [8] Q. Kong, B. Li, X. Song, Y. Wan, and Y. Wang, “High-resolution piano transcription with pedals by regressing onset and offset times,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3707–3717, 2021.
- [9] K. W. Cheuk, D. Herremans, and L. Su, “Reconvat: A semi-supervised automatic music transcription framework for low-resource real-world data,” in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 3918–3926.
- [10] Y. Yan, F. Cwitkowitz, and Z. Duan, “Skipping the Frame-Level: Event-Based Piano Transcription With Neural Semi-CRFs,” in *Advances in Neural Information Processing Systems*, vol. 34. Curran Associates, Inc., 2021, pp. 20 583–20 595.
- [11] C. Hawthorne, I. Simon, R. Swavely, E. Manilow, and J. Engel, “Sequence-To-Sequence Piano Transcription With Transformers,” in *Proceedings of the 22nd International Society for Music Information Retrieval Conference*. ISMIR, 2021, pp. 246–253.
- [12] I. Simon, J. Gardner, C. Hawthorne, E. Manilow, and J. Engel, “Scaling Polyphonic Transcription with Mixtures of Monophonic Transcriptions,” in *Proceedings of the 23rd International Society for Music Information Retrieval Conference*. ISMIR, 2022, pp. 44–51.
- [13] K. W. Cheuk, R. Sawata, T. Uesaka, N. Murata, N. Takahashi, S. Takahashi, D. Herremans, and Y. Mitsufuji, “Diffroll: Diffusion-Based Generative Music Transcription with Unsupervised Pretraining Capability,” in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.
- [14] M. A. Román, A. Pertusa, and J. Calvo-Zaragoza, “A Holistic Approach to Polyphonic Music Transcription With Neural Networks,” in *Proceedings of the 20th International Society for Music Information Retrieval Conference*. ISMIR, 2019, pp. 731–737.
- [15] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, “End-to-end object detection with transformers,” in *European conference on computer vision*. Springer, 2020, pp. 213–229.
- [16] Q. Kong, Y. Xu, W. Wang, and M. D. Plumbley, “Sound Event Detection of Weakly Labelled Data With CNN-Transformer and Automatic Threshold Optimization,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2450–2460, 2020.
- [17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention Is All You Need,” *Advances in neural information processing systems*, vol. 30, 2017.
- [18] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, “Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, pp. 12 993–13 000, Apr. 2020.
- [19] J. C. Brown, “Calculation of a constant q spectral transform,” *The Journal of the Acoustical Society of America*, vol. 89, no. 1, pp. 425–434, 1991.
- [20] M. Cuturi and M. Blondel, “Soft-DTW: A Differentiable Loss Function for Time-Series,” in *International conference on machine learning*. PMLR, 2017, pp. 894–903.

LEARNING MULTIFACETED SELF-SIMILARITY OVER TIME AND FREQUENCY FOR MUSIC STRUCTURE ANALYSIS

Tsung-Ping Chen¹ and Kazuyoshi Yoshii²

¹Graduate School of Informatics, Kyoto University, Japan

²Graduate School of Engineering, Kyoto University, Japan

chen.tsungping.74e@st.kyoto-u.ac.jp, yoshii.kazuyoshi.3r@kyoto-u.ac.jp

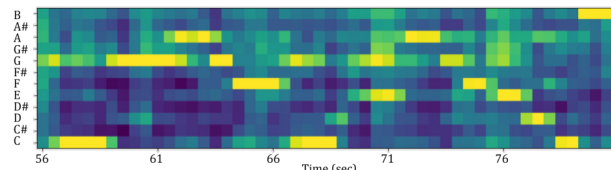
ABSTRACT

This paper describes a deep learning method for music structure analysis (MSA) that aims to split a music signal into temporal segments and assign a function label (e.g., intro, verse, or chorus) to each segment. The computational base for MSA is a spectro-temporal representation of input audio such as the spectrogram, where the compositional relationships of the spectral components provide valuable clues (e.g., chords) to the identification of structural units. However, such implicit features might be vulnerable to local operations such as convolution and pooling operations. In this paper, we hypothesize that the self-attention over the spectral domain as well as the temporal domain plays a key role in tackling MSA. Based on this hypothesis, we propose a novel MSA model built on the Transformer-in-Transformer architecture that alternately stacks spectral and temporal self-attention layers. Experiments with the Beatles, RWC, and SALAMI datasets showed the superiority of the dual-aspect self-attention. In particular, the differentiation between spectral and temporal self-attentions can provide extra performance gain. By analyzing the attention maps, we also demonstrate that self-attention can unfold tonal relationships and the internal structure of music.

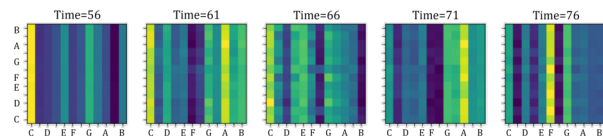
1. INTRODUCTION

Music structure refers to the sequential arrangement of musically coherent units that form a musical work. Music structure analysis (MSA) calls for a comprehensive understanding of various musical elements such as rhythm, melody, and harmony, and has still been an open problem in the field of music information retrieval (MIR), partly due to its multifaceted and ill-defined nature [1].

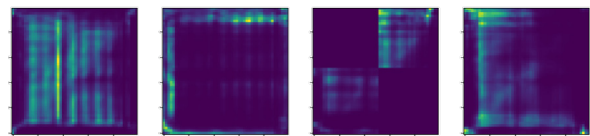
There are two major ways of representing music structure. The *semiotic* representation [2] uses a set of arbitrary symbols (e.g., A-B-C-B-C) for revealing the relationships between segments within a musical piece. The *functional*



(a) Chromagram representing a chorus section



(b) Multifaceted spectral self-attention maps over time



(c) Multi-scale temporal self-attention maps

Figure 1: Non-local dependencies of music such as chord tones and repeated patterns can be captured with spectral and temporal self-attention mechanisms. (a) The chromagram shows that this musical excerpt is dominated by the C major chord. (b) The spectral attention maps show that pitch classes C, E, and G persistently draw attention while the chroma features vary over time. (c) The temporal attention maps delineate the internal structures of this excerpt.

representation uses a set of semantic labels (e.g., intro-verse-chorus-verse-chorus) for indicating the roles of individual units. The functional representation can be converted into the semiotic one, but not vice versa.

A common deep learning approach to MSA involves using a convolutional neural network (CNN) to extract latent features from a spectro-temporal representation of input audio, such as a mel spectrogram or chromagram [3–5]. The assumption underlying this approach is the time-frequency locality of musical features, which should be treated with caution when characterizing the global structure of music. Actually, musical elements have non-local dependencies. Chords consist of musical sounds that are widely distributed over frequency. Musical patterns such as chord progressions or musical phrases are commonly repeated over time. Such non-local time-frequency dependencies can hardly be captured by a CNN that ag-

gregates local features while reducing the time-frequency dimensions with pooling operations [6, 7].

One promising architecture for learning non-local dependencies in a music recording is the SpecTNT [8], a variant of the Transformer-in-Transformer (TNT) [9] for modeling spectrogram-like representations. The SpecTNT iterates feature transforms of the multi-head self-attention (MHSA) mechanism [10] alternately along the spectral and temporal axes while keeping the time-frequency dimensions of input features. This method, however, suffers from a potential performance limitation because the individual characteristics of spectral and temporal dimensions are indistinguishable to the MHSA.

To overcome this limitation, we propose to integrate specialized MHSA mechanisms into the SpecTNT architecture for the MSA task regarding the functional representation. As outlined in Figure 1, our method involves gathering spectral and temporal information alternately from an input spectro-temporal representation with two types of MHSA mechanisms. The *spectral self-attention* extracts compositional relationships among two types of spectral features at each time step. The *temporal self-attention* aggregates spectral information at multiple time scales. The proposed method is systematically evaluated with three corpora consisting of popular music. In addition, the attention maps are analyzed to advocate paying attention to the non-locality of musical features.

The main contribution of this work is to emphasize the non-local dependencies of music over frequency as well as that over time. While non-local temporal correlations have been extensively studied, spectral non-locality remains underrepresented. In this concern, we adapt MHSA mechanisms to both aspects and analyze the self-attention maps to elaborate on the non-locality of musical features. This work may draw attention to such delicate characteristics that could be crucial for various tasks in MIR.

2. RELATED WORK

Segmentation and labeling are two subtasks of MSA [11]. The former detects the boundaries of structural units, and the latter categorizes musical segments either by the relationships with the semiotic representation or by the structural roles with the functional representation.

For the segmentation task, a spectro-temporal representation or a sequence of higher-level features extracted by a CNN is typically used to compute a novelty curve [12–16], from which musical boundaries are retrieved with a peak-picking algorithm [17, 18]. A key feature of our method is that we employ CNNs without any pooling layers for feature extraction. Since adjacent spectral beams are irrelevant in the sense of music, naive local pooling would hinder the learning of spectral patterns.

For the labeling task, the similarity-based approach is commonly taken in support of the semiotic representation of music structure [19–24], yet deep learning classification frameworks have recently been introduced for the estimation of structural functions [25, 26]. For both scenarios, the segmentation of an input piece will be a byprod-

uct of the labeling task. However, a smoothing method is typically required to refine the fragmented segmentation results caused by unusual label changes. Our method performs joint estimation of functional labels and musical boundaries to alleviate the fragmentation issue.

MHSA-based methods have recently been proposed for MSA owing to the excellent representation capability. The SpecTNT for MSA [27] uses Transformer encoders to capture the dependencies between the two axes of an input spectro-temporal representation. For training the SpecTNT with an increased amount of data, structure annotations from multiple datasets are mapped to the same semantic space with a 7-class taxonomy (‘intro’, ‘verse’, ‘chorus’, ‘bridge’, ‘inst’, ‘outro’, and ‘silence’). While the spectral components are often used for temporal modeling or collapsed before temporal modeling, this is the first attempt in the MSA task to retain the spectral dimension. In contrast, the convolution-augmented MHSA (CAMHSA) mechanism [28] captures temporal self-similarities on the self-attention maps derived from multiple types of acoustic features for capturing the repetitive nature of music. These network designs impose inductive biases that can enhance the representation learning.

Given the complementary aspects of these techniques, we integrate specialized MHSA mechanisms into the SpecTNT architecture for better modeling non-local features in spectral and temporal dimensions. Compared with the original CAMHSA [28], we retain the spectral dimension of the input data and aim to estimate the functional structure instead of the semiotic one, because the functional description conveys generic attributes of structural units that are comprehensible to the public. Compared with the original SpecTNT [27], we deal with spectro-temporal characteristics of music and processes input data at the track level rather than at the chunk (or segment) level, because the functional role of a structural unit might depend on the global organization of a musical piece.

3. PROPOSED METHOD

We tackle the functional MSA task with the 7-class taxonomy [27]. The estimation of the functional structure is formulated as a sequence labeling problem. Given a spectro-temporal representation, $\mathbf{X} \in \mathbb{R}^{T \times S}$, with S spectral components and T time steps, the goal of the estimation task is to output a sequence of categorical labels, $\mathbf{C} \in \mathbb{R}^T$, indicating the structural function of each time step $t \in T$. In practice, an extra binary sequence, $\mathbf{B} \in \{0, 1\}^T$, which specifies whether t is a boundary, is generated for smoothing the estimated labels within a segment surrounded by two boundaries. As depicted in Figure 2a, our model consists of three parts: a CNN-based frontend, L stacks of spectral and temporal encoders, and output layers in charge of the predictions of \mathbf{B} and \mathbf{C} respectively (in this paper, we use $L=2$ stacks for experiments).¹

¹The source code is available at <https://github.com/Tsung-Ping/music-structure-analysis>.

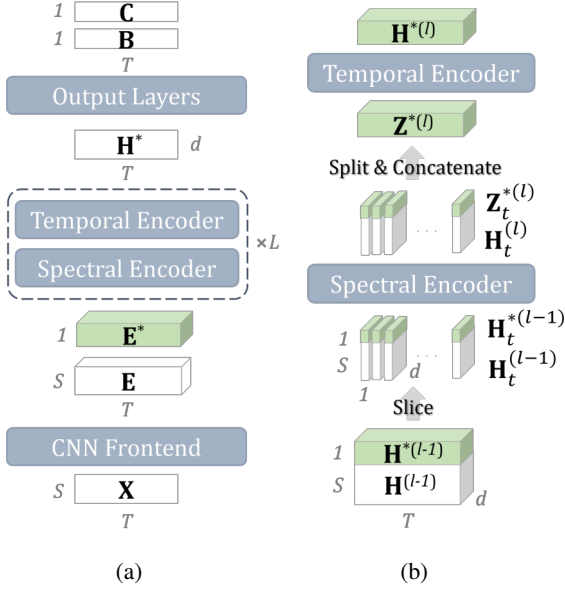


Figure 2: (a) Model architecture. Extra spectral components \mathbf{E}^* are stacked on the frontend output \mathbf{E} before the encoder blocks. (b) Schematic diagram of the spectral and temporal encodings, where the initial input $\mathbf{H}^{(0)} = \mathbf{E}$, $\mathbf{H}^{*(0)} = \mathbf{E}^*$, and the final output $\mathbf{H}^{*(L)} = \mathbf{H}^*$. The spectral encoder represents the time slices ($[\mathbf{H}_t^*; \mathbf{H}_t]$) separately, and then the temporal encoder aggregates the extra components (\mathbf{Z}_t^*) and outputs a representation.

3.1 CNN Frontend

The frontend is composed of an initial stem with two 2-D convolutional layers followed by a residual block [29]. Let f_c denote a convolutional layer with d filters parameterized by θ . The outputs of the stem and the residual block, denoted by $\{\mathbf{X}', \mathbf{E}\} \in \mathbb{R}^{T \times S \times d}$, are computed as follows:

$$\mathbf{E} = f_c(f_c(\mathbf{X}', \theta_3), \theta_4) + \mathbf{X}', \quad (1)$$

$$\mathbf{X}' = f_c(f_c(\mathbf{X}, \theta_1), \theta_2). \quad (2)$$

In effect, two frontends are employed to leverage different types of acoustic features. The two networks take as input the mel spectrogram ($\mathbf{X}_1 \in \mathbb{R}^{T \times S_1}$) and the chromagram ($\mathbf{X}_2 \in \mathbb{R}^{T \times S_2}$) respectively, and output $\mathbf{E}_1 \in \mathbb{R}^{T \times S_1 \times d}$ and $\mathbf{E}_2 \in \mathbb{R}^{T \times S_2 \times d}$. A unified representation is then obtained by concatenating the two outputs along the spectral dimension, i.e., $\mathbf{E} \in \mathbb{R}^{T \times (S_1+S_2) \times d}$. We set $S_1 = 80$, $S_2 = 12$, and $d = 80$ for this work.

Note that adjacent pitch classes are irrelevant in the traditional sense of musical harmony, and we thus design the frontend for the chromagram carefully. Specifically, we concatenate \mathbf{X}_2 and the first 11 columns of \mathbf{X}_2 along the pitch-class axis, i.e., $\hat{\mathbf{X}}_2 = \text{concat}(\mathbf{X}_2, \mathbf{X}_2[:, 1:11])$, and use convolutional layers with kernels that enclose the 12 pitch classes at once. This manipulation enables the CNN to capture key (or tonic)-independent patterns.

3.2 Spectral and Temporal Encoders

The spectral and temporal encoders are both built upon the Transformer encoder [10] that comprises stacks of MHSA

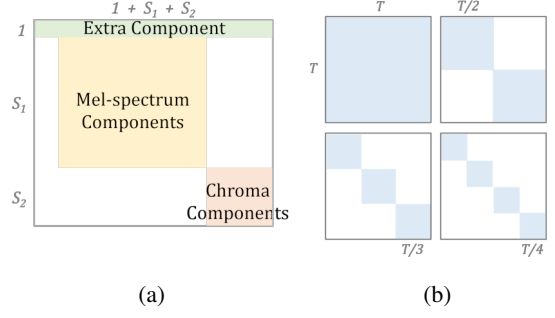


Figure 3: Attention masks for (a) the spectral MHSA and (b) the temporal MHSA. The rows (resp. columns) indicate queries (resp. keys) of the MHSA mechanism. Attention scores outside the colored regions will be filtered out.

blocks and feed-forward networks. Motivated by [28], we replace the standard MHSA of the temporal encoder with the CAMHSA mechanism. Moreover, we leverage relative position encodings [30] for enabling the model to process audio tracks of variable length.

As illustrated in Figure 2b, the two encoders in a stack are sequentially applied to the output of the previous stack. Let $\mathbf{E}_t \in \mathbb{R}^{S \times d}$ denote a time slice of \mathbf{E} at t , and $\mathbf{E}_t^* \in \mathbb{R}^{1 \times d}$ a learnable vector that behaves as an extra spectral component. The spectral encoder (SE) jointly extracts latent features $\mathbf{Z}_t^* \in \mathbb{R}^{1 \times d}$ and $\mathbf{H}_t \in \mathbb{R}^{S \times d}$ from \mathbf{E}_t^* and \mathbf{E}_t . Then, the temporal encoder (TE) exchanges information of \mathbf{Z}_t^* across time as follows:

$$\mathbf{H}^* = [\mathbf{H}_1^{*(L)}; \dots; \mathbf{H}_T^{*(L)}], \quad (3)$$

$$[\mathbf{H}_1^{*(l)}; \dots; \mathbf{H}_T^{*(l)}] = \text{TE}([\mathbf{Z}_1^{*(l)}; \dots; \mathbf{Z}_T^{*(l)}]), \quad (4)$$

$$[\mathbf{Z}_t^{*(l)}; \mathbf{H}_t^{(l)}] = \text{SE}([\mathbf{H}_t^{*(l-1)}; \mathbf{H}_t^{(l-1)}]), \quad (5)$$

where $\mathbf{H}^* \in \mathbb{R}^{T \times d}$ is the final output of the temporal encoder, $[\cdot; \cdot]$ denotes concatenation along the first dimension, l is the index of the stack, $\mathbf{H}_t^{*(0)} = \mathbf{E}_t^*$, and $\mathbf{H}_t^{(0)} = \mathbf{E}_t$. The extra spectral component \mathbf{H}_t^* in each stack mimics the initial CLS token introduced in the BERT model [31] and is used to encapsulate spectral information at each time step. For a detailed description of the intertwined architecture, we refer the readers to [8].

To use the SE and TE for modeling spectral and temporal dependencies, we impose constraints on the attention maps of the MHSA block, as shown in Figure 3. For the SE, the attention between the two types of spectra (i.e., mel spectrum and the chroma features) and the attention on the extra component are masked out because such attentions would likely result in *diluted* representations [32]. While the between-type attentions are prohibited, their relations can be extracted via the extra component. For the TE, contextual information is aggregated simultaneously at four time scales (i.e., T , $T/2$, $T/3$, and $T/4$) in a *structure-aware* manner. Take the scale $T/2$ as an example, a time step $t < T/2$ can only attend the first half of the time axis. Considering that binary and ternary forms are common structures in Western music, such location-related information is expected to enhance the learning of music structure.

We leverage the multi-head nature of the MHSA block for simultaneous multi-scale attention.

3.3 Output Layers and Inference

The output layers consist of a three-layer fully connected neural network and take \mathbf{H}^* as input to estimate the boundary likelihoods, $\mathbf{P}^B \in [0, 1]^T$, and the probability distributions over the 7 classes, $\mathbf{P}^C \in [0, 1]^{T \times 7}$, for all time steps:

$$\mathbf{P}^B = \text{sigmoid}(((\mathbf{H}^* \mathbf{W}_1^B) \mathbf{W}_2^B) \mathbf{W}_3^B), \quad (6)$$

$$\mathbf{P}^C = \text{sigmoid}(((\mathbf{H}^* \mathbf{W}_1^C) \mathbf{W}_2^C) \mathbf{W}_3^C), \quad (7)$$

where $\{\mathbf{W}_1^B, \mathbf{W}_2^B, \mathbf{W}_3^B, \mathbf{W}_1^C, \mathbf{W}_2^C, \mathbf{W}_3^C\} \in \mathbb{R}^{d \times d}$, $\mathbf{W}_3^B \in \mathbb{R}^{d \times 1}$, and $\mathbf{W}_3^C \in \mathbb{R}^{d \times 7}$ are learnable weight matrices. Note that we use the sigmoid function instead of the softmax activation in Eqn (7) for the function labeling is modeled as seven individual binary sequences.

To detect the boundaries \mathbf{B} from \mathbf{P}^B , we use a common peak-picking method [17] implemented in the librosa library [33]. To estimate the function labels \mathbf{C} , we give the segment between two adjacent boundaries (say t_1 and t_2) a label taking the largest average probability, i.e., $\mathbf{C}_t = \arg \max \sum \mathbf{P}_n^C \forall t_1 \leq t, n < t_2$.

3.4 Loss Function

Given the ground-truth boundaries and labels (represented by a sequence of one-hot vectors), $\mathbf{Y}^B \in \{0, 1\}^T$ and $\mathbf{Y}^C \in \{0, 1\}^{T \times 7}$ respectively, we compute the binary cross-entropy (BCE) losses for the model output to compute the overall loss (\mathcal{L}) as follows:

$$\mathcal{L} = \mathcal{L}^B + \mathcal{L}^C, \quad (8)$$

$$\mathcal{L}^B = \text{BCE}(\mathbf{Y}^B, \mathbf{P}^B), \quad (9)$$

$$\mathcal{L}^C = \text{BCE}(\mathbf{Y}^C, \mathbf{P}^C). \quad (10)$$

4. EXPERIMENTS

We conducted comparative experiments using the Beatles [34], RWC [35], and SALAMI [36] datasets. For the Beatles dataset, we used the refined Beatles-TUT annotations for 174 Beatles songs.² For the RWC dataset, we used the 100 songs from the Popular Music Database (denoted by RWC-POP). For the SALAMI dataset, we created a subset consisting of only popular music (SALAMI-POP), which amounted to 245 tracks. The maximum track length for each corpus was around 468 sec, 368 sec, and 438 sec. Following [27], we carried out cross-dataset evaluations for all the experiments. Each of the three corpora served as the test data in turn while the remainder was used for training. We augmented the training set via pitch shifting (within ± 2 semitones) and pre-emphasis (with a coefficient of $\{0.7, 0.97\}$).

4.1 Statistics of the Function Labels

Structural annotations of the three corpora were converted to the 7-class label space with the mapping algorithm proposed in [27]. As illustrated in Figure 4, all the corpora

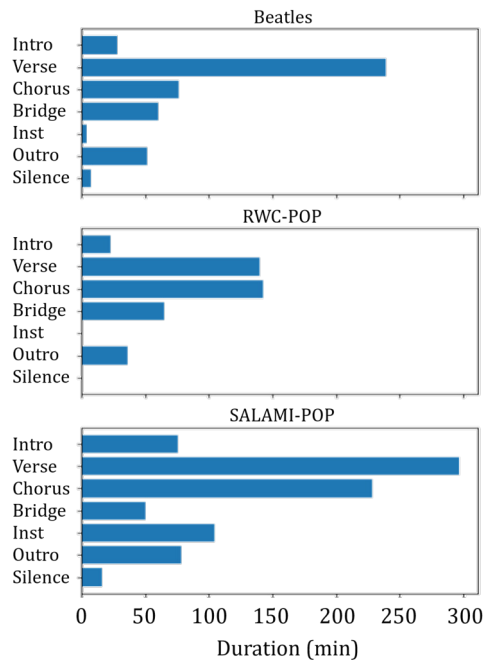


Figure 4: Statistics of the 7 function labels in each of the Beatles, RWC-POP, and SALAMI-POP corpora.

were with concentrated distribution, where Verse and Chorus are the most common labels, since that verse-chorus form is widely used in popular music. It is also worth noting that Inst (i.e., 'instrumental') is extremely rare in the Beatles and RWC-POP datasets as a result of their annotation criteria and the mapping algorithm.

4.2 Input Representation

For each audio track, we computed the mel spectrogram with 80 mel bands and the chromagram with 12 chroma bins. The initial time resolution for both representations was 25 ms. We downsampled the two types of features by a factor of 20 (hence 1 frame = 0.5 sec) with the median filter so that the model could take as input a full-length track under memory constraints.

4.3 Evaluation Metrics

The performance on the MSA task was evaluated with the *mir_eval* library [37] in terms of segmentation and labeling. For segmentation, we computed the F1 score of the Hit Rate [38] with a time tolerance of ± 0.5 sec and ± 3 sec (denoted by HR.5F and HR3F respectively). For labeling, we computed the F1 score of the pairwise agreement [39] at the frame size of 0.1 sec (denoted by PWF).

In addition, the frame-wise labeling accuracy was measured in two ways. First, we converted the sequence of probabilities (\mathbf{P}^C) into the labeling sequence (\mathbf{C}) either by taking the *argmax* function at each time step or by using the proposed *smoothing* strategy (Section 3.3). The derived labeling sequences were denoted by \mathbf{C}_a and \mathbf{C}_s , respectively. Two types of labeling accuracy (ACC_a and ACC_s) were then computed by comparing \mathbf{C}_a and \mathbf{C}_s with the

²<https://pythonhosted.org/msaf/datasets.html>.

Method	ACC _a / ACC _s	HR.5F/ HR3F	PWF
Beatles			
Proposed	0.495/ 0.481	0.521/ 0.638	0.571
ST-MHSA	0.410/ 0.386	0.480/ 0.610	0.576
TE-Only	0.455/ 0.451	0.484/ 0.610	0.547
SE-Only	0.355/ 0.330	0.448/ 0.600	0.594
RWC-POP			
Proposed	0.589/ 0.598	0.570/ 0.712	0.623
ST-MHSA	0.425/ 0.426	0.498/ 0.637	0.537
TE-Only	0.528/ 0.531	0.504/ 0.662	0.578
SE-Only	0.428/ 0.430	0.472/ 0.644	0.562
SALAMI-POP			
Proposed	0.497/ 0.492	0.505/ 0.657	0.600
ST-MHSA	0.411/ 0.401	0.435/ 0.559	0.561
TE-Only	0.422/ 0.411	0.452/ 0.582	0.575
SE-Only	0.425/ 0.390	0.418/ 0.492	0.552

Table 1: The result of the ablation study.

ground-truth labeling sequence, $\bar{\mathbf{C}} \in \{0, 1, \dots, 6\}^T$:

$$\text{ACC}_a = \frac{1}{T} \sum_{t=1}^T \delta_{\bar{\mathbf{C}}_t, \mathbf{C}_{a,t}}, \quad (11)$$

$$\text{ACC}_s = \frac{1}{T} \sum_{t=1}^T \delta_{\bar{\mathbf{C}}_t, \mathbf{C}_{s,t}}, \quad (12)$$

where $\delta_{a,b}$ denotes the Kronecker delta function.

4.4 Baseline Methods

To validate the effectiveness of the spectral and temporal self-attentions, we conducted an ablation study with the following baseline models:

- **ST-MHSA:** Both the spectral and temporal encoders used the standard MHSA as in the SpectTNT [27].
- **TE-Only:** The spectral encoder was removed in a way similar to the CAMHSA work [28].
- **SE-Only:** The temporal encoder was removed from the proposed model. This was a localized prediction model taking only short-term context into account.

The ST-MHSA and TE-only are considered substitutes to the two previous works [27, 28], and we did not make a direct comparison to their results for a couple of reasons. First, the data used are different due to the difficulty of obtaining the exact audio signals: we used only the Beatles dataset while [27] used the Isophonics [34]; the subsets created from the SALAMI dataset are also different (245 tracks in our experiments and 274 tracks in [27]). Second, our model aims to predict semantic labels whereas the model of [28] outputs semiotic representations, and accordingly the data used are different.

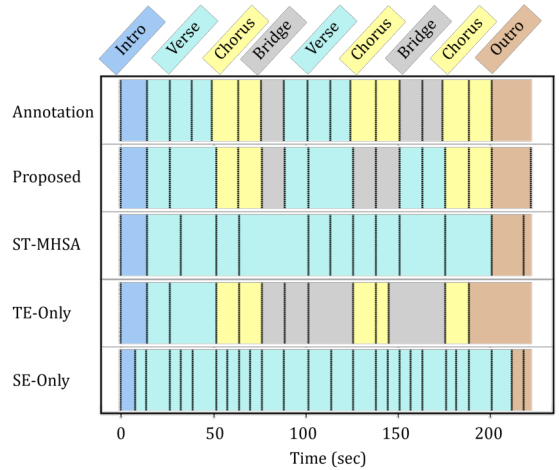


Figure 5: Structure analysis results of a song (“RM-P045”) from the RWC-POP. The first row is the ground-truth annotation, and the other rows are the estimations by the proposed method and baseline models. The estimated boundaries are denoted by dashed lines.

To scrutinize our model design in relation to the performance, we also built variants of our model as follows:

- **w/ Pool:** The pooling operation was inserted into the CNN frontend for the mel spectrogram (Section 3.1). Precisely, we used a pooling layer for \mathbf{X}'_1 before the computation of Eqn (1). Following [12], we reduced the spectral dimension of the mel spectrogram using a max-pooling layer with a kernel size of 6 (while the temporal dimension was kept unchanged).
- **w/o S-Mask:** The spectral attention mask was not used (Section 3.2 and Figure 3a). This is equivalent to using a standard MHSA block for the SE.
- **w/o T-Mask:** The temporal attention mask was not used (Section 3.2 and Figure 3b). This is equivalent to using the CAMHSA mechanism for the TE.

5. RESULTS

We here report and discuss the results of the comparative and ablation experiments.

5.1 Comparison with Baseline Models

The results of the cross-dataset evaluations are summarized in Table 1. The proposed method outperformed the baseline methods on the three corpora in most metrics (with a comparable PWF score to the ST-MHSA and the TE-Only on the Beatles). In comparison with the ST-MHSA, the performance gain of the proposed method was mainly attributed to the tailored MHSA blocks for spectral and temporal modelings, validating the importance of the MHSA adaptation to the task. Given that the TE-Only obtained better ACC scores than the SE-Only, the temporal self-attention was considered to have a greater impact on identifying structural functions than its spectral counterpart.

Method	ACC _a / ACC _s	HR.5F/ HR3F	PWF
Beatles			
Proposed	0.495/ 0.481	0.521/ 0.638	0.571
w/ Pool	0.387/ 0.370	0.446/ 0.566	0.536
w/o S-Mask	0.498/ 0.497	0.520/ 0.654	0.582
w/o T-Mask	0.538/ 0.531	0.528/ 0.643	0.614
RWC-POP			
Proposed	0.589/ 0.598	0.570/ 0.712	0.623
w/ Pool	0.528/ 0.489	0.395/ 0.550	0.546
w/o S-Mask	0.571/ 0.577	0.571/ 0.715	0.621
w/o T-Mask	0.576/ 0.567	0.549/ 0.686	0.607
SALAMI-POP			
Proposed	0.497/ 0.492	0.505/ 0.657	0.600
w/ Pool	0.495/ 0.454	0.418/ 0.522	0.552
w/o S-Mask	0.487/ 0.490	0.480/ 0.628	0.581
w/o T-Mask	0.471/ 0.480	0.485/ 0.635	0.586

Table 2: Evaluations of the model design choices.

Nonetheless, the differences between the SE-Only and TE-Only were not clear in terms of the HR and the PWF scores, implying that spectral and temporal self-attentions both can contribute to the tasks. In addition, we found that our inference strategy smoothed the structural labeling results while having a minor effect on the ACC score.

Figure 5 portrays the structural labeling results for one song from the RWC-POP corpus. Regarding the segmentation results, all the models were able to detect the transitions between different sections, but the finer structure (i.e., repetitions or variants within a coarse section) was sometimes overlooked. In particular, the SE-Only over-segmented the musical track due to the limited contextual information. Regarding the labeling results, the proposed method and the TE-Only were capable of correctly estimating the five function labels in the track, whereas the ST-MHSA and SE-Only failed to identify all the chorus and bridge sections, possibly owing to the insufficient capability of temporal modeling.

5.2 Evaluation of Design Choices

Experiments results regarding the model design are listed in Table 2. As we expected, the severe performance degradation was caused by the pooling operation (w/ Pool) on the three corpora. Spectral components are not pixels that are highly correlated in local regions, and therefore naive local pooling could be detrimental to spectral features. As for the attention masking (w/o {S, T}-MASK), the results suggested that the imposed constraints can have a positive impact on the performance. On the SALAMI-POP, which is the most challenging one among the three corpora, the MHSA mechanism without any constraints resulted in a clear performance drop. In particular, we found that unconstrained spectral components tended to give great attention to the extra component (\mathbf{H}_i^*) rather than themselves.

A similar effect was also reported by previous research in the field of natural language processing [40, 41]. This kind of concentrated attention to a special (or artificial) component that has distinct semantic meanings could downplay the representation capability of the MHSA.

5.3 Evaluation of Spectro-Temporal Self-Attentions

The attention maps implicitly computed with the MHSA mechanism often disclose illuminating relationships between input elements [42–45]. The spectral and temporal self-attentions of our model also exhibited such an effect. As depicted in Figure 1c, the leftmost temporal self-attention map highlighted a potential musical event at around 66 sec, which could be associated with the variant repeat of the first 10 sec of this chorus section (as can be seen in Figure 1a, two triangular patterns span from 56 to 66 sec and from 66 to 76 sec, respectively). This result echoes the observation that self-attention maps can represent music structure [28]. In contrast, the spectral self-attention, as illustrated in Figure 1b, uncovered the tonal relationships between the 12 pitch classes with an emphasis on the notes comprising the tonic chord (assume in the key of C major). Particularly, pitch class E gained persistent attention over this section even though it had a low energy level for most of the time. Through alternate self-attention across the spectral and temporal dimensions, the contextual information of individual aspects can be mingled effectively and provide insights into music structure.

6. CONCLUSION

We have presented a deep learning model for music structure analysis, especially from the perspective of the functional structure representation. The core idea of this study is to learn non-local spectral and temporal dependencies inherent in music with clear distinction. For this purpose, we adapted the multi-head self-attention mechanism for each aspect and leveraged two types of the Transformer encoder to unravel the spectro-temporal relationships. Compared with the ablated variants of the Transformer encoder, the proposed model with the specialized self-attention mechanisms worked better on three datasets in music segmentation and structure labeling. The learned self-attention maps unveiled that the correlations between separated spectral or temporal components can be effective clues for modeling music structure.

In spite of these encouraging results, we acknowledge the computational limitation of our approach. Apart from an M -head temporal self-attention having the memory footprint of $M \times T \times T$, an N -head spectral self-attention involves intermediate attention maps with $T \times N \times S \times S$ space complexity. Given that our method aims to process full-length audio data (hence larger T) and leverage multiple types of acoustic features (hence greater S), memory-efficient self-attention mechanisms are critical to this kind of dual-axis modeling. Time and frequency are intricately interwoven to form the musical fabric, and each individual aspect is worth considerable attention.

7. ACKNOWLEDGMENT

This work was partially supported by JST FOREST No. JPMJFR2270 and JSPS KAKENHI Nos. JP24H00742, JP24H00748, and JP24KJ1379.

8. REFERENCES

- [1] O. Nieto, G. J. Mysore, C. Wang, J. B. L. Smith, J. Schlüter, T. Grill, and B. McFee, “Audio-based music structure analysis: Current trends, open challenges, and applications,” *Transactions of the International Society for Music Information Retrieval (TISMIR)*, vol. 3, no. 1, pp. 246–263, 2020.
- [2] F. Bimbot, E. Deruty, G. Sargent, and E. Vincent, “Semiotic structure labeling of music pieces: Concepts, methods and annotation conventions,” in *Proceedings of the 13th International Society for Music Information Retrieval Conference (ISMIR)*, 2012, pp. 235–240.
- [3] T. Grill and J. Schlüter, “Music boundary detection using neural networks on spectrograms and self-similarity lag matrices,” in *Proceedings of the 23rd European Signal Processing Conference (EUSIPCO)*, 2015, pp. 1296–1300.
- [4] A. Maezawa, “Music boundary detection based on a hybrid deep model of novelty, homogeneity, repetition and duration,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 206–210.
- [5] M. Buisson, B. McFee, S. Essid, and H. C. Crayencour, “Learning multi-level representations for hierarchical music structure analysis,” in *Proceedings of the 23rd International Society for Music Information Retrieval Conference (ISMIR)*, 2022, pp. 591–597.
- [6] T. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, “Feature pyramid networks for object detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 936–944.
- [7] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu, “WaveNet: A generative model for raw audio,” in *Proceedings of the 9th ISCA Speech Synthesis Workshop (SSW)*, 2016, p. 125.
- [8] W. T. Lu, J. Wang, M. Won, K. Choi, and X. Song, “SpecTNT: a time-frequency Transformer for music audio,” in *Proceedings of the 22nd International Society for Music Information Retrieval Conference (ISMIR)*, 2021, pp. 396–403.
- [9] K. Han, A. Xiao, E. Wu, J. Guo, C. Xu, and Y. Wang, “Transformer in Transformer,” in *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2021, pp. 15 908–15 919.
- [10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2017, pp. 5998–6008.
- [11] O. Nieto and J. P. Bello, “Systematic exploration of computational music structure research,” in *Proceedings of the 17th International Society for Music Information Retrieval Conference (ISMIR)*, 2016, pp. 547–553.
- [12] K. Ullrich, J. Schlüter, and T. Grill, “Boundary detection in music structure analysis using convolutional neural networks,” in *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR)*, 2014, pp. 417–422.
- [13] T. Grill and J. Schlüter, “Music boundary detection using neural networks on combined features and two-level annotations,” in *Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR)*, 2015, pp. 531–537.
- [14] M. C. McCallum, “Unsupervised learning of deep features for music segmentation,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 346–350.
- [15] C. Hernandez-Olivan, J. R. Beltrán, and D. Diaz-Guerra, “Music boundary detection using convolutional neural networks: A comparative analysis of combined input features,” *International Journal of Interactive Multimedia and Artificial Intelligence (IJ-MAI)*, vol. 7, no. 2, p. 78, 2021.
- [16] G. Peeters, “Self-similarity-based and novelty-based loss for music structure analysis,” in *Proceedings of the 24th International Society for Music Information Retrieval Conference (ISMIR)*, 2023, pp. 749–756.
- [17] S. Böck, F. Krebs, and M. Schedl, “Evaluating the online capabilities of onset detection methods,” in *Proceedings of the 13th International Society for Music Information Retrieval Conference (ISMIR)*, 2012, pp. 49–54.
- [18] J. Serrà, M. Müller, P. Grosche, and J. L. Arcos, “Unsupervised music structure annotation by time series structure features and segment similarity,” *IEEE Transactions on Multimedia*, vol. 16, no. 5, pp. 1229–1240, 2014.
- [19] B. McFee and D. Ellis, “Analyzing song structure with spectral clustering,” in *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR)*, 2014, pp. 405–410.
- [20] C. Wang and G. J. Mysore, “Structural segmentation with the Variable Markov Oracle and boundary adjustment,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 291–295.

- [21] T. Cheng, J. B. L. Smith, and M. Goto, “Music structure boundary detection and labelling by a deconvolution of path-enhanced self-similarity matrix,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 106–110.
- [22] C. J. Tralie and B. McFee, “Enhanced hierarchical music structure annotations via feature level similarity fusion,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 201–205.
- [23] A. Marmoret, J. E. Cohen, and F. Bimbot, “Barwise music structure analysis with the correlation block-matching segmentation algorithm,” *Transactions of the International Society for Music Information Retrieval (TISMIR)*, vol. 6, no. 1, pp. 167–185, 2023.
- [24] M. Buisson, B. McFee, S. Essid, and H. C. Crayencour, “Self-supervised learning of multi-level audio representations for music segmentation,” *IEEE ACM Transactions on Audio, Speech, and Language Processing (TASLP)*, vol. 32, pp. 2141–2152, 2024.
- [25] G. Shibata, R. Nishikimi, and K. Yoshii, “Music structure analysis based on an LSTM-HSMM hybrid model,” in *Proceedings of the 21th International Society for Music Information Retrieval Conference (ISMIR)*, 2020, pp. 23–29.
- [26] J. Wang, J. B. L. Smith, J. Chen, X. Song, and Y. Wang, “Supervised chorus detection for popular music using convolutional neural network and multi-task learning,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 566–570.
- [27] J. Wang, Y. Hung, and J. B. L. Smith, “To catch a chorus, verse, intro, or anything else: Analyzing a song with structural functions,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 416–420.
- [28] T. Chen, L. Su, and K. Yoshii, “Learning multifaceted self-similarity for musical structure analysis,” in *Proceedings of the Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2023, pp. 165–172.
- [29] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.
- [30] P. Shaw, J. Uszkoreit, and A. Vaswani, “Self-attention with relative position representations,” in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 2018, pp. 464–468.
- [31] J. Devlin, M. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional Transformers for language understanding,” in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 2019, pp. 4171–4186.
- [32] S. Mehri and M. Eric, “Example-driven intent prediction with observers,” in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 2021, pp. 2979–2992.
- [33] B. McFee, C. Raffel, D. Liang, D. P. W. Ellis, M. McVicar, E. Battenberg, and O. Nieto, “librosa: Audio and music signal analysis in Python,” in *Proceedings of the 14th Python in Science Conference (SciPy)*, 2015, pp. 18–24.
- [34] M. Mauch, C. Cannam, M. Davies, S. Dixon, C. Harte, S. Kolozali, D. Tidhar, and M. Sandler, “OMRAS2 metadata project 2009,” in *Proceedings of the 10th International Society for Music Information Retrieval Conference (ISMIR) - Late-Breaking Session*, 2009.
- [35] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, “RWC music database: Popular, classical and jazz music databases,” in *Proceedings of the 3rd International Conference on Music Information Retrieval (ISMIR)*, 2002.
- [36] J. B. L. Smith, J. A. Burgoyne, I. Fujinaga, D. D. Roure, and J. S. Downie, “Design and creation of a large-scale database of structural annotations,” in *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR)*, 2011, pp. 555–560.
- [37] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, and D. P. W. Ellis, “mir_eval: A transparent implementation of common MIR metrics,” in *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR)*, 2014, pp. 367–372.
- [38] D. Turnbull, G. R. G. Lanckriet, E. Pampalk, and M. Goto, “A supervised approach for detecting boundaries in music using difference features and boosting,” in *Proceedings of the 8th International Conference on Music Information (ISMIR)*, 2007.
- [39] M. Levy and M. B. Sandler, “Structural segmentation of musical audio by constrained clustering,” *IEEE Transactions on Speech and Audio Processing*, vol. 16, no. 2, pp. 318–326, 2008.
- [40] K. Clark, U. Khandelwal, O. Levy, and C. D. Manning, “What does BERT look at? An analysis of BERT’s attention,” in *Proceedings of the 2019 ACL Workshop BlackboxNLP*, 2019, pp. 276–286.

- [41] O. Kovaleva, A. Romanov, A. Rogers, and A. Rumshisky, “Revealing the dark secrets of BERT,” in *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 4364–4373.
- [42] C. A. Huang, A. Vaswani, J. Uszkoreit, I. Simon, C. Hawthorne, N. Shazeer, A. M. Dai, M. D. Hoffman, M. Dinculescu, and D. Eck, “Music Transformer: Generating music with long-term structure,” in *Proceedings of the 7th International Conference on Learning Representations (ICLR)*, 2019.
- [43] J. Vig and Y. Belinkov, “Analyzing the structure of attention in a Transformer language model,” in *Proceedings of the ACL Workshop BlackboxNLP*, 2019, pp. 63–76.
- [44] T. Chen and L. Su, “Attend to chords: Improving harmonic analysis of symbolic music using Transformer-based models,” *Transactions of the International Society for Music Information Retrieval (TISMIR)*, vol. 4, no. 1, pp. 1–13, 2021.
- [45] K. Shim, J. Choi, and W. Sung, “Understanding the role of self attention for efficient speech recognition,” in *Proceedings of the 10th International Conference on Learning Representations (ICLR)*, 2022.

A CONTRASTIVE SELF-SUPERVISED LEARNING SCHEME FOR BEAT TRACKING AMENABLE TO FEW-SHOT LEARNING

Antonin Gagneré Slim Essid Geoffroy Peeters
LTCI - Télécom Paris, Institut Polytechnique de Paris, France

antonin.gagnere@telecom-paris.fr

ABSTRACT

In this paper, we propose a novel Self-Supervised-Learning scheme to train rhythm analysis systems and instantiate it for few-shot beat tracking. Taking inspiration from the Contrastive Predictive Coding paradigm, we propose to train a Log-Mel-Spectrogram-Transformer-encoder to contrast observations at times separated by **hypothesized** beat intervals from those that are not. We do this without the knowledge of ground-truth tempo or beat positions, as we rely on the local maxima of a Predominant Local Pulse function, considered as a proxy for Tatum positions, to define candidate anchors, candidate positives (located at a distance of a power of two from the anchor) and negatives (remaining time positions). We show that a model pre-trained using this approach on the unlabeled FMA, MTT and MTG-Jamendo datasets can successfully be fine-tuned in the few-shot regime, *i.e.* with just a few annotated examples to get a competitive beat-tracking performance.

1 Introduction

Beat-tracking, *i.e.* locating the times in a musical audio signal where beats are perceived or notated in the corresponding score, is still one of the most challenging subjects in the Music Information Retrieval (MIR) research field. This is owing to the large use of the beat information in many applications and to the complexity of the task: beats belong to a hierarchy/tree of rhythmic accentuations (hence entailing ambiguities), arise both from perceptual and cognitive cues. It, therefore, requires knowledge of the cultural specificities of the studied music.

To alleviate these issues, data-driven systems purely rely on training data composed of music tracks that have been annotated (supposedly) by experts. However, this labeling process remains costly and as a consequence, the amount of data annotated into beats (at most a few thousands tracks) remains extremely low in MIR, as compared to other research fields (speech or computer vision). For this reason, developing approaches that allow training

beat-tracking systems without annotated data, a.k.a. Self-Supervised Learning (SSL), is important. This is the goal of this paper.

By alleviating the need of large annotated datasets, SSL, has recently gained significant attention in the field of machine learning. The goal is to learn meaningful representations of the input data without the need for human annotations. To do so, the target outputs are directly inferred from the dataset itself, and often referred to as "pretext-task labels". Such supervision can be obtained by masking some part of the input and asking the model to predict it [1–4] or to generate two views of the same input and force a model to learn similar representations for the two views [5–8]. Another popular SSL approach is contrastive learning [9, 10] where one trains a network to predict whether two inputs are from the same class (or not) by forcing their trained embeddings to be more or less close from each other. Usually, upon pre-training completion, the model is fine-tuned in a supervised fashion for one or more downstream tasks, where the data is smaller in size.

Our contributions are the following:

- We propose a novel contrastive SSL scheme producing representations which are useful for automatic rhythm analysis tasks, in particular the beat-tracking task. Its key component is the pretext-task design exploiting Predominant Local Pulse (PLP) local maxima to effectively sample anchor, positive, and negative time-steps for our contrastive loss function.
- We show that the pre-trained model can be fine-tuned in a few-shot learning setting to get competitive beat-tracking results. Moreover, we show that our approach yields, in most cases, at least better performance than Zero-Note Samba (ZeroNS) [11], which is, to the best of our knowledge, the only alternative SSL approach to this problem to date.
- Furthermore we show that our model outperforms ZeroNS in a cross-dataset generalization setting.
- Finally we compare our model to the state of the art in a 8-fold cross-validation setting and show that it is competitive.

Paper organization. The paper is organized as follows. In section 2 we present works related to our proposal. In section 3 we present our proposed contrastive SSL training strategy. Finally in section 4 we present the results of the different experiments we performed. To facilitate reproducibility, we make our code available.¹

¹ https://github.com/antoningagnere/ssl_beat



2 Related Work

In the following, we provide a quick overview of related contrastive SSL techniques and review the attempts made along this line in the field of MIR, especially for beat and downbeat tracking. We also discuss the recent advances made towards solving these important MIR tasks.

2.1 Self-Supervised Representation Learning

Our approach takes inspiration from contrastive methods. In CPC (Contrastive Predicting Coding) [9], representations are learned from sequential data by predicting the future latent representations from the (aggregated) past ones. For this, an encoder is trained to produce latent representations with the task of making it easy to distinguish in the obtained latent space (positive) future latent representations from a set of negative samples. This encourages the model to capture meaningful information. Instead of predicting the future, in Wav2Vec2 [10] the task is to predict masked observations. In Wav2Vec2, features are extracted from an audio signal with a Convolutional Network and fed to a transformer encoder where some frames are masked. Additionally, the audio features are quantized and the model is trained to contrast the masked output with the quantized output and a set of distractors.

2.2 Self-Supervised Learning in MIR

Following the trend in speech processing research, SSL approaches have started to become popular in MIR. On the one hand, these approaches can be used to train general-purpose models, the so-called “foundation models” (such as MULE [12] or MERT [13]), which are supposed to be useful to solve a whole set of downstream tasks (see the MARBLE benchmark [14]). On the other hand, models can be developed to learn representations that are well aligned with a specific MIR task. Among those, learning representations that are equivariant to a semantic distortion of the audio signal has become a popular approach (e.g., for pitch or tempo estimation using siamese networks [15–18]).

Few works have proposed to apply SSL for rhythm analysis tasks. Zero-Note Samba (ZeroNS) [11] leverages the synchronization of the various instrument stems in a music track. For this, they separate music tracks into their percussive and non-percussive parts and train an encoder to force the synchronization between the corresponding latent representations, which are then used for beat tracking. In [19] they used binary metric regularity to derive supervision for their CRF loss, enabling the network to model a hierarchical metrical structure.

2.3 Beat and Downbeat tracking

Before the rise of deep-learning approaches, beat and downbeat tracking systems were based on two-step systems: first audio features were extracted from the audio signal (including an onset detection function, Predominant Local Pulse (PLP), spectral features or a novelty function);

then those were used as “observations” to a probabilistic model (such as Hidden Markov Models or Dynamic Bayesian Network) [20–22].

The shift toward data-driven approaches started with [23] where the authors proposed to process spectral features with bi-directional Long Short-Term Memory (LSTM) networks. [24] then proposed to replace the LSTM with a Temporal Convolutional Network (TCN) to process the spectral features. Later on, the model was improved by solving jointly multiple tasks (beat and downbeat positions, as well as tempo) [25, 26]. Currently, models based on the Transformer architecture, used in a multi-task setting (joint beat-downbeat tracking) are the most successful. In [27] the authors apply the Spectral-Temporal Transformer (SpecTNT) architecture [28] to tackle this task. This architecture combines a spectral transformer that processes harmonic features and a temporal transformer that aggregates the processed features over time. To further improve the performance, the authors combined SpecTNT with a Temporal Convolutional Network (TCN). Beat Transformer [29] incorporates dilated self-attention to capture long-range dependencies. Furthermore, in the middle layers, they alternate time-wise dilated self-attention with instrument-wise self-attention².

3 Proposed Contrastive Learning SSL scheme

In this paper, we propose a novel SSL approach to learn representations useful for rhythm analysis tasks, and instantiate it for the beat tracking downstream task. We aim to learn a projection (an encoder) such that the resulting projections of observations at PLP peaks whose distance from each other is a power of 2 are close to each other, and different otherwise. The two key insights behind this is that: i) a significant fraction of the PLP peaks (supposedly aligned to the tatum grid) is expected to represent beat positions, with high probability, and ii) most of the musical recordings tend to have a binary metric structure (i.e. beats can be musically divided by two and grouped by two). We conjecture that despite being over-simplistic, these ingredients are “good-enough” to define a pretext-task that will be effective for training representations useful for various rhythm analysis tasks, especially beat tracking, provided that a downstream fine-tuning phase is anyway envisaged. In the following we will refer to the distance between two PLP peaks as *tatum-unit* and denote it by *tu*.

We solve this pretext-task using contrastive learning. We learn to distinguish observations at times separated by an interval of a power of 2 in *tu* units, from those that are not. Once computed, the PLP function is used to select an anchor, its associated positive, and a set of negative samples. We further explain the procedure in section 3.1. We then train our encoder to attract the anchor and the positive while repelling the set of negatives in the latent space using a contrastive loss. We describe the architecture of

² The instrument-wise attention is conducted along the stems of a demixed audio signal, contributing to a comprehensive analysis of the audio data.

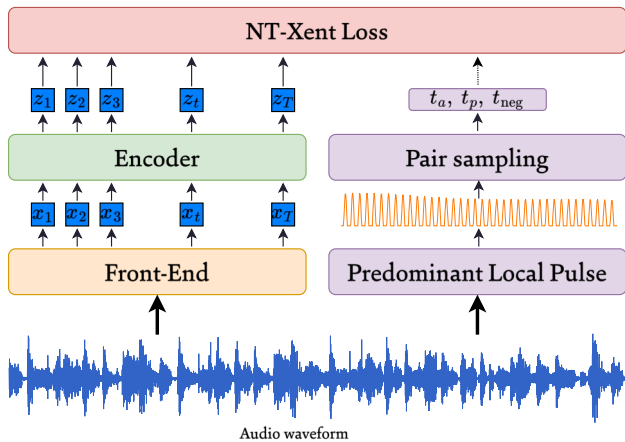


Figure 1: Our proposed contrastive SSL scheme for beat tracking. The left part displays our processed audio waveform to obtain the representations z_t . The right part displays our mining of positive and negatives.

our encoder in section 3.2. Our approach is summarized in Figure 1.

3.1 Mining positive and negatives

The key part of our work is to learn representations in a contrastive way. Therefore we need to define an anchor, a positive and multiple negative samples within each given audio excerpt. We rely on the Predominant Local Pulse (PLP) [30] function to extract local pulse information (see 3.1.1). Given such information, we sample positive and negative times for a selected anchor (see 3.1.2).

3.1.1 Predominant Local Pulse

The PLP method analyzes the Onset Strength Function (OSF) of an audio signal in the frequency domain to find a locally stable tempo for each frame. For this, a “tempogram” (a Short-Time Fourier-Transform, STFT) of the OSF is computed. At each time position, the maximum of the “tempogram” indicates the dominant pulse frequency. Using the corresponding amplitude and phase of this maximum, one can re-synthesize the corresponding temporal signal (a sinusoidal component). Using the usual overlap-and-add (OLA) inverse STFT method, a smooth temporal signal is formed by overlapping-and-adding the sinusoidal components with various dominant pulse frequencies over time. This temporal function is termed PLP and represents a localized enhancement of the original novelty function’s periodicity.

Computation. Given an audio signal, we compute the PLP function with the same frame rate as our audio front-end (*i.e.* 20ms). We used the *beat.plp* function from Librosa [31]. The function is fed with an OSF computed from a spectrogram³ with 2048 points and the default minimum and maximum tempo parameters. We then estimate the local maxima peak y_k of the PLP using the *find_peaks* function from scipy.

³ In a preliminary experiment, we found that using the spectrogram to get the OSF was working better than using the Mel-spectrogram

3.1.2 Sampling from PLP

In the following we will refer to the distance between two PLP peaks as *tatum-unit* and denote it by tu . For simplification, we do the following assumptions. We assume that the tatums correspond to the 8-th note and that most tracks are in a 4/4 meter.⁴ Following this, we consider that the positives have a time distance Δ from the anchor which is a power of two of the tatum unit: $\Delta = i \times \alpha \times tu$ with $\alpha = 2^n$ and $i \in \mathbb{Z} \setminus \{0\}$. In this work we consider $n = 2$ (which corresponds to an inter-distance of two beats).

We define by $Y = \{y_1, \dots, y_K\}$ the set of PLP peaks within a given audio segment. We first sample an anchor a uniformly in $[1, K]$. We denote by y_a the time associated to a (blue arrow in Fig.2). Given this anchor, we sample its associated positive time step p . This positive must be situated $i \times \alpha$ peaks away from the query. For a given anchor a , we therefore sample p uniformly from $Y_a = \{y_{a \pm i \times \alpha}, 0 \leq a \pm i \times \alpha \leq K\}$.

We denote by y_p the time associated to p (green arrow in Fig.2, green empty arrows are all the elements of Y_a).

We then sample N negative time steps at which we define hard negative and easy negative examples. An *easy negative* corresponds to a time step that is not a PLP peak. They are sampled uniformly in $[0, T] \setminus Y$. We also apply a “safety window” (whose duration was empirically determined to one frame) around peak time steps to avoid sampling negatives that are too close to a peak. A *hard negative* corresponds to a time step that is a peak but that is not in Y_a . They are sampled uniformly in $Y \setminus (Y_a \cup \{y_a\})$. We sample $N = 10$ negatives, half of them are hard negatives, and the other half are easy negatives.

To prevent any errors coming from the PLP function we discard audio segments where the inter-peak distance is not almost constant. We empirically set the allowed variation to 20 percent of this inter-peak distance within a segment (more details about this are given on the companion website).

3.2 Architectures

Front-end. We compute Mel spectrogram features from audio sampled at 16kHz using 128 bands, a window size of 2048 samples, and a hop size of 320 samples (20ms frame rate). We apply log compression and normalization⁵. Subsequently, a linear layer projects the frames to the embedding dimension. The resulting sequence x_t serves as the input to the encoder. We use audio segments of 20s long to ensure the model sees a sufficiently large context. However, we did not explore varying the length of audio segments fed into the encoder.

Encoder. For the encoder, we use a Transformer architecture similar to the one used in Wav2Vec2 or Hubert [10, 32]. It is composed of a stack of Transformer

⁴ In a preliminary experiment, we delve deeper into determining the metrical level that the peaks of the PLP correspond to. Our findings suggest that these peaks align with either the beat, the 8-th note or the 16-th note level.

⁵ For normalization, we use the mean and standard deviation computed over the training set.

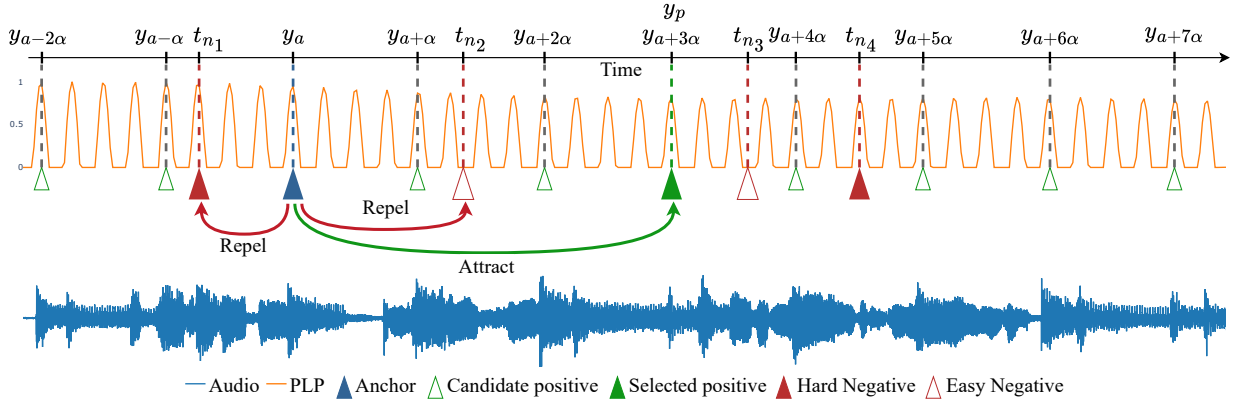


Figure 2: Proposed mining strategy of Positives and Negatives (easy and hard) given an Anchor time in the PLP function. Positive are sampled among peaks of the PLP whose time index is distant from the Anchor by a power of two tatum units tu (here $\alpha = 4 \times tu$); Negatives are the remaining times and are considered Easy if not peaks of the PLP and Hard if peaks of the PLP. Here we sample two hard and two easy negatives.

encoder layers. Each layer is composed of a multi-head self-attention mechanism followed by a feed-forward network. We use 8 layers each of which has 8 attention heads and apply a 0.1 dropout in the attention layer. The encoder outputs the embedding sequence z_t . The embedding dimension is set to 512 and the hidden dimension of the feed-forward network is set to 1024. In total, the model has 19.1M learnable parameters. We did not explore other architectures as the focus of our work was to study the proposed SSL scheme.

3.3 Contrastive Loss

Among the various formulations of the contrastive losses, we have chosen to use the NT-Xent loss [5] one. We define the similarity measure between two vectors u and v as $\text{sim}(u, v) = \frac{u^\top v}{\|u\| \|v\|}$. Given an anchor y_a , a positive y_p and a set of N negatives time-steps $t_{\text{neg}} = \{t_{n_1}, \dots, t_{n_N}\}$, we compute the contrastive loss as follows:

$$\mathcal{L}_{\text{NT-Xent}}(y_a, y_p, t_{\text{neg}}) = -\log \frac{\exp(\text{sim}(z_{y_a}, z_{y_p})/\tau)}{\sum_{i=1}^N \exp(\text{sim}(z_{y_a}, z_{t_{n_i}})/\tau)} \quad (1)$$

We set the temperature to $\tau = 0.1$. For each audio in a batch, we use 80% of the available peaks as anchors. For each of them, we sample their corresponding positive and negatives. We compute the above contrastive loss over each pair and each audio. We then average the losses to obtain the global loss for the batch, that is if we have a total of M pairs in the batch:

$$\mathcal{L} = \frac{1}{M} \sum_{y_a, y_p, t_{\text{neg}}} \mathcal{L}_{\text{NT-Xent}}(y_a, y_p, t_{\text{neg}}). \quad (2)$$

4 Evaluation

To evaluate our model, we performed three experiments. In all three experiments, the model is pre-trained in a SSL way using unlabeled data.

In Experiment 1, we test the Few-Shot Learning (FSL) abilities of our model using only a few data for fine-tuning.

Experiment 2 tests the generalization of our model on unseen conditions and serves as comparison to ZeroNS. Finally, Experiment 3 compares our performance to the ones obtained using fully-supervised beat-tracking models.

4.1 Datasets

For SSL pre-training, we use a combination of unlabeled datasets (in terms of beat positions): the Free Music Archive (FMA) [33], MTG-Jamendo [34], and MagnaTagaTune (MTT) [35]. FMA contains 106,574 full tracks spanning 161 genres. MTG-Jamendo contains around 55,000 full audio tracks. Finally, MTT contains approximately 26,000 excerpts of 30-s duration from 5223 unique tracks. Overall the combined datasets offer around 165k full audio tracks and a total of 8,000 hours.

For fine-tuning and testing, we used the following labeled (into beats) datasets, commonly used in previous works: SMC [36], Ballroom [37] and Hainsworth [38], GTZAN [39,40], RWC [41] and Harmonix [42]. The Harmonix dataset is mainly composed of pop music tracks, whereas the Ballroom, GTZAN, RWC, and Hainsworth datasets offer a wider variety of musical genres.

4.2 Evaluation Metrics

We report the commonly used metrics in the literature including the F-measure with a tolerance window of $\pm 70\text{ms}$, continuity-based measures at the correct metrical level (CMLt & CMLc), and at alternate metrical levels such as double/half and offbeat (AMLt & AMLc) [43].

4.3 Implementation details

4.3.1 Pre-training

For SSL pre-training we kept 0.05% of the data for validation (9,000 tracks). Our model is pre-trained during 200 epochs (equivalent to around 270,000 steps). Training was conducted on 4 A100 GPUs utilizing float 16 precision and a global batch size of 96. We employed the Adam optimizer [44] with an initial learning rate set at $1e-4$ and

applied a polynomial decay learning rate scheduler. The learning rate gradually increased to $5e-4$ within the first 32,000 steps, then reverted to its initial value over the subsequent 250k steps. Additionally, gradient clipping was employed. We keep the model that gives the best validation loss.

4.3.2 Fine-tuning

After SSL pre-training, we need to adapt the model to the downstream task of beat tracking. This is done by adding a linear classification probe $g(\cdot)$ and fine-tuning both the encoder and the linear probe. $g(\cdot)$ projects the embedding into the scalar beat activation function. Instead of feeding $g(\cdot)$ with the output of the encoder, we feed it with a weighted sum of the outputs of each layer of the Transformer [45]. That is $z = \sum_{l=1}^8 \alpha_l z^{(l)}$, where $z^{(l)}$ is the output of layer l . The weights α_l are jointly learned with the linear probe $g(\cdot)$.

The system is trained to minimize the binary cross-entropy loss between the beat activations and the target. Following the literature we widened the beat targets by a window [0.25, 0.5, 1, 0.5, 0.25] [26]. We used the Adam optimizer [44] with an initial learning rate of $1e-5$ and a polynomial decay learning rate scheduler.

During fine-tuning, we utilized audio chunks of sizes similar to those used during pre-training (20s). However, during inference, to avoid potential out-of-memory errors, we split audio excerpts exceeding 45 seconds into 20-second chunks with 5-second overlap. Subsequently, we overlap-add the activations to derive the beat activations for the whole track.

These beat activations are then fed into a Dynamic Bayesian Network (DBN) [46] to predict the beat positions. The DBN is configured to model a tempo range of 40-270 beats per minute with transition lambda set to 45, observation lambda to 9, and a threshold of 0.15.

4.3.3 Data Augmentation

We found that both pre-training and fine-tuning could benefit from data augmentation, in particular time-stretching. We apply time-stretching in two manners: constant factor and time-varying factor. In both cases, we constrain the time-stretching factor to lie in the interval [0.8, 1.2]. For the constant factor case, we used sox effects in TorchAudio [47], and for time-varying factor we used LibTSM [48]. When using a time-varying factor we randomly sample time instants at which the stretching factor is modified (also randomly, see the repository for details). This was found to be particularly beneficial for the pre-training stage. Indeed because we have filtered out tracks where the inter-peak distance is not almost constant, the SSL training data does not contain examples of time-varying tempo. Using time-varying time-stretching allows us to simulate this in a controlled fashion.

We found that it was better to compute the Predominant Local Pulse (PLP) curve before time-stretching and shift the peaks accordingly, rather than on the time-stretched audio.

4.4 Experiment 1: Few-shot learning

Protocol. The goal here is to test the ability of our model to learn with only few examples, Few-Shot Learning (FSL). To be able to compare our results with previously published ones, we replicate the evaluation protocol proposed in ZeroNS [11]. We consider *individually* each dataset (both for fine-tuning and testing): $T \in \{\text{SMC, Ballroom, Hainsworth, GTZAN}\}$. For each dataset T , we split it into 8 folds, we use one for testing T_{test} , one for validation T_{valid} and perform FSL with the remaining ones T_{train} . The FSL ability is evaluated by selecting randomly $k \in \{1, 2, 3, 4, 6, 8, 12, 16, 24, 48, 64, 96\}$ items from T_{train} . For each k we sample 10 variations: $T_{train,i}^k$.

For each choice of k , we fine-tune our pre-trained model on each variation $T_{train,i}^k$ and keep the one that performs the best on T_{valid} .

Results. We give the results in Figure 3 for the T_{test} of each dataset (SMC Mirex, Ballroom, Hainsworth, and GTZAN) and each value of k (x-axis). We report the mean and standard deviation of the metrics over the training set variations $T_{train,i}^k$. Our model performs at least as well as ZeroNS on almost all metrics and datasets. The exceptions are with AMLt on SMC and AMLc and AMLt on GTZAN and SMC. We observe that our model performs significantly better on Hainsworth, with up to 10% absolute improvement in F1 score and almost 20% absolute improvement in CMLt and CMLc. Also, the performance gap is significant on Ballroom when using very few data (less than 10 tracks) where we can observe almost 10% absolute improvement in F1 score and up to 15% improvement in AMLc.

4.5 Experiment 2: Generalization

Protocol. The goal here is to test the generalization ability of our model, *i.e.* training our model on one dataset and testing on another. For this, we replicate the protocol proposed in ZeroNS [11]. For each choice of dataset $T \in \{\text{SMC, Hainsworth, Ballroom}\}$, we split it into 8 folds, we use one for validation T_{valid} , and the remaining seven for training T_{train} . We then use the best-performing model on T_{valid} . Instead of using the linear probe described above, we obtained better results using a MLP (two linear layers interleaved with a ReLU), also fed by the weighted sum of layer sequences (sec 4.3.2). Whatever the choice of T , the test is performed on the GTZAN dataset.

Trained on	Method	F1 (%)	AMLt (%)	CMLt (%)
SMC	Ours	79.5 ± 0.5	88.0 ± 0.6	64.4 ± 0.9
	ZeroNS	74.8 ± 2.1	86.3 ± 2.3	51.0 ± 2.1
Hainsworth	Ours	85.1 ± 0.8	89.9 ± 0.9	73.2 ± 1.8
	ZeroNS	80.6 ± 0.9	89.4 ± 0.7	62.8 ± 2.3
Ballroom	Ours	83.9 ± 0.3	88.4 ± 0.5	72.3 ± 0.9
	ZeroNS	82.6 ± 0.5	89.0 ± 0.8	67.6 ± 1.1

Table 1: Results of Experiment 2: Generalization

Results. We indicate the results in Table. 1. We report the mean and standard deviation of F1, AMLt, and CMLt

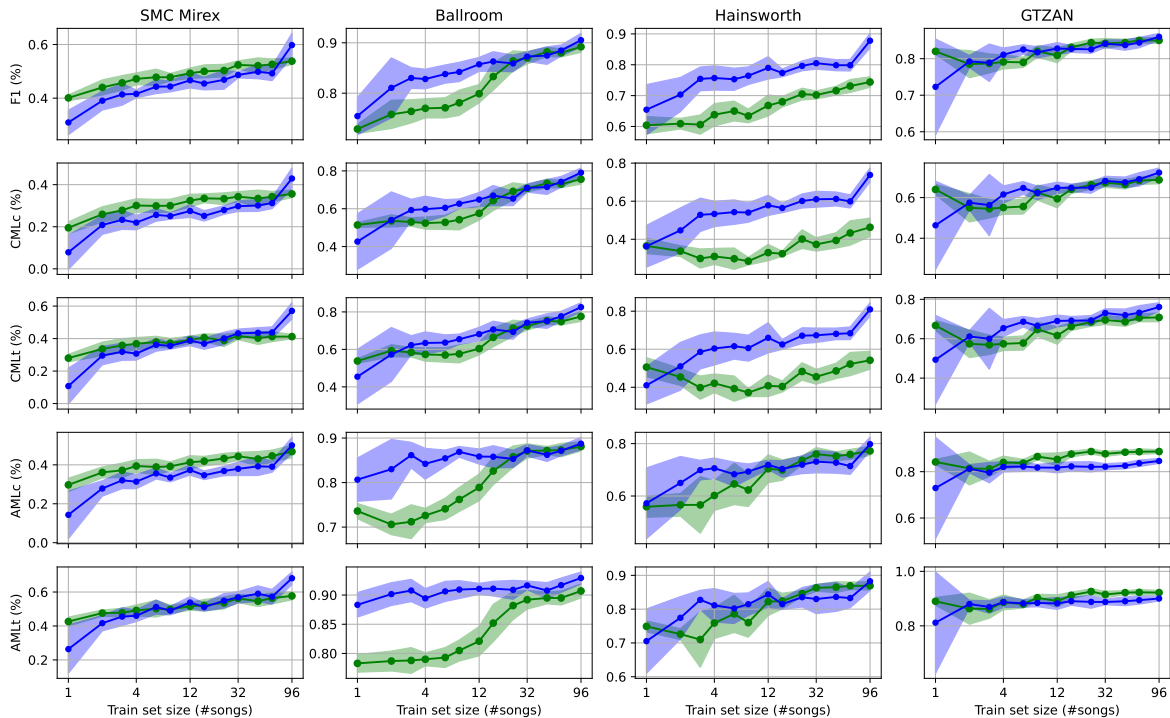


Figure 3: Results of Experiment 1: Few-Shot Learning. Shaded areas representation the standard deviation. (ZeroNS in green and our method in blue)

Method	F1	CMLt	AMLt
Böck [26]	0.885	0.813	0.931
Hung [27]	0.887	0.812	0.920
Zhao [29]	0.885	0.800	0.922
Ours	0.876	0.802	0.918

Table 2: Results of Experiment 3: Comparison with supervised baseline

scores across the different folds. Overall our model performs better than ZeroNS on all datasets except when for the AMLt metric when trained with Ballroom, but the difference is not statistically significant. This means that our model can generalize well to unseen data. Precisely we observe a 5% improvement in F1 score and more than 10% improvement in CMLt when training on SMC or Hainsworth. We nearly reach the F1 score of fully supervised models (presented next) when training solely on 7/8 of Hainsworth (*i.e* 194 tracks).

4.6 Experiment 3: Comparison with supervised baseline

Protocol. The goal here is to compare the performance of our model to the ones provided by fully-supervised models. For this we replicate the commonly used 8-fold cross validation set-up after [26,27,29]. GTZAN is kept as a test set and is never seen in training. We average the metrics over the 8 training folds to obtain the final results.

Results. We give the results in Table 2. It is clear that the proposed beat tracking approach using our self-supervised

pre-training can be competitive with state-of-the-art methods on GTZAN, a dataset covering a wide diversity of genres. While our method does not outperform the best-performing method, it achieves comparable results across all metrics, proving the quality of the learned representations.

5 Conclusion

In this paper, we proposed a novel Self-Supervised Learning approach to learn representations useful for the task of beat tracking using contrastive learning where the selection of anchor, positive and negative peaks derives from a Predominant Local Pulse function.

We assess our proposal positively based on a series of experiments. In a first experiment, we showed that our proposed approach was superior on some datasets to the previous SSL approach, ZeroNS, in a few-shot learning setting. In a second experiment, we show that our model has better generalization capabilities to unseen data. In the last experiment, we show that our model also yields comparable performances to the fully supervised baseline, indicating that our pre-training scheme effectively learns meaningful beat-related representations.

To further improve our method, future work will focus on developing a more sophisticated sampling mechanism that can handle other metrical structures than the binary one used-here (such as 6/8, 3/4). One potential approach is to incorporate additional audio features, such as self-similarity matrices, to gain a deeper understanding of the rhythmic structure within an audio segment and adaptively select positive positions for a given anchor.

6 Acknowledgements

This work was granted access to the HPC resources of IDRIS under the allocation 2022-AD011013924R1 made by GENCI. The material contained in this document is based upon work funded by the ANR-IA and Hi! PARIS.

7 References

- [1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, 2019.
- [2] K. He, X. Chen, S. Xie, Y. Li, P. Dollar, and R. Girshick, “Masked autoencoders are scalable vision learners,” in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022*, 06 2022, pp. 15 979–15 988.
- [3] H. Bao, L. Dong, S. Piao, and F. Wei, “BEit: BERT pre-training of image transformers,” in *International Conference on Learning Representations*, 2022.
- [4] P.-Y. Huang, H. Xu, J. B. Li, A. Baevski, M. Auli, W. Galuba, F. Metze, and C. Feichtenhofer, “Masked autoencoders that listen,” in *Advances in Neural Information Processing Systems*, 2022.
- [5] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, “A simple framework for contrastive learning of visual representations,” in *Proceedings of the 37th International Conference on Machine Learning*, vol. 119, 2020, pp. 1597–1607.
- [6] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar, B. Piot, k. kavukcuoglu, R. Munos, and M. Valko, “Bootstrap your own latent - a new approach to self-supervised learning,” in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 21 271–21 284.
- [7] X. Chen and K. He, “Exploring simple siamese representation learning,” in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, 2021, pp. 15 750–15 758.
- [8] K. He, H. Fan, Y. Wu, S. Xie, and R. B. Girshick, “Momentum contrast for unsupervised visual representation learning,” 2019.
- [9] A. van den Oord, Y. Li, and O. Vinyals, “Representation learning with contrastive predictive coding,” 2019.
- [10] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, “wav2vec 2.0: A framework for self-supervised learning of speech representations,” in *Advances in Neural Information Processing Systems*, 2020.
- [11] D. Desblancs, V. Lostanlen, and R. Hennequin, “Zero-note samba: Self-supervised beat tracking,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.
- [12] M. McCallum, F. Korzeniowski, S. Oramas, F. Gouyon, and A. Ehmann, “Supervised and unsupervised learning of audio representations for music understanding,” 10 2022.
- [13] Y. Li, R. Yuan, G. Zhang, Y. Ma, X. Chen, H. Yin, C. Lin, A. Ragni, E. Benetos, N. Gyenge, R. B. Dannenberg, R. Liu, W. Chen, G. Xia, Y. Shi, W. Huang, Y. Guo, and J. Fu, “MERT: acoustic music understanding model with large-scale self-supervised training,” *CoRR*, vol. abs/2306.00107, 2023. [Online]. Available: <https://doi.org/10.48550/arXiv.2306.00107>
- [14] R. Yuan, Y. Ma, Y. Li, G. Zhang, X. Chen, H. Yin, L. Zhuo, Y. Liu, J. Huang, Z. Tian, B. Deng, N. Wang, C. Lin, E. Benetos, A. Ragni, N. Gyenge, R. B. Dannenberg, W. Chen, G. Xia, W. Xue, S. Liu, S. Wang, R. Liu, Y. Guo, and J. Fu, “MARBLE: music audio representation benchmark for universal evaluation,” in *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., 2023.
- [15] B. Gfeller, C. Frank, D. Roblek, M. Sharifi, M. Tagliasacchi, and M. Velimirovic, “Spice: Self-supervised pitch estimation,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, p. 1118–1128, 2020.
- [16] E. Quinton, “Equivariant self-supervision for musical tempo estimation,” in *Proceedings of the 23rd International Society for Music Information Retrieval Conference, ISMIR 2022, Bengaluru, India, 2022*, 2022.
- [17] A. Riou, S. Lattner, G. Hadjeres, and G. Peeters, “Pesto: Pitch estimation with self-supervised transposition-equivariant objective,” 2023.
- [18] A. Gagneré, S. Essid, and G. Peeters, “Adapting pitch-based self supervised learning models for tempo estimation,” in *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024.
- [19] J. Jiang and G. Xia, “Self-supervised hierarchical metrical structure modeling,” in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.
- [20] D. Ellis, “Beat tracking by dynamic programming,” *Journal of New Music Research*, vol. 36, pp. 51–60, 03 2007.
- [21] G. Peeters, “Beat-marker location using a probabilistic framework and linear discriminant analysis,” 09 2009.

- [22] M. Alonso, B. David, and G. Richard, “Tempo and beat estimation of musical signals,” 10 2004.
- [23] S. Böck and M. Schedl, “Enhanced beat tracking with context-aware neural networks,” in *Proceedings of the 14th International Conference on Digital Audio Effects, DAFx 2011*, 09 2011.
- [24] E. P. MatthewDavies and S. Böck, “Temporal convolutional networks for musical audio beat tracking,” in *2019 27th European Signal Processing Conference (EUSIPCO)*, 2019, pp. 1–5.
- [25] S. Böck, M. E. P. Davies, and P. Knees, “Multi-task learning of tempo and beat: Learning one to improve the other,” in *International Society for Music Information Retrieval Conference*, 2019.
- [26] S. Böck and M. E. P. Davies, “Deconstruct, analyse, reconstruct: How to improve tempo, beat, and downbeat estimation,” in *International Society for Music Information Retrieval Conference*, 2020.
- [27] Y.-N. Hung, J.-C. Wang, X. Song, W.-T. Lu, and M. Won, “Modeling beats and downbeats with a time-frequency transformer,” in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.
- [28] W. Lu, J.-C. Wang, M. Won, K. Choi, and X. Song, “Spectnt: a time-frequency transformer for music audio,” in *International Society for Music Information Retrieval Conference*, 2021.
- [29] J. Zhao, G. Xia, and Y. Wang, “Beat transformer: Demixed beat and downbeat tracking with dilated self-attention,” in *Proceedings of the 23rd International Society for Music Information Retrieval Conference, ISMIR 2022, Bengaluru, India, December 4-8, 2022*, 2022.
- [30] P. Grosche and M. Muller, “Extracting predominant local pulse information from music recordings,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 6, pp. 1688–1701, 2011.
- [31] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, “librosa: Audio and music signal analysis in python,” in *Proceedings of the 14th python in science conference*, vol. 8, 2015.
- [32] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, “Hubert: Self-supervised speech representation learning by masked prediction of hidden units,” *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, vol. 29, p. 3451–3460, oct 2021.
- [33] M. Defferrard, K. Benzi, P. Vandergheynst, and X. Bresson, “FMA: A dataset for music analysis,” in *18th International Society for Music Information Retrieval Conference (ISMIR)*, 2017.
- [34] D. Bogdanov, M. Won, P. Tovstogan, A. Porter, and X. Serra, “The mtg-jamendo dataset for automatic music tagging,” in *Machine Learning for Music Discovery Workshop, International Conference on Machine Learning (ICML 2019)*, 2019.
- [35] E. Law, K. West, M. I. Mandel, M. Bay, and J. S. Downie, “Evaluation of algorithms using games: The case of music tagging,” in *International Society for Music Information Retrieval Conference*, 2009.
- [36] A. Holzapfel, M. E. P. Davies, J. R. Zapata, J. L. Oliveira, and F. Gouyon, “Selective sampling for beat tracking evaluation,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 9, pp. 2539–2548, 2012.
- [37] F. Gouyon, A. Klapuri, S. Dixon, M. Alonso, G. Tzanetakis, C. Uhle, and P. Cano, “An experimental comparison of audio tempo induction algorithms,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 5, pp. 1832–1844, 2006.
- [38] S. W. Hainsworth and M. D. Macleod, “Particle filtering applied to musical tempo tracking,” *EURASIP J. Adv. Signal Process.*, vol. 2004, no. 15, pp. 2385–2395, 2004.
- [39] G. Tzanetakis and P. Cook, “Musical genre classification of audio signals,” *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, pp. 293–302, 2002.
- [40] U. Marchand and G. Peeters, “Swing Ratio Estimation,” in *Digital Audio Effects 2015 (Dafx15)*, Trondheim, Norway, Nov. 2015.
- [41] M. Goto, H. Hashiguchi, T. Nishimura, and R. ichi Oka, “Rwc music database: Popular, classical and jazz music databases,” in *International Society for Music Information Retrieval Conference*, 2002.
- [42] O. Nieto, M. McCallum, M. Davies, A. Robertson, A. Stark, and E. Egozy, “The Harmonix Set: Beats, Downbeats, and Functional Segment Annotations of Western Popular Music,” in *Proceedings of the 20th International Society for Music Information Retrieval Conference*, 2019, pp. 565–572.
- [43] M. Davies, N. Degara Quintela, and M. Plumbley, “Evaluation methods for musical audio beat tracking algorithms,” 2009.
- [44] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015.
- [45] S. wen Yang, P.-H. Chi, Y.-S. Chuang, C.-I. Lai, K. Lakhotia, Y. Y. Lin, A. T. Liu, J. Shi, X. Chang, G.-T. Lin, T. hsien Huang, W.-C. Tseng, K. tik Lee, D.-R. Liu, Z. Huang, S. Dong, S.-W. Li, S. Watanabe,

- A. rahman Mohamed, and H. yi Lee, “Superb: Speech processing universal performance benchmark,” in *Interspeech*, 2021.
- [46] F. Krebs, S. Böck, and G. Widmer, “An Efficient State-Space Model for Joint Tempo and Meter Tracking.” in *Proceedings of the 16th International Society for Music Information Retrieval Conference*, 2018.
- [47] J. Hwang, M. Hira, C. Chen, X. Zhang, Z. Ni, G. Sun, P. Ma, R. Huang, V. Pratap, Y. Zhang, A. Kumar, C.-Y. Yu, C. Zhu, C. Liu, J. Kahn, M. Ravanelli, P. Sun, S. Watanabe, Y. Shi, Y. Tao, R. Scheibler, S. Cornell, S. Kim, and S. Petridis, “Torchaudio 2.1: Advancing speech recognition, self-supervised learning, and audio processing components for pytorch,” 2023.
- [48] J. Driedger and M. Müller, “TSM Toolbox: MATLAB implementations of time-scale modification algorithms,” in *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, Erlangen, Germany, 2014, pp. 249–256.

USING PAIRWISE LINK PREDICTION AND GRAPH ATTENTION NETWORKS FOR MUSIC STRUCTURE ANALYSIS

Morgan Buisson¹ Brian McFee^{2,3} Slim Essid¹

¹ LTCI, Télécom Paris, Institut Polytechnique de Paris, France

² Music and Audio Research Laboratory, New York University, USA

³ Center for Data Science, New York University, USA

ABSTRACT

The task of music structure analysis has been mostly addressed as a sequential problem, by relying on the internal homogeneity of musical sections or their repetitions. In this work, we instead regard it as a pairwise link prediction task. If for any pair of time instants in a track, one can successfully predict whether they belong to the same structural entity or not, then the underlying structure can be easily recovered. Building upon this assumption, we propose a method that first learns to classify pairwise links between time frames as belonging to the same section (or segment) or not. The resulting link features, along with node-specific information, are combined through a graph attention network. The latter is regularized with a graph partitioning training objective and outputs boundary locations between musical segments and section labels. The overall system is lightweight and performs competitively with previous methods. The evaluation is done on two standard datasets for music structure analysis and an ablation study is conducted in order to gain insight on the role played by its different components.

1. INTRODUCTION

Music structure analysis consists of locating segments that compose a track and grouping them into semantic categories, referred to as musical sections [1]. Approaches to solve this task have significantly been advanced in the past few years, notably due to the creation of large audio datasets along with their structural annotations [2–4]. These annotated corpora have allowed researchers to leverage recent progress in deep learning and design systems that learn signal representations to predict song structures.

1.1 Related work

One crucial aspect when analyzing musical structures is the strong temporal dependency among different events

within a track. A musical observation at a given time can impact other observations at any other point in time, and this, at different scales. This multi-level dependency still poses a significant challenge when training music segmentation systems [1]. Recent methods successfully relied on modelling these temporal connections through the use of self-attention mechanism [5–8]. In these cases, the model is equipped with multiple self-attention layers so as to automatically learn to identify such dependencies. While these proved to be effective, they do not rely on any prior knowledge about musical structure and therefore tend to require large training sets or multiple input audio representations (*e.g.* multiple audio features [8], separated instrument stems [7]) so as to better characterize mutual relationships between time instants in the input track. The method introduced in this work proposes to explicitly model such temporal dependencies by leveraging the natural geometry of a track’s self-similarity matrix.

Self-similarity representations have been a useful tool to predict the structure of a track [9–12]. A line of work has for example focused on improving this representation through the use of contrastive learning [13–15]. By extracting better audio features, the resulting self-similarity matrices carry more meaningful patterns that can ease structure prediction performed by downstream segmentation methods [10, 16]. In the proposed approach, the self-similarity representation is not used as direct input to a segmentation system but rather to extract structural link features between time frames within the input track. This step is jointly performed with the audio feature extraction stage, the prediction of segment boundaries and section labels, allowing each task to benefit from the others.

1.2 Contributions

In this work, a supervised approach to segmentation of western popular music is proposed that effectively combines the three music structure principles which have been identified in previous studies [1]: *homogeneity*, *repetition* and *regularity*. To this end, the segmentation task is formulated as a graph partitioning problem where links (*i.e.* edges) between musical audio observations taken at any two time instants (*i.e.* nodes) are first characterized as whether connecting elements from the same segment, section or distinct structural entities (different segment or sec-



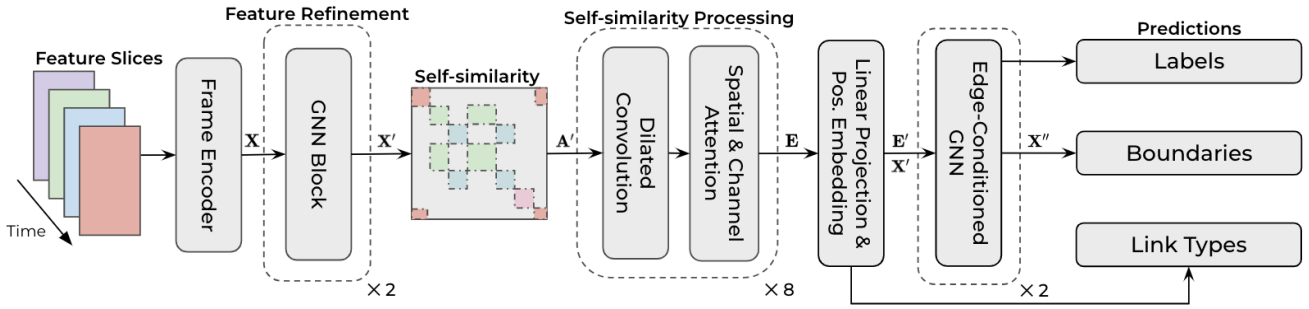


Figure 1. Model overview. Input feature patches are first processed through a frame-level encoder and a first GNN block. A self-similarity of the output features is passed through a 2-dimensional dilated convolutional network to extract link features. Node and link features are finally combined with graph attention network layers to predict boundary locations and section label assignments.

tion). These extracted link features condition a subsequent analysis block based on attention graph neural networks to further refine node features. The system outputs final predictions composed of boundary locations and frame-wise section label likelihoods. Overall, the main contributions of this work are the following: (1) we demonstrate that music segmentation can be modelled as a pairwise link prediction task, which offers a flexible framework that is inspired by well-identified structure principles ; (2) we use graph attention networks to allow frames in the track to dynamically exchange information between each other and we successfully inform this process by the learned link features ; (3) we demonstrate in an ablation study that link features provide some useful structural information about the input track, which significantly improves segmentation performance.

2. METHOD

2.1 Overall approach

The segmentation method proposed in this work proceeds in three main steps, depicted in Figure 1. First, the input track is passed through a frame encoder to obtain a sequence of frame-wise feature vectors. These are further smoothed by a graph neural network (GNN) block, allowing each individual frame to aggregate and combine information from all other time instants in the track. A self-similarity matrix is calculated from these features and fed as input to a 2-dimensional convolutional neural network. A spatial learnable bias is added to the output feature map to inform about each component’s source and destination frames’ relative positions. The link features, along with the smoothed frame features, are effectively combined through an edge-conditioned graph attention module. The updated frame features finally serve to predict segment boundaries and section labels.

2.2 Audio representation

2.2.1 Input features

For a given track, we start by estimating probable beat positions using an off-the-shelf beat tracking algorithm so as

to reduce the length of the feature sequence to be analyzed. Following previous work [14, 15, 17], the input signal is then converted into a log-scaled Mel-spectrogram representation, from which slices centered around each detected beat position are extracted.

2.2.2 Frame encoder

The sequence of mel-spectrogram slices is passed through an encoder to obtain a sequence of feature vectors $\mathbf{X} \in \mathbb{R}^{N \times d}$ where N is the number of detected beats (*i.e.* slices) and d is the embedding dimension. The objective of this step is to extract relevant spectro-temporal information from each slice. The architecture of the encoder is inspired from the work by Won *et al.* [18] for music tagging. It consists of three convolutional blocks to extract low-level features, followed by two transformer encoder layers which temporally summarize the content of each slice. To obtain more robust audio representations, the pre-training strategy proposed by Buisson *et al.* [15] is followed. It uses a contrastive loss to learn an embedding space in which frames from repeating sequences over the whole track are close. In this work, the self-supervised pre-training stage is performed on 20,000 unlabelled tracks, covering various music genres such as rock, popular, rap, jazz, electronic or classical.

2.3 Feature refinement

The sequence of feature vectors \mathbf{X} is processed by a first GNN block. The objective is to further refine local discontinuities by allowing each frame to exchange information with all other frames in the track. To this end, a self-similarity matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$ is calculated from \mathbf{X} such that its elements $A(i, j)$ are defined as:

$$A(i, j) = \exp \left(-\gamma \left\| \frac{x_i}{\|x_i\|_2} - \frac{x_j}{\|x_j\|_2} \right\|_2^2 \right), \quad (1)$$

where the bandwidth parameter γ is simply set as $\gamma = \frac{1}{2s}$, with $s = \text{std} \left(\left\| \frac{x_i}{\|x_i\|_2} - \frac{x_j}{\|x_j\|_2} \right\|_2^2 \right)$ and $\|\cdot\|_2$ denotes the ℓ_2 -norm. The matrix \mathbf{A} can be regarded as the weighted adjacency matrix of a complete graph $G = (V, E)$, where

the set of nodes V corresponds to each frame contained in the track, and its edges E represent the strength of their mutual connections (*i.e.* similarity). However, each feature slice was transformed independently by the frame encoder (see Section 2.2.2). To improve both segment homogeneity and discriminability, two graph convolution layers [19] are applied to smooth the node features \mathbf{X} , of which the update rule for an arbitrary layer l is expressed as:

$$x_i^{(l+1)} = \sigma \left(\sum_{1 \leq j \leq N} \frac{A(j, i)}{N} x_j^{(l)} \mathbf{W}^{(l)} + b^{(l)} \right), \quad (2)$$

where $\mathbf{W}^{(l)}$ and $b^{(l)}$ are learnable weight parameters and σ is an activation function: Exponential Linear Unit (ELU) in this work. Equation (2.3) shows that each frame in the sequence \mathbf{X} receives a weighted combination of all other frames in the track and is then linearly transformed before applying a non-linear activation. A common issue encountered with graph neural networks is the over-smoothing phenomenon [20], where all points end up having the same representation after passing through several layers. To limit this effect, the output features are further processed by a multi-layer perceptron (MLP) [21], yielding the refined node features $\mathbf{X}' \in \mathbb{R}^{N \times d}$.

2.4 Link feature extraction

2.4.1 Motivations

Recognizing the structure of a song can be achieved by learning to fully characterize the mutual relationships between time frames from its beginning to its end. In other words, if for any pair of time points (*i.e.* audio frames), one can successfully predict if they belong to the same musical segment or section, then the overall structure of the song can be easily recovered (*e.g.* through a simple graph traversal). Figure 2 shows a visual representation of this link prediction task.

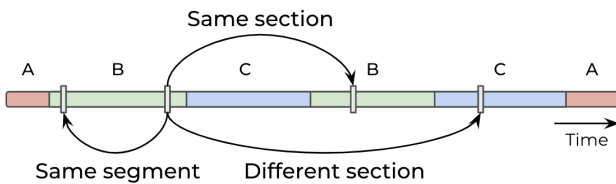


Figure 2. Schematic representation of the link characterization task. For each time instant, the goal is to classify its mutual relationship with all other instants in the input track as either, from the same segment, section or a different section.

It is interesting to notice that each of the structure principles somehow translates into specific characteristics of the self-similarity matrix. The *homogeneity* of musical segments can be observed through the appearance of block-like structures on the main diagonal. Similarly, *repetitions* of sequences can be spotted by diagonal stripes whereas repeating homogeneous segments will appear as off-diagonal blocks. The notion of *regularity* is visible as

the relative size of these patterns, which tends to be consistent within a track and in specific genres such as western popular music. Therefore, the self-similarity representation of a track yields crucial information on its structural organization and can be exploited to extract link-related information. Additionally, it provides an efficient information bottleneck which can improve generalization across different songs.

2.4.2 Self-similarity processing

The refined features $\mathbf{X}' \in \mathbb{R}^{N \times d}$ returned by the first GNN block are used to build a self-similarity matrix $\mathbf{A}' \in \mathbb{R}^{N \times N}$, in the same fashion as in Section 2.3. The goal of the link feature extraction step is to classify each component of the input matrix \mathbf{A}' into three categories: “same-segment”, “same-section” or “different section” links (see Figure 2). To this end, a 2-dimensional convolutional neural network is used, which is composed of blocs as shown in Figure 3. The kernels’ dilation rate is increased expo-

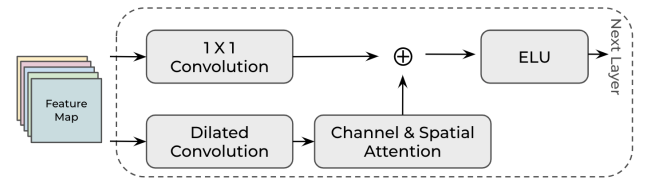


Figure 3. CNN block containing dilated convolution, channel & spatial attention and a residual connection. Each layer contains $C = 16$ channels and has an exponentially increasing dilation rate ranging from 2^0 to 2^7 to efficiently combine structural features at different scales.

nentially at each layer to enlarge the receptive field of the network and capture structural patterns at different scales. Because the goal is to classify each pixel (*i.e.* link), no pooling is applied in-between layers. To further enhance the intermediate feature maps, each convolution is followed by a multi-scale attention module [22] that leverages both spatial and channel interactions. A residual connection is added to the output of each attention block before applying ELU activation. We denote the output feature map as $\mathbf{E} \in \mathbb{R}^{C \times N \times N}$, with C being the number of convolution channels.

2.4.3 Positional embedding

If two frames are similar, then determining whether the pair is “same-segment” or “same-section” may be difficult without information about their position in the piece. To address this ambiguity, a learnable relative positional embedding $\mathbf{B} \in \mathbb{R}^{C \times N}$ is added to \mathbf{E} . Formulated as a function of $|i - j|$, it aims to provide each element in \mathbf{E} information on its relative distance to the main diagonal. We opt for a simple strategy which consists, for a given link between nodes i and j , in adding the $(|i - j|)$ th vector from a learnable embedding matrix \mathbf{B} to $e_{i,j}$. After doing so for every possible links, the result of this addition denoted as \mathbf{E}' , is fed to a linear layer with softmax activation. The link feature extraction network is optimized

using a cross-entropy loss function $\mathcal{L}_{\text{Link}}$ between the link-wise predictions \hat{y}_{link} and the ground-truth y_{link} obtained from structural annotations.

2.5 Edge-conditioned graph attention sub-network

The refined node features \mathbf{X}' and the edge features \mathbf{E}' provide a detailed representation of the input track. The former contains relevant acoustic information, which has been exchanged between frames for better discriminability across musical segments, while the latter provides information about their pairwise links. To efficiently combine these complementary views of the graph, we propose the use of edge-conditioned graph attention networks [23]. Node features are further improved by aggregating information from all other nodes in the graph, weighted by some learnable attention coefficients. These attention coefficients depend on each node’s features and the edge features that link them. For a given node x'_i , the update rule is defined as:

$$x''_i = \mathbf{W}_s \cdot x'_i + \sum_{j \in \mathcal{N}(x'_i)} \alpha_{j,i} (\mathbf{W}_n \cdot x'_j + \mathbf{W}_e \cdot e'_{j,i}), \quad (3)$$

where \mathbf{W} is used to denote learnable weight matrices for the transformation the node features to update (s=“self”), neighboring nodes (n=“neighbor”) and edge features (e=“edge”). The attention coefficients $\alpha_{j,i}$ are obtained as follows:

$$\alpha_{j,i} = \text{softmax}_i \left(\sigma \left(a^T [\mathbf{W}_n \cdot x'_i \parallel \mathbf{W}_n \cdot x'_j \parallel \mathbf{W}_e \cdot e'_{j,i}] \right) \right), \quad (4)$$

with a corresponding to a learnable vector, σ to a LeakyRelu activation, \parallel denotes the concatenation operation and $e'_{j,i}$ is the refined link features going from node j to node i . The softmax_i operation normalizes all incoming edges of node i . The forward-pass formulation from Equation (3) closely resembles that of the transformer, but additionally introduces edge features to calculate attention maps and output node features. We use a series of two graph attention layers with residual connections and ELU activation in-between. Both layers use 8 attention heads, the outputs of all heads are concatenated after the first layer and averaged after the second. The output node features are denoted as $\mathbf{X}'' \in \mathbb{R}^{N \times d}$.

2.6 Boundary and label predictions

The output of the overall system consists in boundary locations, expressed in beat indices, along with frame-wise section-label likelihoods. For boundary prediction, consecutive node features x''_i and x''_{i+1} are first concatenated, along with the corresponding link features $e'_{i,i+1}$ between them. The result of this concatenation is transformed through a linear layer with sigmoid activation to output the probability \hat{y}_{bound} of a segment boundary between these frames. For section-label predictions, we simply feed each frame to a linear layer with softmax activation, resulting in a predicted class assignment matrix $\mathbf{S} \in [0, 1]^{N \times K}$, where K corresponds to the number of section labels. We derive a boundary curve by concatenating the boundary predictions

\hat{y}_{bound} over time. To obtain the final boundary locations, we use the peak picking method after Ullrich *et al.* [24] without any thresholding on the RWC-Pop dataset, and the one from Kim *et al.* [7] for Harmonix. For the section label assignment, a simple majority vote is applied within each detected segment to determine its structural label. Due to the imbalance between boundary and non-boundary points, we use a dice loss $\mathcal{L}_{\text{Bound}}$ to optimize the boundary predictions, as it has proven useful in many segmentation tasks before [25]. We use a cross-entropy loss $\mathcal{L}_{\text{Label}}$ for section label predictions.

2.7 MinCut regularization

The objective of the proposed segmentation system is to assign each frame of the input track to one of K possible section labels. Ideally, we want this assignment to be equal for nodes in the graph that are either in the same segment or section, and orthogonal in the remaining cases. From the perspective of graph theory, given the input graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, this problem comes down to partitioning the set of nodes \mathcal{V} into K disjoint subsets by removing a minimum volume of edges, which is equivalent to maximizing:

$$\frac{1}{K} \sum_{k=1}^K \frac{\text{links}(\mathcal{V}_k)}{\text{degree}(\mathcal{V}_k)} = \frac{1}{K} \sum_{k=1}^K \frac{\sum_{i,j \in \mathcal{V}_k} \mathcal{E}_{i,j}}{\sum_{i \in \mathcal{V}_k, j \in \mathcal{V} \setminus \mathcal{V}_k} \mathcal{E}_{i,j}}, \quad (5)$$

where the numerator corresponds to the volume of edges within each cluster, and the numerator counts the edges between the nodes in a cluster and the rest of the graph. This task is referred to as the K -way *normalized MinCut* problem. Spectral clustering provides an optimal solution of this problem by projecting the nodes into the Laplacian eigenspace [10, 26]. However, calculating the spectrum of the Laplacian matrix is a costly operation and the final class assignment relies on non-differentiable operations, thus preventing it from being optimized along with the rest of the network.

In order to learn a model that finds an approximate spectral clustering solution in a differentiable manner, we base ourselves on the work by Bianchi *et al.* [27]. They propose a continuous relaxation of the normalized MinCut problem, where a GNN is trained to compute a cluster assignment matrix $\mathbf{S} \in [0, 1]^{N \times K}$ by optimizing the objective defined as:

$$\mathcal{L}_{\text{MinCut}} = -\frac{\text{Tr}(\mathbf{S}^T \mathbf{A} \mathbf{S})}{\text{Tr}(\mathbf{S}^T \mathbf{D} \mathbf{S})} + \left\| \frac{\mathbf{S}^T \mathbf{S}}{\|\mathbf{S}^T \mathbf{S}\|_F} - \frac{\mathbf{I}_K}{\sqrt{K}} \right\|_F, \quad (6)$$

where $\mathbf{A} \in \mathbb{R}^{N \times N}$ is the graph adjacency matrix, \mathbf{D} is the degree matrix of \mathbf{A} , K is the number of classes and $\|\cdot\|_F$ corresponds to the Frobenius norm. The left-hand-side term encourages connected nodes to be clustered together. It reaches its minimum when $\text{Tr}(\mathbf{S}^T \mathbf{A} \mathbf{S}) = \text{Tr}(\mathbf{S}^T \mathbf{D} \mathbf{S})$, meaning that the cluster assignments are equal for all the nodes in the same class and orthogonal to the cluster assignments of nodes from different classes. To avoid degenerate minima (uniform cluster assignments or all nodes being assigned to the same cluster), the right-hand-side term encourages the cluster assignments to be orthogonal

and the clusters to be of similar size. While in practice, it is not always desirable to have a perfectly balanced cluster assignment for music segmentation (due to the variable sizes of musical sections), the loss term $\mathcal{L}_{\text{MinCut}}$ acts as an effective regularizer that helps making the cluster assignment sharper. During training, we use the label agreement matrix \mathbf{Y} of each track as adjacency matrix and the predicted label assignment matrix \mathbf{S} defined in Section 2.6 as \mathcal{A} and \mathcal{S} in Equation (2.7) respectively. The whole system is trained end-to-end in a multi-task fashion, so as to minimize the overall loss function $\mathcal{L}_{\text{total}}$ defined as:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{Bound}} + \mathcal{L}_{\text{Label}} + \mathcal{L}_{\text{Link}} + \mathcal{L}_{\text{MinCut}}. \quad (7)$$

3. EXPERIMENTAL SETTING

3.1 Datasets

The proposed method is assessed on two standard datasets for music structure analysis. To reduce the number of possible section labels, we apply the annotation pre-processing step proposed in the work by Wang *et al.* [5]. We end up with a total of 7 unique section labels for both of the following datasets:

RWC-Pop: the Popular subset of the RWC dataset [28] contains 100 songs with section annotations. The original ones provided by the authors (AIST) are used.

Harmonix: the Harmonix dataset [3] is composed of 912 annotated tracks covering various genres of western popular music such as pop, electronic, hip-hop, rock, country and metal. The audio files were retrieved from YOUTUBE and structural annotations were manually adjusted.

3.2 Evaluation metrics

Common evaluation metrics for automatic structure analysis are employed throughout our experiments. For boundary detection, we report the F-measure¹ of the trimmed² boundary detection hit-rate with a 0.5 and 3-second tolerance windows (HR.5F, HR3F respectively). For structural grouping, we report the F-measure of pairwise-frame clustering [30] (PFC) and the F-measure of the normalized conditional entropy (NCE). We additionally measure the weighted label prediction accuracy (Acc), which indicates how well the model predicts frame-wise section labels.

3.3 Implementation details

All tracks are resampled at 22.05 kHz. As input to the frame encoder, we use log-scaled Mel-spectrograms with a window and hop size of 1024 and 256 samples respectively. We compute 64 Mel-bands per frame. The *TorchAudio* library is used for feature extraction [31]. As in previous work [15, 32], beats are estimated for all tracks using the algorithm from Korzeniowski *et al.* [33] implemented in the *madmom* package [34]. Slices of 64 frames

(≈ 0.75 s) centered at each detected beat location are fed as input to the frame encoder. The frame embedding dimension is set to $d = 32$ and kept fixed throughout the whole system. The number of channels in the link-feature extractor is set to $C = 16$, convolutions use kernels of size $k = 5$. All GNN layers are implemented using the *Deep Graph Library* [35] package. The whole model, including the pre-trained frame encoder, contains less than 330K parameters and is implemented³ with Pytorch 2.0 [36].

3.4 Experiments

In order to study the impact of each part of our method, we perform an 8-fold cross-validation ablation study on the Harmonix dataset. At each episode, one element from the system is removed: the pre-training stage (Section 2.2.2), the feature smoothing step (Section 2.3), the link features extraction (Section 2.4.2), the positional embedding (Section 2.4.3) and the MinCut regularization (Section 2.7). We use 6 splits for training, one for validation and the remaining one for testing. Then, we perform a cross-dataset evaluation, where one dataset is used for training (split beforehand into training and validation sets) and the other one for testing.

4. RESULTS

4.1 Ablation study

Results from the ablation study given in Figure 4 show the performance of the system when some of its components are discarded during training and inference. The different metrics are averaged over the 8 splits. In the first scenario, the frame encoder is randomly initialized like the rest of the model. We observe a significant decrease on all metrics, showing that the pre-training stage provides robust initial frame representations which are further tuned by the network during training. It is interesting to notice however that without pre-training the frame encoder, the model still predicts a good label assignment matrix, both in terms of pairwise frame clustering and label accuracy. We assume that the impact of this step is rather limited due to the relatively large size of the Harmonix dataset, which provides enough training examples to still learn useful frame features. In the second case, the MinCut regularization is discarded, which negatively impacts all metrics. This tends to confirm that the MinCut regularization enforces sharper cluster assignments, especially around segment boundaries where these can be more evenly distributed.

The most significant variation in segmentation performance is observed when the link feature extraction step is omitted. In this case, pairwise links between nodes are only characterized by their positional embedding and do not contain any structural information. This observation strongly suggests that the model benefits from both perspectives (node and link features) of the input track. When positional embeddings are removed, the link loss $\mathcal{L}_{\text{Link}}$ stops decreasing after several training iterations. Notably

¹ All evaluations are done using the *mir_eval* package [29].

² The first and last boundaries are discarded during evaluation, as they correspond to the beginning and the end of the track and therefore, do not provide any information regarding the system's performance.

³ Code: github.com/morgan76/LinkSeg

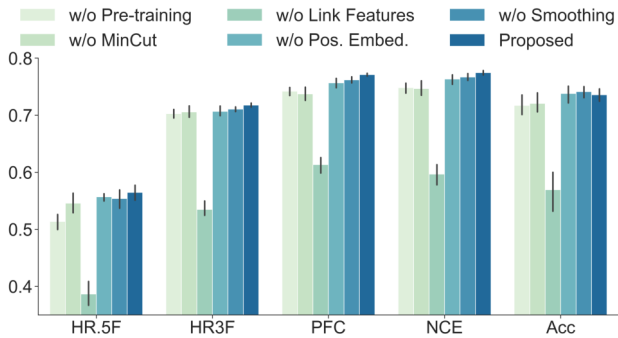


Figure 4. Ablation results on the Harmonix dataset in the cross-validation setting. Metrics are averaged across splits and standard deviation denoted with dark grey vertical bars.

because the network fails to differentiate “same-segment” from “same-section” links, which provides useful structural information near segment boundaries. In the case when the features smoothing step is removed, performance on all metrics, except the label prediction accuracy, is negatively impacted.

4.2 Comparison with previous work

This section compares the performance of our system against recent work for music structure analysis. The first one from Wang *et al.* [14], which we denote as DSF, uses supervised metric learning and spectral clustering [10] for boundary and section label predictions. SpecTNT [5] is based on a spectrogram transformer architecture and directly outputs both a boundary probability curve along with a frame-wise section label assignment. All in One [7] uses demixed spectrograms and several layers of neighborhood attention, operating simultaneously at the instrument and the temporal levels. These three baselines were trained and evaluated on the Harmonix dataset in a cross-validation setting. CBM, for Convolutional Block Matching [37] relies on dynamic programming to find the segmentation that minimizes a cost function. Its parameters were set by cross-validation on the RWC-Pop dataset.

Results on Harmonix and RWC-Pop are given in Table 1. In the cross-validation setting, the proposed method performs worse for boundary detection than the reported baselines. On RWC-Pop, despite being trained on a very small number of tracks (75 at each episode), the model still manages to pick up transitions between structural elements, even so at a high temporal resolution (± 0.5 second). This is to be compared with the first two baselines, namely DSF [14] and SpecTNT [5] which used the whole Harmonix dataset for training, along with additional datasets for the latter. The CBM algorithm [37] shows the strongest performance in this setting, as it explicitly favors musical segments of pre-defined length (which is around 8 bars in most cases for RWC-Pop), whereas our method does not make any assumption on the distribution of section lengths. It is also important to note that our system operates on a rather coarse time resolution (beat level) and only requires

a gross discretization of the input track’s timeline to function. We argue that better performance could be achieved by providing a more fine-grained time division (tatum level for example) but at a higher computational cost.

	HR.5F	HR3F	PFC	NCE	Acc
Harmonix					
DSF [14]	.497	.738	.689	.743	—
SpecTNT _{24s} [5]	.570	—	.700	.714	.701
SpecTNT _{36s} [5]	.558	—	.712	.724	.723
All in One [7]	.660	—	.738	.769	—
<i>Cross-val.</i>	.568	.717	.771	.772	.742
<i>Cross-dataset</i>	.462	.664	.660	.671	.530
RWC-Pop					
DSF [14]	.438	.653	.704	.739	—
SpecTNT _{24s} [5]	.623	—	.749	.728	.675
CBM [37]	.644	.806	—	—	—
<i>Cross-val.</i>	.585	.750	.785	.802	.813
<i>Cross-dataset</i>	.648	.786	.812	.812	.747

Table 1. Boundary detection and structural grouping results on Harmonix dataset. *Cross-val* indicates results that were obtained through cross-validation, averaged across splits. *Cross-dataset* refers to the results obtained when the model is trained on one and tested on the other.

In terms of structural grouping, the method proposed in this work outperforms all baselines on both datasets in most settings. Even though the label prediction method employed is rather simple and directly dependent on the boundary detection results, the model successfully learns to group frames across repetitions of identical musical sections. Finally, the high section label prediction accuracies obtained show that the network not only manages to successfully group frames together, but also predicts the right section label in a vast majority of cases.

Finally, cross-dataset results from RWC-Pop to Harmonix (*Cross-dataset* row) show that the model still generalizes to some extent, despite the very small quantity data used for training. On the other hand, training the model on Harmonix and testing it on RWC-Pop leads to strong performance both in terms of boundary detection and structural grouping, indicating that the network’s generalization capacity increases as more annotated data is available for training.

5. CONCLUSION

This work proposes a new approach to music segmentation by learning to characterize pairwise relationships between time instants in a musical recording. The structural view of the input track obtained from this auxiliary task can be combined with local frame information to effectively predict boundary locations between musical segments and section labels. Future research includes the extension of the link prediction task to various levels of segmentation and arbitrary labels semantic.

6. REFERENCES

- [1] O. Nieto, G. J. Mysore, C.-i. Wang, J. B. Smith, J. Schlüter, T. Grill, and B. McFee, “Audio-based music structure analysis: Current trends, open challenges, and applications.” *Transactions of the International Society for Music Information Retrieval (TISMIR)*, vol. 3, no. 1, pp. 246–263, 2020.
- [2] J. B. L. Smith, J. A. Burgoyne, I. Fujinaga, D. De Roure, and J. S. Downie, “Design and creation of a large-scale database of structural annotations.” in *ISMIR*, 2011.
- [3] O. Nieto, M. C. McCallum, M. E. Davies, A. Robertson, A. M. Stark, and E. Egozy, “The harmonix set: Beats, downbeats, and functional segment annotations of western popular music.” in *ISMIR*, 2019.
- [4] S. Balke, J. Reck, C. WEIS, J. ABESER, and M. Müller, “Jsd: A dataset for structure analysis in jazz music,” *Transactions of the International Society for Music Information Retrieval (TISMIR)*, vol. 5, p. 1, 2022.
- [5] J.-C. Wang, Y.-N. Hung, and J. B. Smith, “To catch a chorus, verse, intro, or anything else: Analyzing a song with structural functions,” in *ICASSP*, 2022.
- [6] G. Peeters, “Self-similarity-based and novelty-based loss for music structure analysis,” in *ISMIR*, 2023.
- [7] T. Kim and J. Nam, “All-in-one metrical and functional structure analysis with neighborhood attentions on demixed audio,” in *2023 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2023, pp. 1–5.
- [8] T.-P. Chen, L. Su, and K. Yoshii, “Learning multi-faceted self-similarity for musical structure analysis,” in *2023 Asia Pacific Signal and Information Processing Association Annual Summit and Conference (AP-SIPA ASC)*. IEEE, 2023, pp. 165–172.
- [9] O. Nieto and T. Jehan, “Convex non-negative matrix factorization for automatic music structure identification,” in *ICASSP*, 2013.
- [10] B. McFee and D. Ellis, “Analyzing song structure with spectral clustering,” in *ISMIR*, 2014.
- [11] T. Grill and J. Schlüter, “Music boundary detection using neural networks on combined features and two-level annotations.” in *ISMIR*, 2015.
- [12] G. Peeters, A. La Burthe, and X. Rodet, “Toward automatic music audio summary generation from signal analysis,” in *ISMIR*, 2002.
- [13] M. C. McCallum, “Unsupervised learning of deep features for music segmentation,” in *ICASSP*, 2019.
- [14] J.-C. Wang, J. B. Smith, J. Chen, X. Song, and Y. Wang, “Supervised chorus detection for popular music using convolutional neural network and multi-task learning,” in *ICASSP*, 2021.
- [15] M. Buisson, B. McFee, S. Essid, and H.-C. Crayencour, “A repetition-based triplet mining approach for music segmentation,” in *ISMIR*, 2023.
- [16] J. T. Foote and M. L. Cooper, “Media segmentation using self-similarity decomposition,” in *Storage and Retrieval for Media Databases 2003*, vol. 5021. SPIE, 2003, pp. 167–175.
- [17] M. Buisson, B. McFee, S. Essid, and H.-C. Crayencour, “Learning multi-level representations for hierarchical music structure analysis,” in *ISMIR*, 2022.
- [18] M. Won, K. Choi, and X. Serra, “Semi-supervised music tagging transformer,” in *ISMIR*, 2021.
- [19] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” in *International Conference on Learning Representations (ICLR)*, 2017.
- [20] Q. Li, Z. Han, and X.-M. Wu, “Deeper insights into graph convolutional networks for semi-supervised learning,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.
- [21] K. Han, Y. Wang, J. Guo, Y. Tang, and E. Wu, “Vision gnn: An image is worth graph of nodes,” *Advances in neural information processing systems*, vol. 35, pp. 8291–8303, 2022.
- [22] D. Ouyang, S. He, G. Zhang, M. Luo, H. Guo, J. Zhan, and Z. Huang, “Efficient multi-scale attention module with cross-spatial learning,” in *ICASSP*, 2023.
- [23] T. Monninger, J. Schmidt, J. Rupperecht, D. Raba, J. Jordan, D. Frank, S. Staab, and K. Dietmayer, “Scene: Reasoning about traffic scenes using heterogeneous graph neural networks,” *IEEE Robotics and Automation Letters*, vol. 8, no. 3, pp. 1531–1538, 2023.
- [24] K. Ullrich, J. Schlüter, and T. Grill, “Boundary detection in music structure analysis using convolutional neural networks.” in *ISMIR*, 2014.
- [25] S. Jadon, “A survey of loss functions for semantic segmentation,” in *2020 IEEE conference on computational intelligence in bioinformatics and computational biology (CIBCB)*. IEEE, 2020, pp. 1–7.
- [26] U. Von Luxburg, “A tutorial on spectral clustering,” *Statistics and computing*, vol. 17, pp. 395–416, 2007.
- [27] F. M. Bianchi, D. Grattarola, and C. Alippi, “Spectral clustering with graph neural networks for graph pooling,” in *International conference on machine learning*. PMLR, 2020, pp. 874–883.

- [28] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, “Rwc music database: Popular, classical and jazz music databases.” in *ISMIR*, 2002.
- [29] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, D. P. Ellis, and C. C. Raffel, “mir_eval: A transparent implementation of common mir metrics,” in *ISMIR*, 2014.
- [30] M. Levy and M. Sandler, “Structural segmentation of musical audio by constrained clustering,” *IEEE transactions on audio, speech, and language processing*, 2008.
- [31] Y.-Y. Yang, M. Hira, Z. Ni, A. Chourdia, A. Astafurov, C. Chen, C.-F. Yeh, C. Puhersch, D. Pollack, D. Genzel, D. Greenberg, E. Z. Yang, J. Lian, J. Mahadeokar, J. Hwang, J. Chen, P. Goldsborough, P. Roy, S. Narenthiran, S. Watanabe, S. Chintala, V. Quenneville-Bélaire, and Y. Shi, “Torchaudio: Building blocks for audio and speech processing,” *arXiv preprint arXiv:2110.15018*, 2021.
- [32] J. Salamon, O. Nieto, and N. J. Bryan, “Deep embeddings and section fusion improve music segmentation,” in *ISMIR*, 2021.
- [33] F. Korzeniowski, S. Böck, and G. Widmer, “Probabilistic extraction of beat positions from a beat activation function.” in *ISMIR*, 2014.
- [34] S. Böck, F. Korzeniowski, J. Schlüter, F. Krebs, and G. Widmer, “Madmom: A new python audio and music signal processing library,” in *Proceedings of the 24th ACM international conference on Multimedia*, 2016.
- [35] M. Y. Wang, “Deep graph library: Towards efficient and scalable deep learning on graphs,” in *ICLR workshop on representation learning on graphs and manifolds*, 2019.
- [36] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimeshein, L. Antiga *et al.*, “Pytorch: An imperative style, high-performance deep learning library,” *Advances in neural information processing systems*, 2019.
- [37] A. Marmoret, J. E. Cohen, and F. Bimbot, “Barwise music structure analysis with the correlation block-matching segmentation algorithm,” *Transactions of the International Society for Music Information Retrieval (TISMIR)*, vol. 6, no. 1, pp. 167–185, 2023.

Papers – Session II

SIX DRAGONS FLY AGAIN: REVIVING 15TH-CENTURY KOREAN COURT MUSIC WITH TRANSFORMERS AND NOVEL ENCODING

Danbinaerin Han^{1*}
Hannah Park⁴

Mark Gotham²
Sihun Lee³

Dongmin Kim³
Dasaem Jeong⁴

¹ Graduate School of Culture Technology, KAIST, Daejeon, South Korea

² Department of Digital Humanities, King’s College London, UK

³ Dept. of Artificial Intelligence, ⁴ Dept. of Art & Technology, Sogang University, Seoul, South Korea

naerin71@kaist.ac.kr, mark.gotham@kcl.ac.uk, {dmkim, hannah, sihunlee, dasaemj}@sogang.ac.kr

ABSTRACT

We introduce a project that revives a piece of 15th-century Korean court music, *Chihwapyeong* and *Chwipunghyeong*, composed upon the poem *Songs of the Dragon Flying to Heaven*. One of the earliest examples of *Jeongganbo*, a Korean musical notation system, the remaining version only consists of a rudimentary melody. Our research team, commissioned by the National Gugak (Korean Traditional Music) Center, aimed to transform this old melody into a performable arrangement for a six-part ensemble. Using *Jeongganbo* data acquired through bespoke optical music recognition, we trained a BERT-like masked language model and an encoder-decoder transformer model. We also propose an encoding scheme that strictly follows the structure of *Jeongganbo* and denotes note durations as positions. The resulting machine-transformed version of *Chihwapyeong* and *Chwipunghyeong* were evaluated by experts and performed by the Court Music Orchestra of National Gugak Center. Our work demonstrates that generative models can successfully be applied to traditional music with limited training data if combined with careful design.

1. INTRODUCTION

Six dragons fly on the east land; every endeavour is a heavenly blessing. This is the first line of lyrics in *Yongbieocheonga*, the first text written in the Korean alphabet (Hangul, 한글). Sejong the Great, one of the most respected figures in Korean history, invented and introduced Hangul in 1446. In addition to this remarkable achievement, he ordered scholar-officials to write *Yongbieocheonga*, and composed music to accompany the lyrics. Three other pieces composed at the time are *Yeo-Min-Lak*, *Chi-Hwa-Pyeong* and *Chwi-Pung-Hyeong*.

* Work mainly done during her master’s at Sogang University

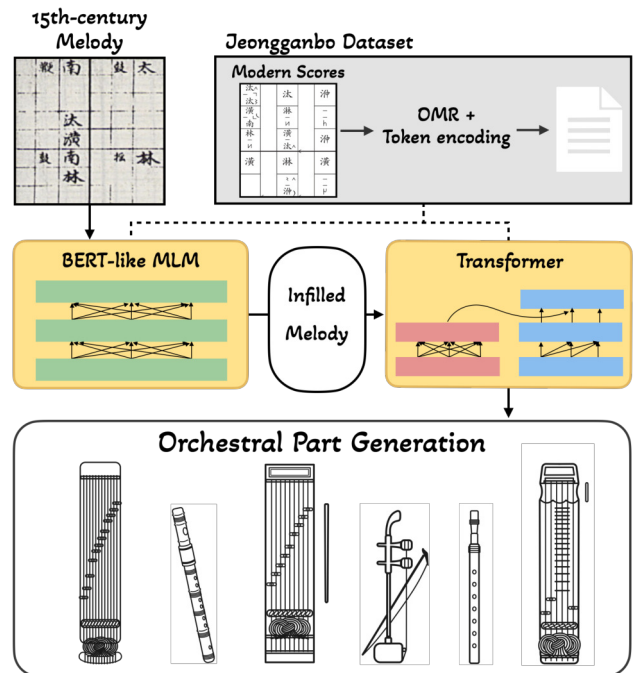


Figure 1: Overview of the proposed research framework

These compositions are still preserved in the *Veritable Records of Sejong*, which is the oldest surviving musical score in Korea [1]. More detailed information is available here [2].

Among these three pieces, only *Yeominlak* is handed down to the present day, while the other two are no longer performed. The National Gugak Center¹, which is the primary organization dedicated to the preservation and development of traditional music, commissioned the task of reconstructing these two pieces in a performable format using artificial intelligence systems. Given a simple melody of 512 *gaks* (measures) of *Chihwapyeong* or 132 *gaks* of *Chwipunghyeong*, the system must generate scores for six different instruments.

Our solution encompasses a wide range of tasks in the field of music information retrieval—constructing a specialized dataset, optical music recognition, designing a

¹ *Gugak* (국악) is the Korean term for traditional music

domain-specific encoding scheme, training models with limited data, and generating music of concert-level quality. In this paper, we present in detail the different frameworks used in the project: two types of transformer-based models; a symbolic dataset of Korean court music acquired through optical music recognition; and a novel “Jeonggan-like” encoding method that notates monophonic melody by combining notes’ position and pitch, along with a beat counter that informs the transformer the temporal position. The effectiveness of the proposed techniques was validated through quantitative metrics and subjective evaluation by experts from the National Gugak Center. Finally, we introduce a web demo that allows users to examine and generate traditional Korean court music interactively.

This project has significance not only for cultural preservation but also for wider considerations in machine learning and music generation. One of the many benefits to be had from the inter-cultural study of music is the different perspectives expressed in ‘the music itself’ as well as any notational and/or theoretical traditions that go alongside it. As presented in previous research [3], the encoding of music makes significant differences in machine learning tasks. In thinking through different ways of digitally encoding music, we stand to learn a great deal from the various syntaxes that have been used in diverse traditional contexts.

2. RELATED WORKS

Recent advances in neural network-based music generation have resulted in much artistic output. Since 2020, the *AI Music Generation Challenge* [4] has been held annually, focusing on generating songs in the style of Irish and Swedish folk music. This event has allowed for exploration of new methods for generation and evaluation of traditional music through the means of deep learning models.

The *Beethoven X project* [5] utilized neural networks to learn Beethoven’s compositional style and complete his unfinished 10th Symphony. The resulting work has been performed by an orchestra—a project outline similar to that of ours.

Attempts at automatic generation have been made for traditional music from beyond the West, including Persia [6] and China [7]. The limited progress in such areas is often due to the distinctive traditional musical systems that demand deep understanding and unique methodologies. Such idiosyncrasies put much interest and meaning in the computational research of traditional music, since it can present new methods and perspectives to the field as a whole, while also helping preserve diverse musical heritages.

3. JEONGGANBO DATASET

3.1 Jeongganbo Notation

As depicted in Figure 1, Korean court music is performed on a variety of instruments, including plucked string instruments (*Gayageum* and *Geomungo*), bowed string instruments (*Haegum* and *Ajaeng*), and wind instruments



Figure 2: An example of Jeongganbo in the original notion (below) and a broadly equivalent conversion to Western classical notion (above). Dashed lines are part of neither notation and added simply to clarify the temporal alignment between the two systems.

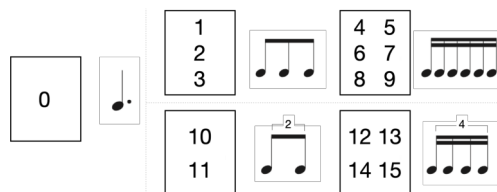


Figure 3: Jeonggan-like encoding position labels

(*Daegum* and *Piri*), among others. These instruments are played together in a heterophonic texture, with each instrument employing its distinctive playing techniques and ornamentations.

Much of Korean court music is written in *Jeongganbo*, a traditional musical notation system. *Jeongganbo* is recognized as the first system in East Asia capable of simultaneously representing both pitch and duration of notes [8,9]. This versatility has been instrumental in passing down court music throughout history [10].

Jeongganbo uses grid-divided boxes (*Jeonggans*) as the basic unit of time. The *number* of characters (notes) and their *position* within each *jeonggan* varies to denote rhythm. Figure 2 provides an example passage, and figure 3 provides a schematic overview of possible positions.

Here, we provide a broad introduction to this rhythmic notation system in quasi-Western musical theoretic language. Each *jeonggan* is broadly equivalent to a beat. If a *jeonggan* features only one character, this note event starts at the beginning of the beat and lasts the beat’s full duration. The first box (‘0’) in figure 3 is in this form as is the second *jeonggan* of figure 2 where the ‘compound beats’ correspond to the duration ♩. (in this case for the note B♭4). At the next metrical level we have the ‘column’ division of the ‘rows’. This number of ‘rows’ relates broadly to the top level division of the beat. The use of three vertically stacked characters refers to 3 equal divisions of this beat (here, 3 x ♩s). For example, in figure 3, the numbers 4–9 feature a 3-part division of the ♩ beat into 3 x ♩s (positions 4, 6, 8), and a 2x division of those ♩s (e.g., 4–5). If the following *jeonggan* is empty, the previously played note is sustained.

Playing techniques and ornamentations called *sigimsae* are sometimes notated for each instrument. When *sigimsae* are placed to the right of notes, they serve as ornamentations or embellishments for the corresponding note; when written on their own, they indicate timed instructions to play a specific note or musical phrase. For convenience,


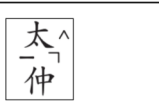
Jeonggan		
JG-like	:0 太	:1 太 ⌘ :7 □ :3 仲
REMI-like	0 太	0 太 ⌘ $\frac{1}{2}$ □ $\frac{2}{3}$ 仲
ABC-like	太 6	太 ^ 1.5 ▮ 0.5 仲 1

Figure 4: Comparison between encoding schemes

the example score is notated horizontally, but in practice, the score page is read from top to bottom and right to left. A line in *Jeongganbo* can consist of anything from four to twenty beats, with each line representing a phrase unit.

3.2 Machine Readable Dataset

We have constructed a dataset of 85 pieces by applying optical musical recognition (OMR) to all compositions available within the manuscripts published by the National Gugak Center. The manuscripts cover the entire repertoire of remaining Korean court music². OMR was necessary since the scores are only provided as PDF images and the semantic data is unavailable. We implemented and trained an encoder-decoder transformer with CNN by synthesizing various *Jeonggan* images in a rule-based approach [11]. In total, the dataset comprises 28 010 jeonggans across 85 pieces. When counting each instrument part independently, the combined total amounts to 141 820 jeonggans. Out of 90 pieces notated in jeongganbo for ensembles of at least two different instruments in the published manuscripts, we excluded 5 pieces that have discrepancies in the total number of jeonggans across instruments.

4. JEONGGAN-LIKE ENCODING

In the field of symbolic music generation for Western monophonic and polyphonic music, encoding schemes such as ABC notation, which denotes pitch and duration separately, are effective and prevalent [12, 13]. However, when it comes to Korean court music, whose heterophonic structure is a defining characteristic, it is crucial that the intricate alignment of different melodies be well-represented in encoding. The genre also exhibits prolonged notes and considerable variations in note lengths, which proves to be a challenge for learning algorithms, especially when data is limited.

These distinct musical qualities call for a specialized encoding scheme; for this, we propose *Jeonggan (JG)-like encoding*, which closely follows the positional notation of *Jeongganbo*. This symbolic music encoding method is modeled to inherently reflect the composition and notation style of traditional Korean court music.

The detailed rules of encoding are as follows. The boundary of a *Jeonggan* is designated as a bar (|) to-

² The term “court music” used in this paper originally refers to *Jeong-ak*. Jeong-ak includes not only court music but also salon music and military music. However, for readability, we use “court music” here.

ken. Change of measure (called *Gak*) is indicated by a line break ($\backslash n$). As illustrated in Figure 3, the position of each note is denoted by a number between 0 and 15, after which the pitch symbol follows.

Ornamentations (*sigimsae*) can either have a duration or not. *Sigimsae* with duration, such as the ‘ \neg ’ symbol in Figure 4, are handled in the same way as pitch symbols. *Sigimsae* without duration such as ‘ \wedge ’, which appear at the side of the pitch character, are placed after the corresponding pitch symbol.

There are several advantages that we can expect to gain from using *JG-like* encoding. First, with position-based encoding, the duration-related vocabulary is limited to just 16 entries. In contrast, duration-based encoding schemes require learning each duration token as a separate entry, resulting in a significantly larger vocabulary. Additionally, rather than determining the length of a note with a single calculation, JG-like encoding allows for the flexible adjustment of note lengths during inference via combination of *jeonggan* boundary and position tokens. This enables generation of music that is more adaptable to the time step and takes into account the sequence of the input source, which can be expected to result in more dynamic and context-aware music generation.

4.1 Other Possible Encodings

REMI (revamped MIDI-derived events) [14] first proposed the usage of beat-position feature rather than time-shifting to encode temporal position. We also experiment with REMI-like encoding which adopts three token types: beat position, new beat (instead of new measure), and pitch tokens. We intentionally design REMI-like and JG-like encoding to share the same structure and result in the same number of tokens for a given melody. They differ in that JG encoding provides intra-JG position, while REMI encoding provides the beat position of the note. According to the position labels shown in Figure 3, any of [0, 1, 4, 10, 12] can correspond to beat position 0. However, in JG-like encoding, each occurrence of position tokens limits the possibilities of subsequent ones. For instance, a position token of 0 implies that no more notes will occur in the same *jeonggan*, and if the first note is 1, one or more additional notes should follow with values of 2-3 or 6-9. In contrast, in REMI-like encoding, any offset value can follow a beat position of 0. To examine the impact of this position-based logic on the generation process, we use REMI-like encoding as our first baseline for comparison.

As a second baseline, we implement an ABC-like encoding scheme that does not have a separate bar token and encodes each note as a combination of pitch and duration values. Note that we do not omit duration tokens that are equal to unit length as ABC encoding typically does.

5. ORCHESTRAL PART GENERATION

5.1 Transformer Sequence-to-sequence Model

We implement an encoder-decoder transformer [15] model to generate melodies for different instruments based on a

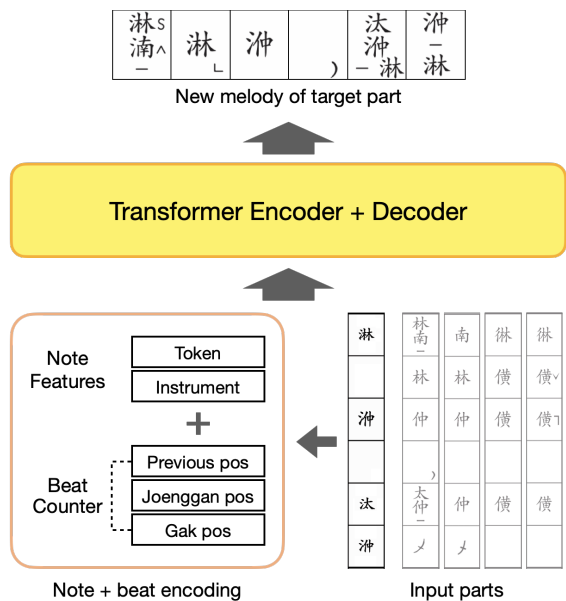


Figure 5: Orchestral part generation

given instrument’s melody, leveraging its ability to learn long-term dependencies. Unlike RNN-based models, the transformer calculates relationships between all elements in a sequence via the self-attention mechanism, enhancing its capability in symbolic music generation [16, 17, 18]. The model consists of an encoder that processes the input sequence and a decoder for generating the output sequence. Our objective is to generate melodies that synchronize with the input melody across musically equivalent phrases; self- and cross-attention within the model enable understanding of musical context at measure and bar levels, capturing the repeating structure of melodies and accents prominent in traditional Korean court music.

5.2 Beat Counter

Instead of sinusoidal [15] or learned [19] positional embedding commonly utilized in transformer-based models, we implemented a ‘beat counter’ embedding that provides information about temporal position.

For a model to learn to ‘parse’ semantic position only from the tokens’ sequential position is challenging, if not impossible, with limited training data and a small number of transformer layers. Therefore, we explicitly encode the musical position of each symbol as a combination of measure index, beat index, and sub-beat index (in-*jeonggan* position) as shown in Figure 5. This information is summed into note embedding, just like positional encoding of transformer [15].

As previous research of PopMAG [20] demonstrated, metrical position embeddings can replace the positional encoding of transformers in symbolic music. A minor difference between PopMAG and our approach is that the model predicts only the appearance of new measures or new beats without the index of them, and that the new beat can be used for elongating the duration of previous note.

The same idea of embedding the beat counting has been

previously applied to RNN-based Irish melody generation in ABC format [21], while its advantage was not properly evaluated. A similar idea, using metrical position instead of or along with absolute token position, has also been applied to transformer architecture [20, 22, 23]. However, our results presented in Section 6.3 demonstrate that this beat counter embedding is essential for making the model properly understand the musical contents.

6. EXPERIMENT AND RESULTS

6.1 Training

We split the *Jeongganbo* dataset into three subsets: 75 pieces for training, 5 for validation, and 5 for testing. Each piece contains melodies for up to 6 instruments. The sequence-to-sequence model takes 4 measures of melody, each from a randomly selected number of instruments, as input to the encoder; and given a target instrument condition, it generates the corresponding 4 measures of melody for the target instrument. Note that the number of beats in a single measure is at least 4 or to a maximum of 20 in our dataset.

The transformer encoder and decoder both consist of 6 layers, 4 attention heads, and a hidden dimension size of 128 with dropout of 0.2. We train for 35 000 updates across 300 epochs using negative log-likelihood loss. We also employ mixed precision training [24] to enhance performance and efficiency. We utilize the Adam optimizer with an initial learning rate of 0.001 and apply a cosine learning rate scheduler with 1000 warmup steps. Using a batch size of 16, training can be conducted on a single Nvidia RTX A6000 GPU.

6.2 Evaluation Metrics

6.2.1 Length Match Rate

As an evaluation metric, we check whether the input and output melodies share the same number of measures, a consistency necessitated by our task. Since the length of a measure can change in the middle of a piece, this metric serves as an indicator of the model’s ability to capture the musical context of the input melody and accordingly generate a musically complete melody. We measure *length match rate* as the percentage of generated melodies whose number of *jeonggans*, after decoding the output tokens, matches that of the input melody.

6.2.2 F₁-Score

Regarding the generation task as one with a fixed answer, we can measure the accuracy of the generated melody by directly comparing it with the ground-truth target melody. Thus, as a general accuracy metric, we calculate the F₁-score of predicted notes, where only the notes with the exact same onset position and pitch are counted as correct. To make a fair comparison between encoding methods, note onset positions in JG-like encoding were converted to those in the REMI-like format. Ornamentations without duration were not counted.

	<i>Piri to Geom.</i>		<i>Every to Daeg.</i>	
	len-mat	F1	len-mat	F1
JG-like	0.942	0.679	1.0	0.614
REMI-like	0.923	0.567	1.0	0.532
ABC-like	1.0	0.704	0.903	0.542
JG w/o Counter	0.135	0.043	0.269	0.052
REMI w/o Counter	0.269	0.081	0.192	0.039
ABC w/o Counter	0.403	0.090	0.115	0.016

Table 1: Quantitative evaluation results

6.3 Results and Discussion

In our sequential generation process, melody for the instrument geomungo, characterized by its low pitch range and simple melodies, is the first to be generated from the initial piri melody. The daegeum, typically featuring the most complex and nuanced melodies among the six instruments, is the last in line. Table 1 displays the results of objective evaluation, specifically focusing on geomungo and daegeum.

For generation of geomungo melodies, ABC-like encoding yields the best results. This appears to be due to the simple and regular melodic structure of the geomungo which fits in well with ABC-like encoding. On the other hand, in the task of generating daegeum melodies, JG-like encoding achieves higher F₁-scores. This indicates that JG-like encoding outperforms other methods in generating complex and varied melodies. We also discover that as rhythmic complexity increases, the measure length match rate of ABC-like encoding decreases.

To examine the effectiveness of the beat counter technique, we compare our model that incorporates beat counter with a baseline model that instead employs absolute position embedding [19], a technique commonly used in symbolic music generation.

The results in the lower part of Table 1 show that the models without beat counter fail to generate melodies with appropriate lengths. The problem is less severe in ABC-like encoding, as processing accumulating duration tokens can be easier than counting *jeonggan* boundaries. This demonstrates the efficacy of the beat counter technique in JG-like encoding, and its ability to replace traditional positional encoding.

7. 15TH CENTURY MELODY TRANSFORMATION

To generate an entire ensemble score using our method, we require an initial input melody with a specified instrument. However, the remaining 15th-century score of *Chihwapyeong* and *Chwipunghyeong* only provide a single melody without any mention of instruments. It also features rhythmic groupings of eight beats, which is rare in court music that is played today. We therefore need to transform the old melody for a specific instrument used in court music; to maintain the outline of the original melody while achieving plausible transformation, we train a masked language model on our *Jeongganbo* dataset be-

fore infilling the 15th-century melody.

7.1 BERT-like Masked Language Model

Bidirectional Encoder Representations from Transformers (BERT) [25] is a self-supervised language representation learning model that uses a bidirectional transformer instead of a causal transformer decoder. It is trained with a masked language model (MLM) objective, where tokens in the input sentence are randomly masked and the model predicts the original vocabulary ID of said masked tokens. Because of its advantage in exploiting bidirectional context, BERT-like models have also been adapted for music audio generation [26] and symbolic music generation [27, 28] along with representation-learning purpose adaptation on symbolic music [22, 29].

7.1.1 Piano-roll-like Encoding

One of the main limitations of using a BERT-like model for generative tasks is that the sequence of given (unmasked) tokens and masked tokens has to be pre-defined. This means that one has to decide the number and position of new tokens to be inserted for a given original sequence. To avoid this, we use piano-roll-like encoding for the MLM, a technique widely employed in works on music generation with limited rhythmic patterns such as in Bach Chorales [30, 31, 32, 33]. Here, each *jeonggan* is represented as six frames, with each frame including features for symbol (pitch or *sigimsae* with duration) and for ornamentation. We also apply the aforementioned beat counter in piano-roll encoding.

7.1.2 Training with Masking

Following examples in MusicBERT [29], we train the model with masked language model objective with various masking methods: i) masking 5% of frames, ii) replacing 5% of frames, iii) masking 20% of note onsets, iv) replacing 10% of note onsets, v) erasing 10% of note onsets, vi) masking the entire 6 frames of 15% of *jeonggans*, and vii) masking 50% of ornamentations.

Though the model can be trained to handle an arbitrary number of input instruments, we only train the model with a single instrument as with our orchestration transformer, since the main intended usage of the model is to create variations of a single melody. We train a 12-layer model with the same dataset and hyperparameter settings as with the orchestration model.

7.2 Inference Procedure

For converting and performing monophonic melodies, we opt for a 30x ♩ span which equals to 10 *jeonggans*. This also corresponds to the rhythmic pattern of the 4–7th movement of *Yeominlak*. The original *Chihwapyeong* and *Chwipunghyeong* melody, which can be interpreted in an 8/8 time signature, were modified by strategically inserting empty *jeonggans* to the 5th and 7th positions, to imitate *Yeominlak*'s rhythmic pattern. Utilizing the masked language model, the modified melodies were seamlessly

transformed into a piri melody. Piri, a double-reed instrument known for its loud volume, was chosen as the main instrument for conveying the original melody due to its prominent role in contemporary court music.

As the models were all trained on 4-measure chunks, we generate the full sequence of 512 or 132 measures using a moving window, providing two measures of previously generated output as teacher-forcing inputs and generating one more measure for each four-measure input. These were applied in a similar manner to both melody transformation and orchestral part generation. Once the melody is transformed into a piri melody, we feed it to the orchestral transformer to generate parts for five other instruments. We sequentially generate for each instrument with the previously generated part as input. The final generation order is as follows: piri, geomungo, gayageum, ajaeng, haegeum, and daegeum.

Following the initial generation of melodies for all six instruments, we perform a refinement step. Here, each instrument’s melody is regenerated with the melodies of the other five as input. This additional process helps to reinforce the melodies that initially had to be generated without the context of the other instruments.

7.3 Expert Reviews

The Court Music Orchestra of the National Gugak Center performed the generated *Chihwapyeong* and *Chwipunghyeong* on the birth anniversary of King Sejong at Gyeongbokgung Palace on May 14th, 2024. They performed it again at the National Gugak Center on June 2nd, 2024 with an introduction to technical background by the authors. Due to time constraints, only partial excerpts from the entire score were performed.

The musicians gave positive opinions such as “*genre-specific rhythm and melodic flow were well-represented*” and “*the generated pieces presented ornamentation techniques and melodic progressions specialized for each instrument.*” Still, there were a few instances where notes that did not fit the scale appeared, and when notes outside the appropriate range were present, the performers had to alter or omit them or change their octave to perform the piece. However, the generated results were acknowledged to closely resemble the target style of Yeominlak. Thus, the Court Music Orchestra decided to play the pieces in a similar ensemble size to Yeominlak without further modification.

We additionally evaluate the generated scores, focusing on the effects of the refinement step. The evaluation criteria were carefully selected to assess aspects that require a deep understanding of the genre. These criteria include 1) the appropriateness of the scale and range for each instrument (*scale*), 2) the proper use of unique characteristics and ornamentations specific to each instrument (*sigimsae*), 3) the suitability of the rhythmic structure of strong and weak beats (*rhythm*), and 4) the harmony and coherence among the instruments when performed together as an ensemble (*harmony*).

Seven employees from the National Gugak Center who

	<i>No Refinement</i>	<i>With Refinement</i>
Scale	4.0 (± 0.53)	4.0 (± 0.53)
Sigimsae	3.4 (± 0.73)	4.0 (± 0.53)
Rhythm	2.9 (± 0.35)	3.3 (± 0.45)
Harmony	2.9 (± 0.83)	3.3 (± 0.70)

Table 2: the average and std of opinion scores from 7 judges for systems with and without refinement.

majored in Korean traditional music instruments or theory participated in a subjective survey. We name the pre-refinement generation results piece A, and the final output after the refinement step piece B. The evaluators were not informed of this distinction. The participants assessed the pieces for the four criteria on a 5-point scale (1-5) and provided qualitative feedback on the two compositions. The results are summarized in Table 2. These results demonstrate that the proposed refinement process effectively enhances the overall quality of the generated music, especially for *sigimsae* of each instrument.

8. CONCLUSION

Throughout this work, we explored how music generation models can resurrect ancient melodies into new compositions that meet style of current-day Korean court music.

Venturing into relatively uncharted territory, we approached each step meticulously—from data curation and parsing to model architecture design—while carefully considering the unique nuances of the musical tradition. To enhance the quality of the generated outputs, we proposed a novel encoding framework and validated its effectiveness through objective and subjective measures. This endeavour to tackle an underrepresented non-Western music genre through diverse MIR lenses hopefully expands the horizons of the field.

The *Jeongganbo* dataset and its conversion to Western staff notation in MusicXML is available online, along with other code of this project, and video recording of the performance.³ To the best of our knowledge, this will be the first dataset of machine-readable *Jeongganbo*. We believe that this dataset can significantly contribute to computational ethnomusicology beyond its usage as a training dataset for music generation demonstrated in this paper.

We also provide an interactive web demo⁴ that showcases our proposed generative model. While this project focused on reviving melodies from the 15th-century, the web demo allows users to input their own melodies and create orchestrations of Korean court music. The interactive platform enables users to directly engage with the generative model in the web browser.

We hope that this project contributes to moving closer to leveraging machine learning to make traditional music more accessible and enjoyable for modern audiences.

³ <https://github.com/MALerLab/SejongMusic>

⁴ <https://six-dragons-fly-again.site/>

9. ACKNOWLEDGEMENTS

We sincerely appreciate the National Gugak Center and its staff who supported this project, including director-general Kim Youngwoon (김영운), head of the research bureau Kim Myung-suk (김명석), research officers Park Jeonggyeong (박정경) and Han Jungwon (한정원). We are deeply grateful to the musicians of the Court Music Orchestra for their invaluable contributions and efforts to vitalize our humble results. This research was also supported by the National R&D Program through the National Research Foundation of Korea (NRF) funded by the Korean Government (MSIT) (RS-2023-00252944, Korean Traditional Gagok Generation Using Deep Learning).

10. REFERENCES

- [1] Y. Kim, "Chapter III. critical assessment : The rhythmic interpretation of jeongganbo," in *Korean Musicology Series, vol. 4*, 2010.
- [2] National Gugak Center, "Publications," <https://www.gugak.go.kr/site/program/board/basicboard/view?menuid=001003002005&pagesize=10&boardtypeid=24&boardid=13154&lang=en>, 2024.
- [3] G. Micchi, M. Gotham, and M. Giraud, "Not all roads lead to Rome: Pitch representation and model architecture for automatic harmonic analysis," *Transactions of the Int. Society for Music Information Retrieval (TISMIR)*, vol. 3, no. 1, pp. 42–54, 2020.
- [4] B. Sturm, "The Ai music generation challenge 2022: Summary and results," in *Proc. of the 4th Conference on AI Music Creativity (AIMC 2023)*, Brighton, UK, 2023.
- [5] M. Gotham, K. Song, N. Böhlefeld, and A. Elgammal, "Beethoven X: Es könnte sein!(it could be!)," in *Proc. of the 3rd Conference on AI Music Creativity (AIMC 2022)*, Online, 2022, pp. 13–15.
- [6] M. Ebrahimi, B. Majidi, and M. Eshghi, "Procedural composition of traditional Persian music using deep neural networks," in *Proc. of 5th Conference on Knowledge Based Engineering and Innovation (KBEI)*. Tehran, Iran: IEEE, 2019, pp. 521–525.
- [7] J. Luo, X. Yang, S. Ji, and J. Li, "MG-VAE: Deep chinese folk songs generation with specific regional styles," in *Proc. of the 8th Conference on Sound and Music Technology (CSMT) Revised Selected Papers*. Shanxi, China: Springer, 2020, pp. 93–106.
- [8] A. E. Gnanadesikan, *The writing revolution: Cuneiform to the internet*. John Wiley & Sons, 2008, vol. 8.
- [9] Y. Kim, "Chapter II. Korean notational systems," in *Korean Musicology Series, vol. 4*, 2010.
- [10] R. Koehler *et al.*, *Traditional music: sounds in harmony with nature*. Seoul Selection, 2015.
- [11] D. Kim, D. Han, D. Jeong, and J. J. Valero-Mas, "On the automatic recognition of jeongganbo music notation: dataset and approach," *Preprint on Research Square*, 2024.
- [12] B. Sturm, J. F. Santos, and I. Korshunova, "Folk music style modelling by recurrent neural networks with long short term memory units," in *late-breaking demo session of 16th Int. Society for Music Information Retrieval Conf.*, Málaga, Spain, 2015.
- [13] B. L. Sturm, J. F. Santos, O. Ben-Tal, and I. Korshunova, "Music transcription modelling and composition using deep learning," *arXiv preprint arXiv:1604.08723*, 2016.
- [14] Y.-S. Huang and Y.-H. Yang, "Pop music transformer: Beat-based modeling and generation of expressive pop piano compositions," in *Proc. of the 28th ACM International conference on multimedia*, New York, NY, USA, 2020, pp. 1180–1188.
- [15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. of 31st Conference on Neural Information Processing Systems (NIPS 2017)*, Long Beach, CA, USA, 2017.
- [16] C.-Z. A. Huang, A. Vaswani, J. Uszkoreit, I. Simon, C. Hawthorne, N. Shazeer, A. M. Dai, M. D. Hoffman, M. Dinculescu, and D. Eck, "Music transformer," in *Proc. of The 7th International Conference on Learning Representations*, New Orleans, LA, USA, 2019.
- [17] B. Yu, P. Lu, R. Wang, W. Hu, X. Tan, W. Ye, S. Zhang, T. Qin, and T.-Y. Liu, "Museformer: Transformer with fine- and coarse-grained attention for music generation," in *Proc. of The 36th Annual Conference on Neural Information Processing Systems*, New Orleans, LA, USA, 2022, pp. 1376–1388.
- [18] Y.-J. Shih, S.-L. Wu, F. Zalkow, M. Müller, and Y.-H. Yang, "Theme transformer: Symbolic music generation with theme-conditioned transformer," *IEEE Transactions on Multimedia*, vol. 25, pp. 3495–3508, 2023.
- [19] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," in *Proc. of the 34th International Conference on Machine Learning*, Sydney, Australia, 2017, pp. 1243–1252.
- [20] Y. Ren, J. He, X. Tan, T. Qin, Z. Zhao, and T.-Y. Liu, "Popmag: Pop music accompaniment generation," in *Proc. of the 28th ACM Int. Conf. on Multimedia*, 2020, pp. 1198–1206.
- [21] D. Jeong, "Virtuosotune: Hierarchical melody language model," *IEIE Transactions on Smart Processing & Computing*, vol. 12, no. 4, pp. 329–333, 2023.

- [22] Z. Wang and G. Xia, “MuseBERT: Pre-training music representation for music understanding and controllable generation,” in *Proc. of the 22nd Int. Society for Music Information Retrieval Conf.*, Online, 2021, pp. 722–729.
- [23] Z. Guo, J. Kang, and D. Herremans, “A domain-knowledge-inspired music embedding space and a novel attention mechanism for symbolic music modeling,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 4, 2023, pp. 5070–5077.
- [24] P. Micikevicius, S. Narang, J. Alben, G. Diamos, E. Elsen, D. Garcia, B. Ginsburg, M. Houston, O. Kuchaiev, G. Venkatesh *et al.*, “Mixed precision training,” *arXiv preprint arXiv:1710.03740*, 2017.
- [25] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in *Proc. of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Minneapolis, Minnesota, USA, 2019, pp. 4171–4186.
- [26] H. Flores García, P. Seetharaman, R. Kumar, and B. Pardo, “Vampnet: Music generation via masked acoustic token modeling,” in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.*, Milan, Italy, 2023.
- [27] R. Dahale, V. Talwadker, P. Rao, and P. Verma, “Generating coherent drum accompaniment with fills and improvisations,” in *Proc. of the 23rd Int. Society for Music Information Retrieval Conf.*, Bengaluru, India, 2022.
- [28] L. Casini, N. Jonason, and B. L. T. Sturm, “Investigating the viability of masked language modeling for symbolic music generation in abc-notation,” in *Proc. of 13th International Conference on Computational Intelligence in Music, Sound, Art and Design*, Aberystwyth, UK, 2024, pp. 84–96.
- [29] M. Zeng, X. Tan, R. Wang, Z. Ju, T. Qin, and T.-Y. Liu, “MusicBERT: Symbolic music understanding with large-scale pre-training,” in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, Online, 2021, pp. 791–800.
- [30] F. Liang, M. Gotham, M. Johnson, and J. Shotton, “Automatic stylistic composition of bach chorales with deep LSTM,” in *Proc. of 18th Int. Society for Music Information Retrieval Conf.*, Suzhou, China, 2017, pp. 449–456.
- [31] G. Hadjeres, F. Pachet, and F. Nielsen, “DeepBach: a steerable model for Bach chorales generation,” in *Proc. of the 34th International Conference on Machine Learning*, Sydney, Australia, 2017, pp. 1362–1371.
- [32] C.-Z. A. Huang, T. Cooijmans, A. Roberts, A. Courville, and D. Eck, “Counterpoint by convolution,” in *Proc. of 18th Int. Society for Music Information Retrieval Conf.*, Suzhou, China, 2017.
- [33] E. Choi, H. Kim, J. Nam, and D. Jeong, “Teaching chorale generation model to avoid parallel motions,” in *Proc. of The 16th International Symposium on Computer Music Multidisciplinary Research (CMMR 2023)*, Tokyo, Japan, 2023.

LESSONS LEARNED FROM A PROJECT TO ENCODE MENSURAL MUSIC ON A LARGE SCALE WITH OPTICAL MUSIC RECOGNITION

David Rizo^{1,2} Jorge Calvo-Zaragoza¹ Patricia García-Iasci¹ Teresa Delgado-Sánchez³

¹ Pattern Recognition and Artificial Intelligence Group, University of Alicante, Spain

² Instituto Superior de Enseñanzas Artísticas de la Comunidad Valenciana, Spain

³ Biblioteca Nacional de España, Spain

{drizo, jorge.calvo, pgarcia.iasci}@ua.es mariateresa.delgado@bne.es

ABSTRACT

This paper discusses the transcription of a collection of musical works using Optical Music Recognition (OMR) technologies during the implementation of the Spanish PolifonIA project. The project employs a research-oriented OMR application that leverages modern Artificial Intelligence (AI) technology to encode musical works from images into structured formats. The paper outlines the transcription workflow in several phases: selection, preparation, action, and resolution, emphasizing the efficiency of using AI to reduce manual transcription efforts. The tool facilitated various tasks such as document analysis, management of parts, and automatic content recognition, although manual corrections were still indispensable for ensuring accuracy, especially for complex musical notations and layouts. Our study also highlights the iterative process of model training and corrections that gradually improved transcription speed and accuracy. Furthermore, the paper delves into challenges like managing non-musical elements and the limitations of current OMR technologies with early musical notations. Our findings suggest that while automated tools significantly accelerate the transcription process, they require continuous refinement and human oversight to handle diverse and complex musical documents effectively.

1. INTRODUCTION

In recent years, many institutions have digitized their collections to preserve them and make them available online for broader public access. Digital images, however, merely contain a grid of pixels and lack inherent musical meaning; thus, they do not lend themselves to the myriad possibilities offered by music information retrieval and digital musicology approaches, ranging from plain-text content searches to more sophisticated analytical purposes. To leverage these technologies, the music depicted in the

images must be encoded in a structured format, such as MEI [1] or MusicXML [2], among others.

Over the past few years, Optical Music Recognition (OMR) technologies have been employed to facilitate the encoding of music scores into structured digital formats [3]. Alfaro-Contreras et al. [4] demonstrated that the most effective method for obtaining digitally encoded scores is through the use of OMR technology. Their research indicates that the accuracy of OMR in recognizing musical notations varies depending on the type of document, the quality of the source material, and the complexity of the notation.

Despite its advances, OMR technology seldom produces flawless results, and the extent of necessary post-editing is determined by the intended use of the digitized content. For instance, some initiatives, such as F-Tempo¹, utilize OMR outputs—even when they contain errors—for conducting search operations. However, when a polished transcription is required, manual corrections become indispensable. This was the case considered in the digitization of a vast array of files for the KernScores database.²

The limitations of OMR technology are not solely determined by its recognition accuracy. To date, no OMR system is capable of comprehensively processing the entire spectrum of symbols found in all kinds of musical notations. The complexity of analyzing orchestral scores, with their varied layouts and the inclusion of *ossias*, or managing compositions where different parts are noted on separate sheets, further complicates the scenario. Consequently, in many practical applications, the encoding is ultimately carried out by human transcribers using computerized notation software like MuseScore³ or Sibelius.⁴ In specific projects such as Didone [5], about 4 000 18th-century Italian Opera arias are manually transcribed in Finale⁵ before being converted into MusicXML. This methodology was similarly employed to achieve the encoding of modern versions of Renaissance compositions from the “Josquin Research Project”.⁶

Furthermore, several OMR solutions exist for tran-



© D. Rizo and J. Calvo-Zaragoza. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** D. Rizo and J. Calvo-Zaragoza, “Lessons learned from a project to encode Mensural music on a large scale with Optical Music Recognition”, in *Proc. of the 25th Int. Society for Music Information Retrieval Conf.*, San Francisco, United States, 2024.

¹ f-tempo.org (accessed April 8th, 2024).

² kern.ccarh.org/ (accessed April 8th, 2024).

³ musescore.org (accessed April 8th, 2024).

⁴ www.avid.com/sibelius (accessed April 8th, 2024).

⁵ www.finalemusic.com (accessed April 8th, 2024).

⁶ josquin.stanford.edu (accessed April 8th, 2024).

scribing Common Western Modern Notation (CWMN), with Audiveris⁷ standing out as the sole open-source option alongside several proprietary alternatives, including SmartScore,⁸ PhotoScore,⁹ and PlayScore 2.¹⁰ The performance of these varies significantly based on the sheet music’s complexity and clarity. An evaluation of their efficiency in recognizing content from music theory books is detailed in the work of Moss et al. [6], highlighting the challenges they face in complex situations.

For early notations, the choices are much more limited. The SIMSSA project [7] considered two software tools—Gamut and Aruspix [8]—for automatic information extraction from images, although these tools are no longer actively supported. Additionally, the project developed an OMR meta-workflow named Rodan, enabling users to assemble custom processing systems from a library of image processing and machine learning modules [9]. While Rodan is not tailored to any particular musical notation, its components are predominantly focused on plainchant. Recently, a web-based OMR application named MuRET has been introduced as a research-oriented tool designed to facilitate the scientific study of the complete OMR workflow across various scenarios and notations [10]. This includes analyzing the real impact of improvements in automatic recognition models and their integration for practical purposes in the work of transcribers.

In this paper, we outline the entire process undertaken in the context of the Spanish PolifonIA project, for which MuRET has been utilized and refined to transcribe the entire collection of white Mensural notation held by the National Library of Spain (BNE) from scratch. We will detail all stages of the process, aiming to provide useful takeaways for other similar projects and transcription tools based on OMR. This includes discussing both manual and automated stages, the steps that may benefit from advancements in OMR techniques, those that still require human intervention, and which processes need to be streamlined due to their significant impact on workflow performance.

To illustrate the aforementioned aspects, figures will detail how, by the end of the project, more than 60 books containing around 12,000 images—some consisting of several pages—were encoded in just 18 person-months. Additionally, the figures will showcase how an iterative approach of transcription, correction, and AI-model training gradually accelerated the whole process.

The remainder of the paper is organized as follows. First, the data that has been transcribed is briefly introduced in Section 2. The following Section 3 describes the whole workflow used for obtaining a final digital score from a set of images in the source collection. This workflow will be analyzed from a quantitative point of view in Section 4, and then discussed from a qualitative perspective in Section 5. Finally, Section 6 concludes the work and discusses possible ideas for future research.

2. DATA

Although the workflow and evaluation described in subsequent sections are somewhat generic, this section provides details of the digitized collection to contextualize its significance.

The collection considered for the project totals 63 works, almost entirely in print editions dating from 1533 to 1811, and mostly written in white Mensural notation. The genres of these works are varied, comprising mainly vocal polyphonic pieces, although there is a presence of instrumental, dramatic, and even treatises. Their functions are predominantly religious, with some presence of profane songs. Their formal structure is linked to this, highlighting the complexity of formats in religious works ranging from Passion Cycle and Missae to the simpler forms of chansons or motets, among others. In polyphonic works, the parts are usually written in separate books.

Regarding printers, the collection features works from the Italian School such as: Scoto, Gardano or Vicenti from the Venetian; Dorico and Robbetti from the Roman; and Carlino and Beltrano from the Neapolitan. Le Roy and Ballard are prominent in the Paris School, along with the Flemish School’s Phalesius, Bellere, and Susato. Spanish publishers include Ibarra, Doblado, and Martinez Dávila in Madrid editions.

3. TRANSCRIPTION WORKFLOW

The transcription workflow can be broken down into several sequential phases.

The first stage involves the selection and compilation of works to be transcribed, either in PDF format or as a set of individual images. These are properly ordered through their file names following a lexicographic criterion for avoiding the need for time-consuming manual reordering within the tool.

For the sake of time and organizational management, the works are classified into different collections according to similarities in notation and/or publisher. This allows works sharing similar visual aspects to utilize the same machine learning models without adjustments between them. Considered features include notation type (plain chant, mensural, transitional scores, modern notation), engraving method (handwritten or typeset—where the copyist or printer is noted for sharing typography and layout styles), contents (treatises, instrumental and vocal music including lyrics), and the presence of elements such as basso continuo.

The next step involves uploading the works to MuRET. This tool employs OMR models that drastically reduce the image sizes to heights of 256 pixels. Although it utilizes IIIF servers that manage image resizing, for transcription purposes, it is not necessary to import high-quality images, but rather those with sufficient resolution to be readable on the user’s device. To avoid wasting server processing time and space, a prior down-sampling of images is advisable.

In the next stage, we refine the content imported into the tool. Most of the imported image sets contain covers, empty pages, and indexes that, while not containing strictly

⁷ github.com/Audiveris (accessed April 8th, 2024).

⁸ www.musitek.com (accessed April 8th, 2024).

⁹ www.neuratron.com/photoscore.htm (accessed April 8th, 2024).

¹⁰ www.playscore.co (accessed April 8th, 2024).

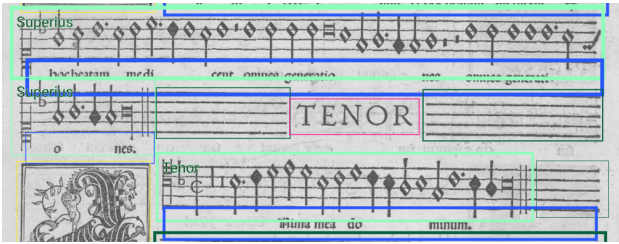


Figure 1. Example of document analysis and part assignment. Box colors represent different region types.

music information, are useful for extracting metadata and should be discarded but not removed, as they can guide the transcription process. Although some processes exist to automate this detection of pages with actual music content [11], MuRET does not include this desirable feature. Once the images are loaded and filtered, longer works to be transcribed must be divided into sections, such as the different parts of a mass (Kyrie, Agnus, etc.) or the movements of a concerto.

The final block aims to perform the actual transcription of the works. It consists of four main operations that will be detailed below: analyzing the document layout and dividing it into regions of interest, associating each staff with a part or instrument, recognizing the music contained in each staff and its encoding, and, finally, using all that information, scoring up all the parts to form a final digital score.

The document analysis and staff-level recognition of music symbols are performed using deep learning technologies [10]. Generally, we follow the same scheme for handling new works to be transcribed. First, models trained with previous collections are applied, mistakes are corrected, and then iteratively, new models are built, either specific for the collection if it is very different from previous documents, or following the proposal in [12], general for all transcribed collections. When faced with a new manuscript, the strategy is to first evaluate with the latest general model. If this does not perform well—which is evaluated subjectively by the user—we proceed to label, with or without the help of the OMR output, about twenty pages of the new work, then build specific OMR models and, in addition, enrich the general model for future works.

3.1 Document analysis

Upon arranging the images, the initial action in transcribing a manuscript, termed *document analysis*, involves dividing each image into distinct elements. This process detects various region types within the images, such as staves, lyrics, part names, among others, as illustrated in Figure 1. Typically, an image encompasses only a single page. However, scans of entire books are also common, resulting in images that depict multiple pages simultaneously, akin to the example shown in the figure.

3.2 Part management

The majority of materials requiring processing are polyphonic, composed of multiple voices or instruments. These

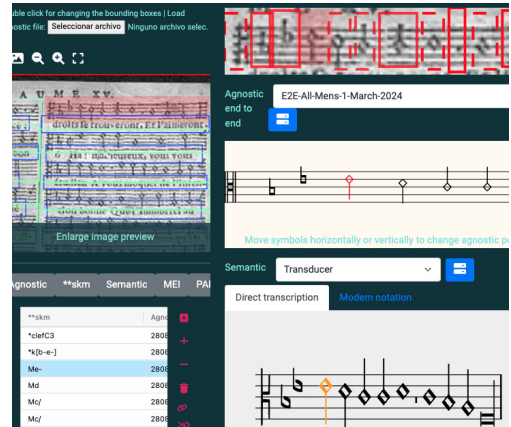


Figure 2. Agnostic representation and its semantic conversion in MuRET.

materials come in various formats, such as compositions with parts spread over several pages, or choir-books that display two voices on a single page (see Figure 1), among others. Occasionally, the document intended for transcription is dedicated to music theory, as seen in music treatises [6], predominantly featuring textual content with occasional musical illustrations. Currently, the assignment of parts is performed manually.

3.3 Region-wise content recognition

After distinguishing and assigning the various staves to their respective parts, it becomes essential to extract the musical elements located within each staff.

The approach applied divides the recognition of the musical content in a staff in two steps (see Figure 2). First, it extracts what is referred to as *agnostic representation* [13], i.e., tokens that have not yet been assigned a specific musical meaning, as well as their absolute vertical positions on the staff, regardless of the clef used. Then, these are automatically transduced into a meaningful ***mens* encoding [14], that can be manually post-edited.

After the automatic recognition, the eventual mistakes must be corrected. We found four different kinds of errors, with different impacts on the time required to be corrected. The easiest mistake is that of the vertical position of a recognized symbol (1), that is amended just with a mouse or keyboard action. A symbol whose type is wrongly detected (2) requires a slight higher effort, as it takes some seconds to find the expected symbol among all the possibilities. The removal of a symbol (3) is a very quick operation, while adding an undetected symbol (4) requires drawing a box over the manuscript image.

Note that for those difficult manuscripts for which all automatic models generate too many errors, as that shown in Figure 2, it might be preferable to manually add all agnostic symbols as described above.

3.4 Scoring up and exporting

As above mentioned, most of the works transcribed in the project are organized into separate parts or choral books, where different voices or instruments are scattered across

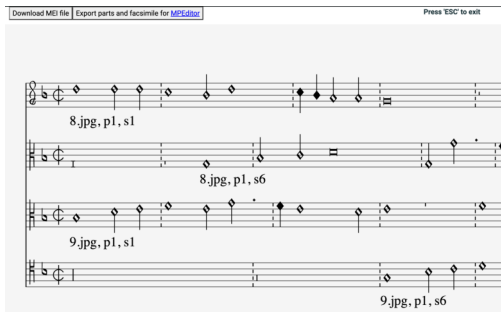


Figure 3. Example of alignment in MuRET. Some textual data is included below the staves to indicate reference points of the corresponding source image. Dashed bar lines are used to help detecting alignment errors.

different pages. Having already identified to which instrument each staff belongs (Sect. 3.2), this operation is simply accomplished by concatenating all the staves of the same part.

However, in Mensural notation, a preliminary step is required to correctly align the voices. In this notation, some notes may have different durations depending on the context, despite their appearance. The contextual resolution of durations, resulting in changes called *perfection* and *alteration*, can be carried out in MuRET either automatically, by applying the rules established in [15], or manually, by editing the `**mens` code.

In any case, mistakes such as missing symbols, incorrect duration elements, or invalid perfection assignments can only be detected by visually inspecting the aligned score (see Figure 3).

The final step of the process is exporting the transcription into an interchange or storage format. In the particular case of MuRET, the MEI standard is considered, offering two possible export formats: a parts-based MEI format that includes graphical information in the facsimile element or the arranged score MEI file.

4. QUANTITATIVE EVALUATION

MuRET records all operations performed by the user, saving the timestamp of each action and the element on which it is performed.

In this evaluation we address three questions. The suitability of using a transcription tool such as MuRET in a real-world scenario, the relative importance in OMR operations compared to the other tasks, and the ability of machine learning approaches to improve their accuracy as training datasets are iteratively expanded.

The first question is evaluated by comparing the performance of the tool with the theoretical hypothesis proposed in [4]. The first two rows of Table 1 show the times reported in [4] for processing 126 typeset pages of a *Magnificat*, either totally manually, or using an OMR.¹¹ Note that in that work, only the agnostic representation is obtained, and the time required for performing all the other tasks, such as the document analysis, or document preparation is

discarded. Automatic processing times are in all cases less than 1 second after loading the models into memory.

The next two rows show the process performed in the current project with the same *Magnificat*. First, the time to perform OMR processes (document analysis and recognition of agnostic symbols in each staff), then the entire transcription process, including all phases of the workflow. The final review of the scoring up has been excluded from these figures because in many cases the time is spent on musicological discussions of the manuscript rather than mechanical issues.

Finally, we have added to the table the worst case of those encountered in the project because it is a very difficult one due to the very low resolution of the images, which would have been extremely tedious to transcribe without the help of OMR (see Fig. 2), and the best case found for which the existing general OMR models have been able to correctly detect almost all symbols, and no part management was required.

The times reported demonstrate the suitability of using an OMR approach, but also the major impact on the whole process of the other, non-directly OMR processes, which cannot be overlooked.

Table 1. Summary of annotation times per page.

Scenario	Avg. Time/Page
<i>Magnificat work</i>	
Manual agnostic annotation [4]	49'19" ± 11'27"
OMR of agnostic representation [4]	15'23" ± 2'44"
OMR: doc. analysis and agnostic	22'07" ± 20'42"
Whole transcription process	29'09" ± 23'37"
<i>Whole project collection</i>	
Worst case (whole transcription)	52'31" ± 23'31"
Best case (whole transcription)	4'30" ± 1'51"

Regarding the second question, compare the relative importance of classic OMR operations with other operations such as document preparation or parts management for an entire collection, we show in Table 2 the times of all actions performed in MuRET grouped by all the workflow phases described in Sect. 3. The figures show that as it could be expected, the recognition of the musical symbols in each staff is the most time consuming task, followed by the semantic conversion and the document analysis, and what a priori could seem a slow operation, the manual assignment of parts to the staves, is a very small portion of the total, even lower than the preparation of images and organization into sections prior to the transcription itself.

Finally, to evaluate how incremental training of OMR models leads to better OMR behavior, we report in Fig.4 the number of operations performed on each image throughout the life of the project. Using the date axis is interesting because as the project has progressed, we have had more accurate OMR models because we have been trained on more data.

In the figure, we have used the number of operations instead of times because the time depends on the laptop on

¹¹ This value is computed from the values of Figure 2 in [4]

Table 2. Summary of time per phases.

Phase	Processing times
Document preparation	830'08''
Document analysis	4.913'36''
Part management	429'37''
Agnostic representation	1.9536'59''
Semantic encoding	10.923'55''

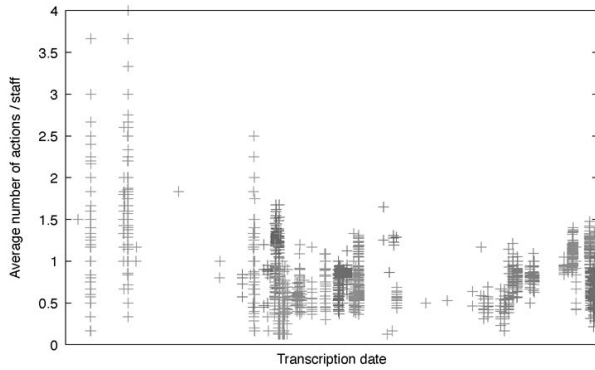


Figure 4. Evolution of transcription operations over time for all images in the project. Each point represents the number of operations required to transcribe a page.

which the operation was performed, since all classification models in MuRET are executed in the browser. Also, depending on the work, the number of staves of each image varies. To solve this, the graph shows the relationship between the number of operations and the number of staves of each image.

The figure shows how the average number of operations over time tends to be lower as the date progresses. We observe that an initial specialization of the OMR engine does help, and after that the user effort is stabilized.

5. QUALITATIVE EVALUATION OF THE WORKFLOW: OPPORTUNITIES

In this section we analyze the suitability of the involving stages of the transcription workflow to draw good practices for the development of OMR tools. We will discuss which operations we believe could be fully or partially automated to speed up the process and avoid tedious and repetitive work as much as possible.

The preparation of the works to be transcribed and its correct organization have been decisive for the success of the project. Since there is not yet a universal OMR model capable of dealing with any possible entry, the grouping of the pieces according to time period and typographic or calligraphic style, and the arrangement of the transcription following these groups, has been a key factor. In cases where, for some reason, we interleaved a piece out of that order, the performance decreased. While this clustering process was performed manually by computer scientists and musicologists, automating it could help to know in advance which existing OMR model could be applied to a new manuscript. Also, it is interesting to automatically de-

tect whether no model is able to process the manuscript and manual labeling of a number of pages is required to build a specific one.

A factor that we have already mentioned is that of the image resolution and, implicitly, the weight of the files. Although IIF servers are able to deal with the resizing of images, we have experienced a noticeable speed-up when the uploaded images are of smaller sizes.

Initially in the project, each work was processed following the different steps sequentially image by image. After processing some, instead, another approach was proved to be more convenient: perform all the operations of each phase for all the images of the work in batches. This allowed us to follow up on the work and detect possible errors made or not detected. It is important to note that in cases where we did all the tasks on each page and only reviewed them once, we made more errors. This approach was enhanced by a new feature added to MuRET in the middle of the project: the possibility of automatically tagging all work for later correction, which drastically improved transcription times by saving us OMR processing times for each page and staff (done “offline”).

A key aspect with a huge impact on the throughput of the workflow has been the (sometimes questionable) decisions of the MuRET developers in terms of UI/UX. The simplicity on the correction of the agnostic staff level automatic transcription and its automatic conversion into a meaningful semantic encoding in `**mens` format helped to minimize the impact of inaccurate OMR model predictions. A paradigmatic example has been the change in MuRET for the way of processing ligatures. The first OMR models in MuRET were not able to detect different mensural ligatures, but all different ligatures as a common symbol. The conversion of all ligatures to their final `**mens` encoding took longer than automatically encoding and correcting an entire page. During the transcription project, this tool was able to detect all the individual components of the ligature (plicas and note heads). Being quite accurate, when failing, the correction of the individual components took the same time as deleting the whole detection and adding them again. In a later version, this approach was changed by another one where the ligature was converted in a lower number of elements (different notes with or without plicas) with a bit worse OMR performance. However, for the purpose of final correction times, this change was appropriate because from then on, the correction time for errors was equivalent to the correction time for any other element.

Following this line, an aspect that could improve the efficiency of use of the system would be an easiest correction procedure of wrongly detected agnostic symbols. Currently, the user has to locate the symbol into a grouped list of possibilities. Even though this a specific criticism to MuRET, any simple mechanism in any transcription tool for locating the desired element to use, as some keyboard filtering approach, would significantly reduce the correction times.

The separation between agnostic representation and its final semantic encoding has proven to be an efficient way of processing early music. The ease of checking that the

graphic symbols are the same as those in the manuscript, regardless of musicological considerations, has greatly accelerated the process, allowing each member of the team to focus on one of the phases, leaving the expert musicologist to deal only with the final transcription. It's worth to mention, that in this process, a specialized language model to detect syntactic mistakes would have improved the efficiency of the process, as we have devoted most of the time to visually inspect the output of the OMR classifiers, even more than correcting wrong symbols.

Although the conversion of the agnostic representation of each staff into a final encoding is performed automatically, we've found cases where it has been necessary to make some adjustments, such as the encoding of implied accidentals. MuRET does not use any WYSIWYG approach but asks the user correct directly writing `**mens` format. Having a steep learning curve, the code has proven to be efficient for performing this kind of operations.

In that regard, another important feature, without which the correction operations would have been more tedious, has been the proper synchronization of the views of the different representations of the selected transcribed musical symbol: when selecting the agnostic symbol, it was automatically highlighted in the original manuscript preview, and in the final encoding. The absence of this feature in the final MuRET scoring-up process has made the final review and correction time consuming and error-prone.

For dealing with many different works with a large number of images each, it is very important to keep track of the status of the work. MuRET asks the user to record the status of each phase (document analysis, part linking, music transcription) for each image. Although a priori this seems reasonable, we usually forgot to perform this operation, and the simple task of going back individually to mark each image and step as completed has been a time consuming operation. For any transcription tool, it is extremely important to include a project management tool to easily annotate and visualize, either individually for each image or in batch, the progress status of the transcription, including the addition of user comments.

An interesting result of our transcription experience is that some operations do not require any algorithm, but are simply performed with a correct graphical user interface. This has been the case for document analysis labeling of new manuscripts for which no model was good enough to correctly identify the regions of interest. At the beginning of the project, when this situation arose, we had to manually label a number of pages of the manuscript to build a new model that was subsequently improved with new samples. For collections in which the layout of the regions of interest and the parts to which they belong is repeated over several pages, this process does not need any complex machine learning process, but a process of reusing the existing tagging is enough. During the project, MuRET included a tool to copy the document analysis and link parts to other images. This simple tool turned this tedious and repetitive operation of tagging the pages first into only a minor issue.

A notable case occurs in the event that the tool, or a component of a tool, does not support a required feature. For instance, bar-lines crossing a note in late Mensural no-

tations or the rendering of *signum congruentiae* is not supported in Mensural notation by the engraving tool used in MuRET, Verovio [16]. In those cases, our principle has been to store a specific element, such a text, and print them to be visualized, and once they are supported by the tools, replace them.

Finally, when focusing on the transcription of musical content, most tools discard many non-musical elements such as titles, part, instrument or voice names, capital letters miniatures. All this information, if automatically detected, could help to the users to have a better overview of large works to organize the transcription process.

6. CONCLUSIONS AND FUTURE WORK

Most of the OMR community's efforts are focused on achieving high accuracy rates in automated music reading. We have shown in Section 4 that the use of an OMR tool has proven to be an adequate means to transcribe a whole collection of works saving an enormous amount of time and effort for the user. While this approach is valid, it is important not to overlook aspects that are not intrinsically OMR and that can impact even more than the performance of the transcription tool on the effort required to transcribe collections.

In this work, we have shared our experience in transcribing a complete collection of works written in Mensural notation, describing all the steps taken and discussing issues we believe are important to achieve a streamline process, both from the perspective of the OMR tool used and in the preparation of the collections to be transcribed.

This paper has not addressed aspects that would be interesting to explore in the future. Some are related to the functioning of the computer system itself, such as the impact of classification times of automatic systems on the overall process and program response delays, as well as the measurement of the impact of execution errors or a comprehensive study from the perspective of human-computer interaction (HCI) in operations such as editing the staff transcription made or the final scoring up.

Other factors to consider are purely musical, such as the use of musical language models, both melodic and harmonic, for error detection, the impact of using one musical encoding over another, assistance in aligning lyrics with music, the treatment of abbreviations in the lyrics, or the detection of specific properties of the notation type such as the semitonia subintelecta in Mensural notation, the processing of multiple voices in piano-form music, the detection of hidden graphical elements such as the digit '3' in triplets in common western music notation, or finally the specific cases described by Byrd and Simonsen [17].

Regarding the OMR system, it is interesting to compare different strategies at work within a complete transcription system, not just in isolation. For instance, replace the MuRET stages (document analysis, agnostic representation, semantic encoding), for those based on graphical primitives and later semantic encoding reconstruction [18], or the direct obtaining of the final encoding from a complete page [19].

7. ACKNOWLEDGMENTS

This paper is part of the I+D+i TED2021-130776A-I00 (PolifonIA) project, funded by MCIN/AEI /10.13039/501100011033 and European Union NextGenerationEU/PRTR. The authors would like to thank Antonio Madueño, Adrián Roselló, Juan Carlos Martínez-Sevilla, and Antonio Ríos-Vila for their essential contribution to the project.

8. REFERENCES

- [1] P. Roland, “The music encoding initiative (MEI),” in *Proceedings of the First International Conference on Musical Applications Using XML*, jan 2002, pp. 55–59.
- [2] M. Good and G. Actor, “Using MusicXML for File Interchange,” *Web Delivering of Music, International Conference on*, vol. 0, p. 153, 2003.
- [3] J. Calvo-Zaragoza, J. Hajič, and A. Pacha, “Understanding optical music recognition,” *ACM Computing Surveys (CSUR)*, vol. 53, no. 4, pp. 1–35, 2020.
- [4] M. Alfaro-Contreras, D. Rizo, J. Iñesta, and J. Calvo-Zaragoza, “OMR-assisted transcription: a case study with early prints,” in *Proceedings of the 22nd International Society for Music Information Retrieval Conference*. Online: ISMIR, Nov. 2021, pp. 35–41.
- [5] A. Torrente and A. Llorens, “The Musicology Lab: Teamwork and the Musicological Toolbox,” in *Music Encoding Conference Proceedings 2021*. Humanities Commons, 2022, pp. 9–20.
- [6] F. Moss, N. Nápoles-López, M. Köster, and D. Rizo, “Challenging sources: a new dataset for omr of diverse 19th-century music theory examples,” in *Proceedings of the 4th International Workshop on Reading Music Systems (WoRMS 2022)*, November 2022.
- [7] I. Fujinaga, A. Hankinson, and J. Cumming, “Introduction to SIMSSA (single interface for music score searching and analysis),” in *Proceedings of the 1st International Workshop on Digital Libraries for Musicology*, ser. DLfM ’14. New York, NY, USA: Association for Computing Machinery, 2014, p. 1–3.
- [8] L. Pugin, J. Hockman, J. Burgoyne, and I. Fujinaga, “Gamera Versus Aruspix: Two optical music recognition approaches,” in *9th International Conference on Music Information Retrieval, Drexel University, Philadelphia, USA, September, 2008*, 2008, pp. 419–424.
- [9] I. Fujinaga and G. Vigiensoni, “The art of teaching computers: The SIMSSA optical music recognition workflow system,” in *27th European Signal Processing Conference, EUSIPCO 2019, A Coruña, Spain, September 2-6, 2019*. IEEE, 2019, pp. 1–5.
- [10] D. Rizo, J. Calvo-Zaragoza, J. Martínez-Sevilla, A. Roselló, and E. Fuentes-Martínez, “Design of a music recognition, encoding, and transcription online tool,” in *16th International Symposium on Computer Music Multidisciplinary Research, Tokyo*, November 2023.
- [11] A. Pacha and H. Eidenberger, “Towards self-learning optical music recognition,” in *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2017, pp. 795–800.
- [12] J. C. Martínez-Sevilla, A. Rosello, D. Rizo, and J. Calvo-Zaragoza, “On the performance of optical music recognition in the absence of specific training data,” in *Proceedings of the 24th International Society for Music Information Retrieval Conference, ISMIR 2023, Milan, Italy, November 5-9, 2023*, A. Sarti, F. Antonacci, M. Sandler, P. Bestagini, S. Dixon, B. Liang, G. Richard, and J. Pauwels, Eds., 2023, pp. 319–326.
- [13] J. Calvo-Zaragoza and D. Rizo, “End-to-end neural optical music recognition of monophonic scores,” *Applied Sciences*, vol. 8, no. 4, 2018.
- [14] D. Rizo, N. Pascual-León, and C. Sapp, “White Mensural Manual Encoding: from Humdrum to MEI,” *Cuadernos de Investigación Musical*, 2019.
- [15] M. E. Thomae, J. E. Cumming, and I. Fujinaga, “The mensural scoring-up tool,” in *Proceedings of the 6th International Conference on Digital Libraries for Musicology*, ser. DLfM ’19. New York, NY, USA: Association for Computing Machinery, 2019, p. 9–19.
- [16] L. Pugin, R. Zitellini, and P. Roland, “Verovio - A library for Engraving MEI Music Notation into SVG,” in *International Society for Music Information Retrieval*, jan 2014.
- [17] D. Byrd and J. G. Simonsen, “Towards a standard testbed for optical music recognition: Definitions, metrics, and page images,” *Journal of New Music Research*, vol. 44, pp. 169–195, 1 2015.
- [18] J. Hajič and P. Pecina, “The muscima++ dataset for handwritten optical music recognition,” in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 1. IEEE, 2017, pp. 39–46.
- [19] A. Ríos-Vila, J. M. Iñesta, and J. Calvo-Zaragoza, “End-to-end full-page optical music recognition for mensural notation,” in *ISMIR*, 2022, pp. 226–232.

THE CHANGING SOUND OF MUSIC: AN EXPLORATORY CORPUS STUDY OF VOCAL TRENDS OVER TIME

Elena Georgieva¹, Pablo Ripollés^{1,2,3}, Brian McFee^{1,2,4}

¹ Music and Audio Research Laboratory, New York University,

² Center for Language, Music, and Emotion, New York University,

³ Department of Psychology, New York University,

⁴ Center for Data Science, New York University

{elena, pripolles, brian.mcftee}@nyu.edu

ABSTRACT

Recent advancements in audio processing provide a new opportunity to study musical trends using quantitative methods. While past work has investigated trends in music over time, there has been no large-scale study on the evolution of vocal lines. In this work, we conduct an exploratory study of 145,912 vocal tracks of popular songs spanning 55 years, from 1955 to 2010. We use source separation to extract the vocal stem and fundamental frequency (f_0) estimation to analyze pitch tracks. Additionally, we extract pitch characteristics including mean pitch, total variation, and pitch class entropy of each song. We conduct statistical analysis of vocal pitch across years and genres, and report significant trends in our metrics over time, as well as significant differences in trends between genres. Our study demonstrates the utility of this method for studying vocals, contributes to the understanding of vocal trends, and showcases the potential of quantitative approaches in musicology.

1. INTRODUCTION

Current technologies for audio processing provide new opportunities to study musical trends using quantitative methods. While researchers have analyzed music for generations, studying the evolution of music at a large scale has only been possible recently, due to the availability of large datasets [1–3]. Additionally, recent improvements in source separation technology have allowed researchers to study individual instruments [4, 5]. However, the vocal lines of songs have been understudied, even though they are often the most salient part of a song [6, 7], and many popular songs are built around the vocal line.

In this study, we examine trends in the vocal lines of 145,912 songs over 55 years (from 1955 to 2010). We use modern source separation methods to isolate vocal lines of

songs (30–60 second-long excerpts) from their respective accompaniments. Altogether, our dataset makes up over 59 days of continuous listening. This work is exploratory: we examine what trends and patterns can be observed from such a large corpus of vocal data. We have made our list of track IDs publicly available, along with our implementations.¹

2. RELATED WORK

The transformation of music over time has received a lot of focus in recent years. This is partially thanks to the release of open-source resources such as The Million Song Dataset (MSD) [1]. The MSD is a free collection of audio features and metadata for one million contemporary music tracks. Datasets such as MSD allow researchers to quantitatively analyze patterns in music at a large scale.

Serrà *et al.* used musical ‘codewords’ based on MSD clips to identify changes in pitch, timbre, and loudness over time [2]. They found that newer songs have less variety in pitch transitions, more homogenized timbres, and increased loudness. Parmer *et al.* did similar work using the MSD to study musical complexity from 1960–2010. They found that pitch complexity has been generally stable over that time period, while loudness and rhythm complexity has decreased and timbral complexity has increased [8]. Parmer also studied the complexity of popular songs from the Billboard chart,² and found that the complexity of popular songs is concentrated around the mean complexity level of all songs. This supports the inverted U-shaped model for music complexity and likeability: that listeners prefer intermediate levels of complexity [9, 10].

Another team of researchers used the MSD songs along with quantitative modeling to study musical influence: the impact that a particular artist has on the music by other musicians [3]. They identified clusters of songs that were indicative of a genre, and studied how those clusters evolved over time. A different study used a corpus of 17,000 songs from Billboard to study the “Evolution of Popular Music” between 1960 and 2010 in the United States [11]. They used timbral and harmonic features derived from Billboard songs, and identified three musical stylistic revolutions in



© E. Georgieva, P. Ripollés, B. McFee. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).
Attribution: E. Georgieva, P. Ripollés, B. McFee, “The Changing Sound of Music: An Exploratory Corpus Study of Vocal Trends Over Time”, in *Proc. of the 25th Int. Society for Music Information Retrieval Conf.*, San Francisco, CA, USA, 2024.

¹ <https://github.com/elena-theodora/ismir2024-changing-sound-of-music>

² <https://www.billboard.com/charts/hot-100>

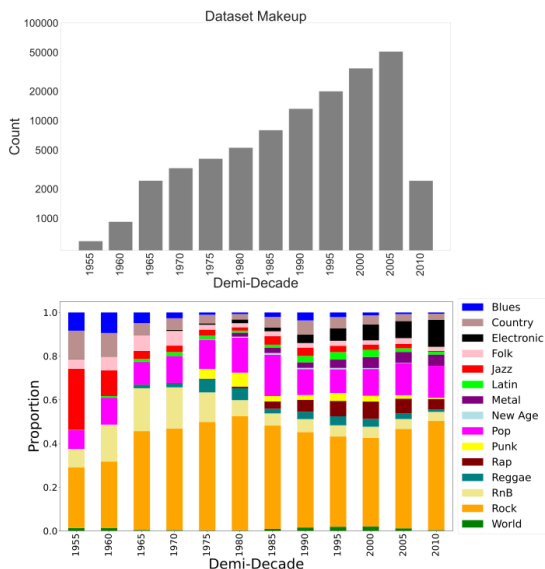


Figure 1: Top: Chronological distribution of the dataset organized in 5-year demi-decades. Bottom: Relative distribution of genres in the dataset by demi-decade.

Musical Genre & Number of Tracks					
Rock	64,203	Country	6,166	Punk	3,444
Pop	19,560	Metal	5,841	Latin	3,352
Rap	8,755	Reggae	4,689	Blues*	2,359
Electronic	8,754	Jazz	3,953	World*	2,044
RnB	8,647	Folk	3,630	New Age*	515

Figure 2: Number of tracks per musical genre. Blues, World, and New Age music (labeled “*”), were excluded from the by-genre analyses due to lower track count.

1964, 1983, and 1991. Other researchers have studied a more niche topic in detail over decades, including the evolution of a single band’s performances [12], changes in dynamics/compression in mainstream music [13], or spectral characteristics of recordings over time [14].

In an early vocal corpus study, in 1959, Alan Lomax’s Cantometrics project analyzed over 4,000 traditional vocal music songs from 400 cultures [15]. Researchers listened to songs and labeled them with 37 “style-factors,” for example group cohesion in singing, and tense or relaxed vocal quality. The Cantometrics project suggested a correlation between song style and social norms of cultures.

In a more recent study, researchers developed a set of features to capture pitch and melodic embellishments of world vocal performances [16]. Using these features, they trained a classifier to distinguish vocal from non-vocal segments and learn a dictionary of singing style elements. Results showed that clusters were distinguished by characteristic uses of singing techniques such as vibrato and melisma. A different study categorized a collection of 360 Dutch folk songs, and found that the aspects of melody that are important for establishing similarity are contour, rhythm, and motifs [17]. Despite these previous works on vocal datasets, there has been no large-scale study on the evolution of the vocal lines of popular music over the years.

3. DATASET

We used a subset of the MSD [1] that has genre labels (the Tagtraum MSD annotations [18]). 278,619 tracks had genre labels available. Next, a group of songs was dropped due to a low presence of vocals in the excerpt, indicated by a low ratio of RMS (root mean square) energy of the separated vocal stem to RMS energy of the full audio file (see 4.1). Songs that did not have the release year available were also dropped. In a final filtering step, we chose to conduct analyses only starting in the year 1955, as data was sparse before 1955. The final dataset had 145,912 songs.

Figure 1 shows a chronological distribution of songs in demi-decade bins (i.e., 1990-1994). We observe a strong bias towards more recent songs. A relative distribution of genres across years shows fewer genres in earlier years, with a greater variety in more recent years. Figure 2 lists the number of tracks in each musical genre in the dataset. Blues, World, and New Age music (labeled with a '*'), were excluded from the by-genre analyses due to having a lower number of tracks.

Our dataset inherits biases from the MSD. The tracks in the MSD were selected based partly on their association with ‘familiar’ artists, as determined by The Echo Nest, followed by inclusion of tracks from similar artists.³ The creators of the MSD also included artists that fit the 200 most frequently-occurring Echo Nest descriptive terms, as well as songs that were extreme in acoustic attributes. In general, songs in the dataset are generally widely listened-to, and the majority come from North America or Europe. There are much more data in recent years (1990s onward) than in earlier years. There is very little non-western and classical music in the dataset. The Latin music genre does contain non-western music, primarily performed in Spanish or Portuguese. Our findings apply to this dataset, not necessarily to music as a whole, and our work will have biases if applied to other datasets. Importantly, these dataset biases do not affect our methods.

4. METHOD

4.1 Source Separation

First, we used source separation to separate the vocal line of each song from the mix. For this, we use Hybrid Transformer Demucs (HT Demucs), a hybrid temporal/spectral bi-U-Net [5]. After computing the ratio of the vocal stem’s RMS energy to the overall mix’s RMS energy, we excluded any songs with a ratio below 0.08 (Figure 4). This ratio was set using a preliminary sub-sample of the data. These excerpts are either purely instrumental songs (non-vocal), or the clip happens to capture a part of the audio file with very few or no vocals (i.e., a guitar solo).

4.2 Pitch Characteristics

To study pitch characteristics, we did fundamental frequency (f_0) estimation on the estimated vocal stems using PYIN [19] as implemented in Librosa v0.8.1 [20].

³ <http://millionsongdataset.com/faq/>

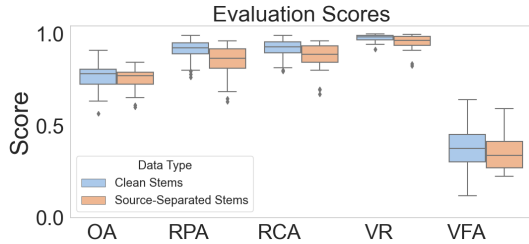


Figure 3: f_0 extraction evaluation scores for 29 clean and source-separated vocal stems from MedleyDB, when compared with the MedleyDB f_0 annotations.

We set the lower frequency limit at 70Hz and the upper limit at 900Hz, aligning with the human vocal range described in other works, while also extending one musical whole step in each extreme [21]. We chose PYIN over CREPE, another f_0 -estimator, as it allows us to set a lower and higher pitch bound for f_0 -estimation [22]. Other than changing the sampling rate to 44.1 kHz, we used the Librosa defaults: frame length 2048, hop length 512, number of thresholds for peak estimation 100, switch probability 0.01, and no-trough probability 0.01. We collect an f_0 estimate approximately every 12 milliseconds.

We evaluated the pitch tracking accuracy of the PYIN algorithm on source-separated audio by running PYIN on 29 monophonic vocal stems from MedleyDB. We used `mir_eval` to compute the standard evaluation metrics used in MIREX: Voicing Recall (VR), Voicing False Alarm (VFA), Raw Pitch Accuracy (RPA), Raw Chroma Accuracy (RCA) and Overall Accuracy (OA) [23]. First, we compared several different PYIN settings: the number of thresholds, switch probability, and no-trough probability parameters, and found the Librosa defaults performed among the best. Next, we compared the accuracy of PYIN on clean stems and source-separated stems, each respectively compared to the annotations included in MedleyDB, and observed only a small decrease in accuracy. The median evaluation metrics of our method on the 29 clean vocal stems were: for clean stems, OA 0.781, RPA 0.924, RCA 0.928, VR 0.984, VFA 0.375, and for source separated stems, OA 0.771, RPA 0.865, RCA 0.888, VR 0.964, VFA 0.340 (see Figure 3). The source separation process only slightly reduces the accuracy of our f_0 -estimation.

Some tracks in the dataset have vocal harmonies. PYIN tends to track the pitch of the most prominent voice. We ran a query on last.fm,⁴ and found that tags for vocal harmonies are present in less than 1% of songs in the dataset. We assume that the presence of vocal harmonies is uncorrelated with the variables we study: time and genre. Through informal listening, we found that Demucs and PYIN were comparably effective for older and newer audio recordings from the time period we study, 1955-2010.

PYIN also provides a voicing detection estimate, which we used to identify contiguous regions of pitched sound in the vocal stem. We converted f_0 values in hertz to cents

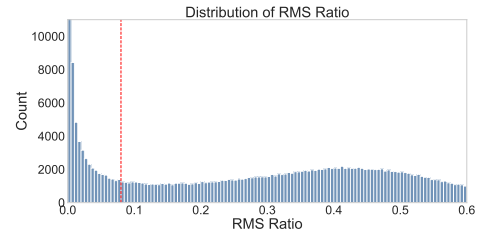


Figure 4: Distribution of the ratios of the vocal stem RMS energy to the full mix RMS energy in the data. A threshold of 0.08 was used to discard non-vocal clips.

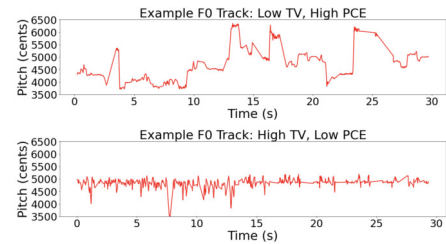


Figure 5: Example f_0 tracks. The top track, selected from the Metal genre, has a low TV and high PCE. The bottom track, from the Rap genre, has a high TV and low PCE.

using Eqn (1), where 16.35Hz is the frequency of C0:

$$f_0[c] = 1200 \cdot \log_2 \frac{f_0[\text{Hz}]}{16.35}. \quad (1)$$

For example, "middle C" on a piano is 4800 cents, and $C\#/Db$ is 4900 cents. Using this information we extracted pitch features, dropping unvoiced frames. We calculated mean pitch (in cents) of each song, defined as the mean of each f_0 array.

We also calculated total variation (TV) [16]. TV summarizes the rate of pitch change and is defined in Eqn (2):

$$\text{TV}(x) = \frac{1}{N} \sum_{i=1}^{N-1} |x_{i+1} - x_i| \quad (2)$$

for a given f_0 contour $x = (x_1, \dots, x_N)$. TV is calculated independently for each voiced region within a song and then aggregated to a single total. Our TV calculations do not change the time interval between f_0 values.

4.3 Pitch Class Entropy

We calculated pitch class entropy (PCE) to measure the degree of unpredictability for the set of vocal pitches. Entropy was calculated over the probability of occurrence of each pitch class (independent of octave) in the vocal line [24]. Higher values of PCE indicate a greater spread in the pitch distribution, while lower values indicate a smaller and more predictable set of pitches. There is a theoretical maximum PCE of $\log_2(12) \approx 3.59$, achieved by a uniform distribution of the 12 pitch classes.

Figure 5 illustrates two example f_0 tracks with somewhat extreme TV and PCE values.

4.4 Statistical Analyses

We used R (4.2.2) and RStudio (2022.12.0+353) to implement linear regression with the `lm` function. Post-hoc tests

⁴ <https://www.last.fm/>

were implemented using the *emmeans* package with Tukey correction for multiple comparisons.

5. EXPERIMENT AND RESULTS

For each of our variables of interest (mean pitch, TV, PCE), we followed the same procedure. We first ran a linear regression to examine the relationship between the variable of interest (e.g., TV) and the year of track release (e.g., TV \times year). We then calculated a linear regression between the variable of interest and musical genre (e.g., TV \times genre). Finally, we calculated independent linear regressions between the variable of interest and year of track release for the twelve most frequently-occurring genres. We calculated independent regressions because each of the genres becomes prevalent in the dataset during different years (i.e., Rap music starting in 1984).

When looking at musical genres, we chose to study the twelve genres with the most song entries in the dataset. We began analyzing each genre at the first year of a five-year period where at least ten songs were released in that genre annually. The twelve genres with corresponding start years were as follows: Country (1956), Electronic (1979), Folk (1963), Jazz (1955), Latin (1986), Metal (1980), Pop (1961), Punk (1977) Rap (1984), Reggae (1972), RnB (1957), and Rock (1956).

5.1 Mean Pitch

We found a significant positive relationship between mean pitch for a track and the year it was released ($\beta = 0.957$, $t = 7.23$, $p < .001$). For every one-year increase in the release year, the mean pitch of the track increased by a little less than one cent, on average (see Figure 6).

Next, we assessed mean pitch and musical genres. We found a significant main effect of genre ($F(1, 140982) = 1378.6$, $p < 0.001$). All genres were significantly distinct (all p values < 0.001) except for: electronic and pop music ($t=1.891$, $p=0.765$), jazz and Latin ($t = 0.276$, $p=1.000$), jazz and rock ($t=-2.854$, $p = 0.158$), and Latin and rock ($t = -3.005$, $p = 0.107$). Data for the mean pitch per song in each of these genres is illustrated in Figure 7.

We found a significant main effect of year for nine of the twelve musical genres, though the direction of the trends varied (see Figure 8). Specifically, country music ($\beta = 3.991$, $t = 7.970$, $p < 0.001$), folk music ($\beta = 1.374$, $t = 2.395$, $p < 0.001$), jazz ($\beta = 2.901$, $t = 4.482$, $p = 0.017$), metal ($\beta = 7.269$, $t = 4.612$, $p < 0.001$), punk ($\beta = 3.301$, $t = 3.815$, $p < 0.001$), reggae ($\beta = 2.053$, $t = 3.301$, $p < 0.001$) and rock ($\beta = 1.097$, $t = 5.529$, $p < 0.001$) showed a significant positive relationship between year and mean pitch. Conversely, rap ($\beta=-6.653$, $t=-7.757$, $p<0.001$) and RnB ($\beta=-3.800$, $t=-11.75$, $p<0.001$) showed a significant negative relationship between year and mean pitch. No significant effect was found for electronic, Latin, or pop music.

5.2 Total Variation

The results for the TV and year regression between TV and year showed a significant negative relationship ($\beta = -$

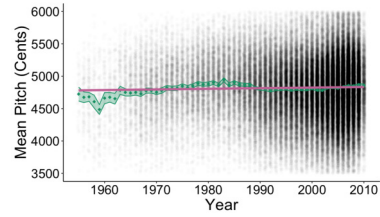


Figure 6: Mean pitch in cents as a function of year globally. Each dot represents a song. The red line represents the predicted slope with 95% confidence intervals. The green diamond and ribbon represent the mean per year and the standard error. This relationship was significant, with mean pitch increasing by approximately one cent per year.

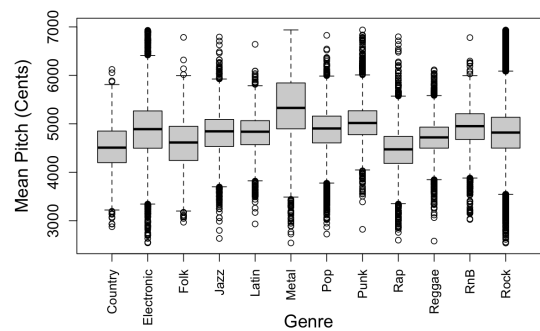


Figure 7: Mean pitch per song in each of the twelve genres across the dataset. Means are shown with boxes representing the interquartile range, error bars indicating the 95% confidence interval, and outliers as circles. There were significant differences between all genres except electronic and pop, jazz and Latin, jazz and rock, and Latin and rock

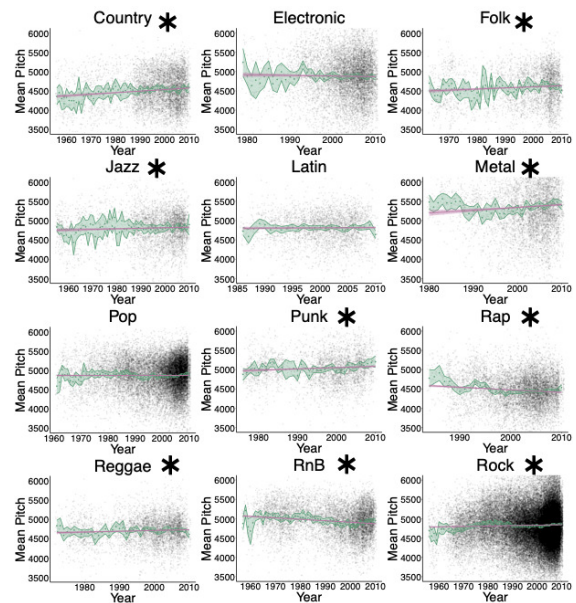


Figure 8: Relationship between mean pitch (in cents) and year for each genre. “*” denotes a significant effect of year. The red line represents the predicted slope with 95% confidence intervals. The green diamond and ribbon represent the mean per year and the standard error.

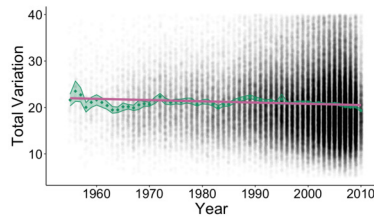


Figure 9: Total Variation as a function of year. Each dot represents a song. The red line represents the predicted slope with 95% confidence intervals. The green diamond and ribbon represent the mean TV per year and the standard error. There was a significant negative correlation between TV and year.

0.027, $t = -10.47$, $p < .001$; see Figure 9). When assessing the relationship between TV and musical genre, we found a significant main effect of genre ($F(11, 140,980) = 1247.9$, $p < 0.001$). Post-hoc tests showed that all genres were significantly different from one another (all p values < 0.05) except for country and Latin ($t=0.661$, $p = 1.000$), country and punk ($t = -3.214$, $p = 0.059$), electronic and punk ($t = 0.877$, $p = 0.999$), electronic and reggae ($t=0.269$, $p=1.000$), folk and pop ($t = -2.407$, $p = 0.4013$), folk and rock ($t = 0.185$, $p = 1.000$), Latin and pop ($t = 3.006$, $p = -.107$), and punk and reggae ($t = -0.569$, $p = 1.000$; see Figure 10). Importantly, TV was significantly higher for rap music than for all other genres.

We found a significant main effect of year on TV for eleven of the twelve musical genres, though the direction of the trends varied (see Figure 11). Specifically, metal music ($\beta = 0.066$, $t = 3.338$, $p < 0.001$), reggae music ($\beta = 0.058$, $t = 6.512$, $p < 0.001$), and RnB ($\beta = 0.013$, $t = 3.44$, $p < 0.001$) showed a significant positive relationship between year and TV. Conversely, electronic music ($\beta=-0.126$, $t=-4.803$, $p<0.001$), folk ($\beta=-0.065$, $t=-7.852$, $p<0.001$), jazz ($\beta=-0.079$, $t=-6.418$, $p<0.001$), Latin music ($\beta=-0.041$, $t=-2.349$, $p=0.019$), pop ($\beta=-0.033$, $t=-8.698$, $p<0.001$), punk ($\beta=-0.045$, $t=-3.259$, $p=0.001$), rap ($\beta=-0.158$, $t=-11.06$, $p<0.001$) and rock ($\beta=-0.062$, $t=-13.79$, $p<0.001$) showed a significant negative relationship between year and TV. No significant effect was found for country music.

5.3 Pitch Class Entropy

A linear regression showed a statistically significant negative relationship between PCE and year ($\beta = -0.004$, $t = -50.02$, p -value < 0.001 ; see Figure 12). There was a ceiling effect for PCE, with some of the tracks hitting close to the theoretical maximum of 3.59.

We ran a linear model with genre as the only main effect and found a significant main effect of genre on PCE ($F(11, 140,982) = 759.64$, $p < 0.001$). Post-hoc tests showed that all genres were significantly different than one another (all p -values < 0.05) except for folk and Latin ($t=-0.936$, $p=0.999$), folk and pop ($t=2.484$, $p=0.350$), folk and reggae ($t=1.255$, $p=0.984$), jazz and RnB ($t=3.123$, $p=0.077$; approaching significance), Latin and rap ($t=-2.952$, $p=0.123$), Latin and reggae ($t=2.218$, $p=0.536$), and pop and reggae ($t=-1.054$, $p=0.996$; see Figure 13).

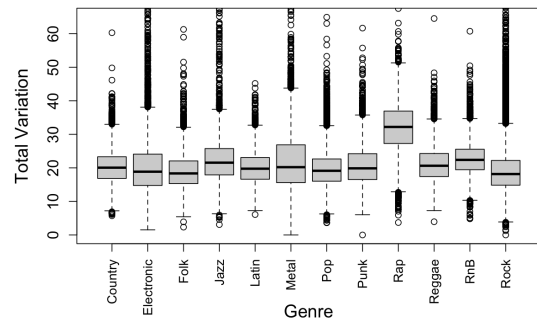


Figure 10: Total variation in each of the twelve genres across the whole dataset. Means are shown with interquartile range, 95% confidence interval error bars, and outliers. There were significant differences in TV between all genres except between country and Latin, country and punk, electronic and punk, electronic and reggae, folk and pop, folk and rock, Latin and pop, and punk and reggae.

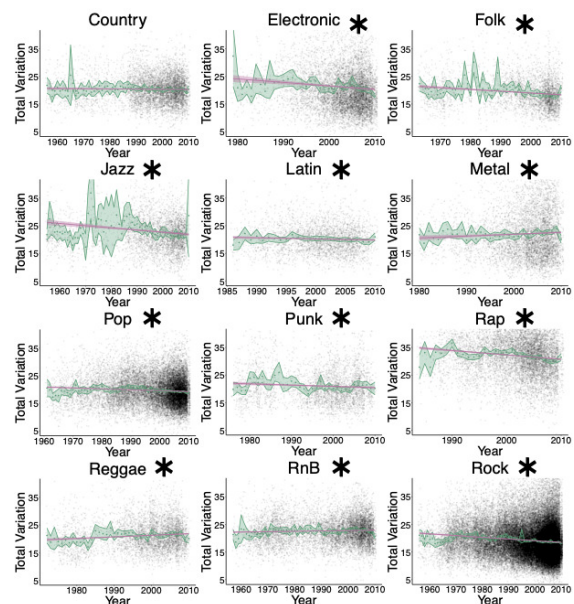


Figure 11: Relationship between TV and year for each genre. “*” denotes a significant effect of year. The red line represents the predicted slope with 95% confidence intervals. The green diamond and ribbon represent the mean per year and the standard error.

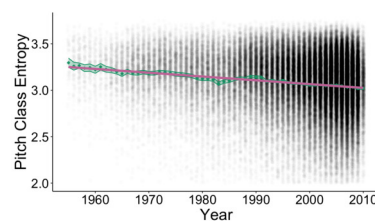


Figure 12: Pitch Class Entropy as a function of year. Each dot represents a song. The red line represents the predicted slope with 95% confidence intervals. The green diamond and ribbon represent the mean PCE per year and the standard error. There was a significant negative correlation between PCE and year

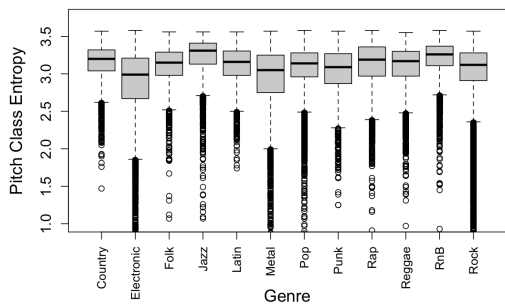


Figure 13: Pitch class entropy in each of the genres. Means are shown with interquartile ranges, 95% confidence interval error bars, and outliers. There were significant differences in PCE between all genres except between folk and Latin, folk and pop, folk and reggae, jazz and RnB, Latin and rap, Latin and reggae, and pop and reggae.

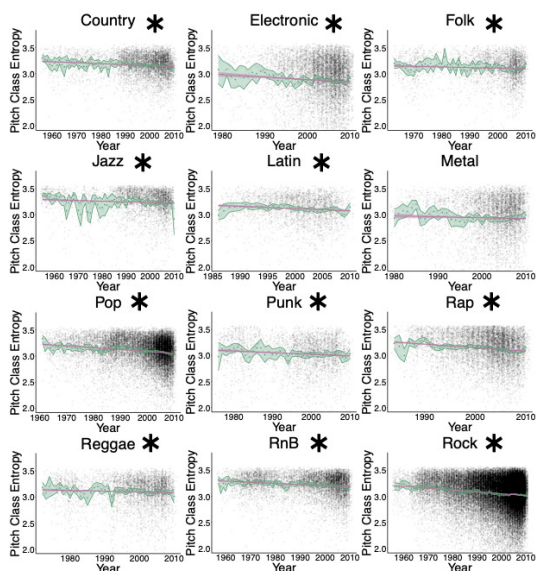


Figure 14: Pitch class entropy in each of the genres. “*” denotes a significant main effect of year. The red line represents the predicted slope with 95% confidence intervals. The green diamond and ribbon represent the mean per year and the standard error.

Finally, we found a significant negative relationship between PCE and year for eleven musical genres (country: $\beta=-0.002$, $t=-7.847$, $p<0.001$; electronic: $\beta=-0.005$, $t=-5.126$, $p<0.001$; folk: $\beta=-0.001$, $t=-3.650$, $p<0.001$; jazz: $\beta=-0.001$, $t=-4.382$, $p<0.001$; Latin: $\beta=-0.004$, $t=-5.047$, $p<0.001$; pop: $\beta=-0.003$, $t=-18.61$, $p<0.001$; punk: $\beta=-0.003$, $t=-4.938$, $p<0.001$; rap: $\beta=-0.007$, $t=-11.61$, $p<0.001$; reggae $\beta=-0.001$, $t=-2.849$, $p=0.004$; RnB: $\beta=-0.002$, $t=-12.64$, $p<0.001$; rock: $\beta=-0.003$, $t=-28.75$, $p<0.001$; see Figure 14). The effect of year for metal music ($\beta=-0.002$, $t=-1.717$, $p=0.086$) approached significance.

6. DISCUSSION

In this study, we analyzed vocal pitch characteristics across years and genres. We found musical genres are often significantly different from one another in mean pitch, total variation, and pitch class entropy. The data generally ex-

hibited a significant negative relationship between year and total variation and year and pitch class entropy, respectively. This was the case both overall and for 8 and 11 musical genres, respectively (see Figure 10 and Figure 13).

If TV and PCE are taken to be measures of musical complexity, these findings could mean vocals, in this dataset at least, are getting less complex over time. This is somewhat in line with previous studies using the MSD. Serrà *et al.* found that newer songs have less variety in pitch transitions and more homogenized timbres, and Parmer *et al.* found that pitch complexity has been generally stable, but loudness and rhythm complexity have decreased [2] [8]. Our findings also parallel those of recent publications looking generally at Western popular music. In a recent study, authors found over five decades, lyrics have become simpler in their vocabulary richness, readability, complexity, and repetitiveness [25]. In another study analyzing popular melodies from 1950 to 2023, Hamilton and Pearce identified melodic revolutions that correspond to decreases in melodic complexity [26].

In our study, we observed that the rap genre had a higher TV than the other genres (see Figure 10), showing that rap songs feature more pitch variation than other musical genres, on average. This could be because rap vocals tend to have less sustained pitch than other genres. Previous work showed that pitch variance in rap music is a complex and significant feature of the genre [27, 28]. Rap music, only coming into prevalence in this dataset in 1984, may have influenced the genres that exhibit a significant positive relationship between year and total variation, counter to the all-genre-pooled negative trend: metal, reggae, and RnB.

We found mean pitch increased over time (see Figure 6). Gender and vocal range are key factors when considering pitch, and genre-specific gender prevalence may exist. However, we did not find a sufficiently reliable gender or vocal range classifier to support further analysis.

Interestingly, mean pitch was the highest for the metal genre, which has a low presence of female vocalists compared to other genres [29]. Therefore, the higher mean pitch of the metal genre cannot be fully explained by a higher prevalence of high-voiced singers. The average mean pitch of metal vocals sits quite high in a typical tenor range [21]. We hypothesize this is because screaming in metal music tends to have a higher f_0 than singing, but more investigation into metal vocals is needed [30].

7. CONCLUSION

In this exploratory research, we examined trends in the vocal lines of 143,152 songs spanning 55 years. Our work has identified relationships between vocal pitch and popular musical genres over time, providing valuable insights into the changing sound of music. We have demonstrated the utility of the methods presented here for studying vocals, and believe they have the potential to be applied to the study of other musical instruments as well as general musical phenomena including historical and cultural trends, changes in musical forms and structures, and stylistic differences across genres and periods.

8. REFERENCES

- [1] T. Bertin-Mahieux, D. P. W. Ellis, B. Whitman, and P. Lamere, “The million song dataset,” in *Proceedings of the 12th International Society for Music Information Retrieval Conference, ISMIR 2011, Miami, Florida, USA*, A. Klapuri and C. Leider, Eds., 2011, pp. 591–596.
- [2] J. Serrà, A. Corral, M. Boguñá, M. Haro, and J. L. Arcos, “Measuring the evolution of contemporary western popular music,” *Scientific reports*, vol. 2, 05 2012.
- [3] U. Shalit, D. Weinshall, and G. Chechik, “Modeling musical influence with topic models,” in *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, ser. JMLR Workshop and Conference Proceedings, vol. 28. JMLR.org, 2013, pp. 244–252.
- [4] D. Stoller, S. Ewert, and S. Dixon, “Wave-u-net: A multi-scale neural network for end-to-end audio source separation,” in *Proceedings of the 19th International Society for Music Information Retrieval Conference, ISMIR 2018, Paris, France*, E. Gómez, X. Hu, E. Humphrey, and E. Benetos, Eds., 2018, pp. 334–340.
- [5] S. Rouard, F. Massa, and A. Défossez, “Hybrid transformers for music source separation,” *IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP 2023, Rhodes Island, Greece, June 4-10, 2023*, pp. 1–5, 2023. [Online]. Available: <https://doi.org/10.1109/ICASSP49357.2023.10096956>
- [6] A. M. Demetriou, A. Jansson, A. Kumar, and R. M. Bittner, “Vocals in music matter: the relevance of vocals in the minds of listeners,” in *Proceedings of the 19th International Society for Music Information Retrieval Conference, ISMIR 2018, Paris, France, September 23-27, 2018*, E. Gómez, X. Hu, E. Humphrey, and E. Benetos, Eds., 2018, pp. 514–520.
- [7] M. Bürgel, L. Picinali, and K. Siedenburger, “Listening in the mix: Lead vocals robustly attract auditory attention in popular music,” *Frontiers in Psychology*, vol. 12, 12 2021.
- [8] T. Parmer and Y. Ahn, “Evolution of the informational complexity of contemporary western music,” in *Proceedings of the 20th International Society for Music Information Retrieval Conference, ISMIR 2019, Delft, The Netherlands, November 4-8, 2019*, A. Flexer, G. Peeters, J. Urbano, and A. Volk, Eds., 2019, pp. 175–182.
- [9] B. P. Gold, M. T. Pearce, E. Mas-Herrero, A. Dagher, and R. J. Zatorre, “Predictability and uncertainty in the pleasure of music: A reward for learning?” *Journal of Neuroscience*, vol. 39, no. 47, pp. 9397–9409, 2019.
- [10] V. K. Cheung, P. M. Harrison, L. Meyer, M. T. Pearce, J.-D. Haynes, and S. Koelsch, “Uncertainty and surprise jointly predict musical pleasure and amygdala, hippocampus, and auditory cortex activity,” *Current Biology*, vol. 29, no. 23, pp. 4084–4092.e4, 2019.
- [11] M. Mauch, R. M. MacCallum, M. Levy, and A. M. Leroi, “The evolution of popular music: Usa 1960–2010,” *Royal Society Open Science*, vol. 2, no. 5, p. 150081, 2015. [Online]. Available: <https://royalsocietypublishing.org/doi/abs/10.1098/rsos.150081>
- [12] F. Thalmann, E. Nakamura, and K. Yoshii, “Tracking the Evolution of a Band’s Live Performances over Decades,” in *Proceedings of the 23rd International Society for Music Information Retrieval Conference*. Bengaluru, India: ISMIR, Dec. 2022, pp. 850–857. [Online]. Available: <https://doi.org/10.5281/zenodo.7342596>
- [13] E. Deruty and F. Pachet, “The MIR perspective on the evolution of dynamics in mainstream music,” in *Proceedings of the 16th International Society for Music Information Retrieval Conference, ISMIR 2015, Málaga, Spain, October 26-30, 2015*, M. Müller and F. Wiering, Eds., 2015, pp. 722–727.
- [14] P. D. Pestana, Z. Ma, J. D. Reiss, A. Barbosa, and D. A. A. Black, “Spectral characteristics of popular commercial recordings 1950–2010,” *AES NY*, 2013.
- [15] E. Oehrle, “Reviews - cantometrics: an approach to the anthropology of music by alan lomax. berkeley, ca: University extension media center, 1976. handbook and cassette available.” *British Journal of Music Education*, vol. 9, no. 1, p. 83–86, 1992.
- [16] M. Panteli, R. Bittner, J. P. Bello, and S. Dixon, “Towards the characterization of singing styles in world music,” *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.
- [17] A. Volk and P. van Kranenburg, “Melodic similarity among folk songs: An annotation study on similarity-based categorization in music,” *Musicae Scientiae*, vol. 16, no. 3, pp. 317–339, 2012.
- [18] H. Schreiber, “Improving genre annotations for the million song dataset,” in *Proceedings of the 16th International Society for Music Information Retrieval Conference, ISMIR 2015, Málaga, Spain, October 26-30, 2015*, M. Müller and F. Wiering, Eds., 2015, pp. 241–247.
- [19] M. Mauch and S. Dixon, “Pyin: A fundamental frequency estimator using probabilistic threshold distributions,” in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 659–663.

- [20] B. McFee, A. Metsai, M. McVicar, S. Balke, C. Thomé, C. Raffel, F. Zalkow, A. Malek, Dana, K. Lee, O. Nieto, D. Ellis, J. Mason, E. Battenberg, S. Seyfarth, R. Yamamoto, viktorandreevich-morozov, K. Choi, J. Moore, R. Bittner, S. Hidaka, Z. Wei, nullmightybofo, D. Herénú, F.-R. Stöter, P. Friesch, A. Weiss, M. Vollrath, T. Kim, and Thassilo, “librosa/librosa: 0.8.1rc2,” <https://doi.org/10.5281/zenodo.4792298>, May 2021.
- [21] T. Stefan Kostka, T. Dorothy Payne, and B. Almén, *Tonal Harmony*. McGraw-Hill Education, 2017. [Online]. Available: <https://books.google.com/books?id=Cs2UAQAACAAJ>
- [22] J. W. Kim, J. Salamon, P. Li, and J. P. Bello, “Crepe: A convolutional representation for pitch estimation,” *2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018, Calgary, AB, Canada, April 15-20, 2018*, pp. 161–165, 2018. [Online]. Available: <https://doi.org/10.1109/ICASSP.2018.8461329>
- [23] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, and D. P. W. Ellis, “Mir_eval: A transparent implementation of common mir metrics,” in *International Society for Music Information Retrieval Conference*, 2014. [Online]. Available: <https://api.semanticscholar.org/CorpusID:17163281>
- [24] J. L. Snyder, “Entropy as a Measure of Musical Style: The Influence of A Priori Assumptions,” *Music Theory Spectrum*, vol. 12, no. 1, pp. 121–160, 03 1990. [Online]. Available: <https://doi.org/10.2307/746148>
- [25] E. Parada-Cabaleiro, M. Mayerl, S. Brandl, M. Skowron, M. Schedl, E. Lex, and E. Zangerle, “Song lyrics have become simpler and more repetitive over the last five decades,” *Scientific Reports*, vol. 14, 03 2024.
- [26] M. Hamilton and M. Pearce, “Trajectories and revolutions in popular melody based on u.s. charts from 1950 to 2023,” *Scientific Reports*, vol. 14, 07 2024.
- [27] M. Ohriner, “Analysing the pitch content of the rapping voice,” *Journal of New Music Research*, vol. 48, pp. 413 – 433, 2019.
- [28] R. Komaniecki, “Vocal pitch in rap flow,” *Intégral*, vol. 34, pp. 25–46, 2020.
- [29] A. Epps-Darling, H. Cramer, and R. T. Bouyer, “Artist gender representation in music streaming,” in *International Society for Music Information Retrieval Conference*, 2020. [Online]. Available: <https://api.semanticscholar.org/CorpusID:232317721>
- [30] S. Mesiä and P. Ribaldini, “Heavy metal vocals : A terminology compendium,” in *Modern Heavy Metal: Markets, Practices and Culture, Helsinki: Aalto University*, 2015. [Online]. Available: <https://api.semanticscholar.org/CorpusID:204842147>

MUSIC PROOFREADING WITH REFINPAINT: WHERE AND HOW TO MODIFY COMPOSITIONS GIVEN CONTEXT

Pedro Ramoneda[‡]
Universtat Pompeu Fabra
Barcelona
pedro.ramoneda@upf.edu

Martin Rocamora
Universtat Pompeu Fabra
Barcelona
martin.rocamora@upf.edu

Taketo Akama
Sony Computer Science Laboratories
Tokyo
taketo.akama@sony.com

ABSTRACT

Autoregressive generative transformers are key in music generation, producing coherent compositions but facing challenges in human-machine collaboration. We propose RefinPaint, an iterative technique that improves the sampling process. It does this by identifying the weaker music elements using a feedback model, which then informs the choices for resampling by an inpainting model. This dual-focus methodology not only facilitates the machine’s ability to improve its automatic inpainting generation through repeated cycles but also offers a valuable tool for humans seeking to refine their compositions with automatic proofreading. Experimental results suggest RefinPaint’s effectiveness in inpainting and proofreading tasks, demonstrating its value for refining music created by both machines and humans. This approach not only facilitates creativity but also aids amateur composers in improving their work.

1. INTRODUCTION

Advanced autoregressive models [1, 2] have enabled the automatic generation of complex musical performances [3–7]. However, while autoregressive models generate music in a strictly forward-moving manner, human composers often follow a more iterative approach, frequently revisiting and refining earlier sections of a piece before proceeding [8–10]. Although there are some iterative methods for music generation [11–13], there are still areas for improvement in terms of controllability and human-in-the-loop aspects, such as inferring where to modify composition and inpainting capability to enable partial modification.

Iterative refinement proved effective for image generation; in particular, Lezama’s Token-Critic [14] shows how feedback mechanisms can enhance image synthesis. Similarly, such feedback could benefit music composition for iteratively refining generated music. Within the spectrum of music composition tools, the Piano Inpainting Application (PIA) [15] stands out for its capabilities for automatic

[‡] Work conducted at Sony Computer Science Laboratories, Inc. Tokyo.



© P. Ramoneda, M. Rocamora & T. Akama. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** P. Ramoneda, M. Rocamora & T. Akama, “Music Proofreading with RefinPaint: Where and How to Modify Compositions given Context”, in *Proc. of the 25th Int. Society for Music Information Retrieval Conf.*, San Francisco, USA, 2024.

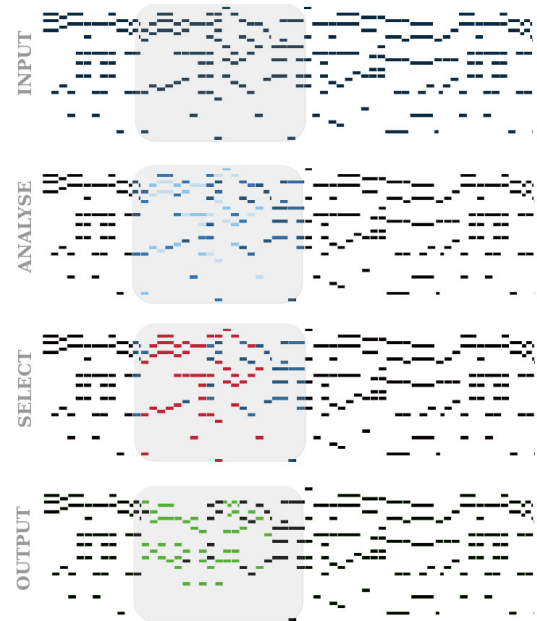


Figure 1: A user selects a MIDI section for enhancement (gray rectangle). Our methodology uses token-level feedback (blue) to highlight critical notes or sequences (red) for regeneration (green). This cycle repeats iteratively.

music generation that addresses the missing parts of musical performances, a technique referred to as inpainting. We highlight their handling of the musical context both before and after the selected gaps, enabling precise note-level inpainting. On account of that, inspired by image generation’s success with iterative feedback and how PIA handles music context, our research explores applying these concepts to enhance controllability, human-in-the-loop functionality, and iterative refinement capability in automatic music generation.

In this work, drawing from Token-Critic and PIA, we propose RefinPaint, which aims to boost automatic inpainting and proofreading in music generation. Our approach includes an iterative process of identifying areas in a composition needing modification and applying inpainting techniques to these areas. In this context, proofreading refers to automatically identifying and correcting errors or inconsistencies in a music composition. This dual-focus methodology facilitates the machine’s ability to improve its automatic inpainting generation through repeated cycles, and offers a valuable tool for humans seeking to refine

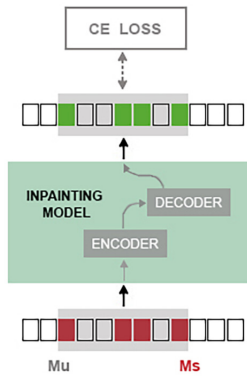


Figure 2: Encoder-decoder architecture for inpainting, given a user-provided mask M_u with a subset mask M_s .

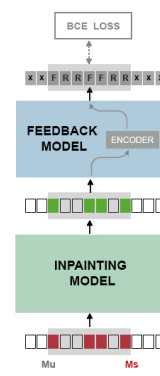


Figure 3: The Feedback algorithm identifies the most realistic tokens by training it to discern between real and synthetic music tokens.

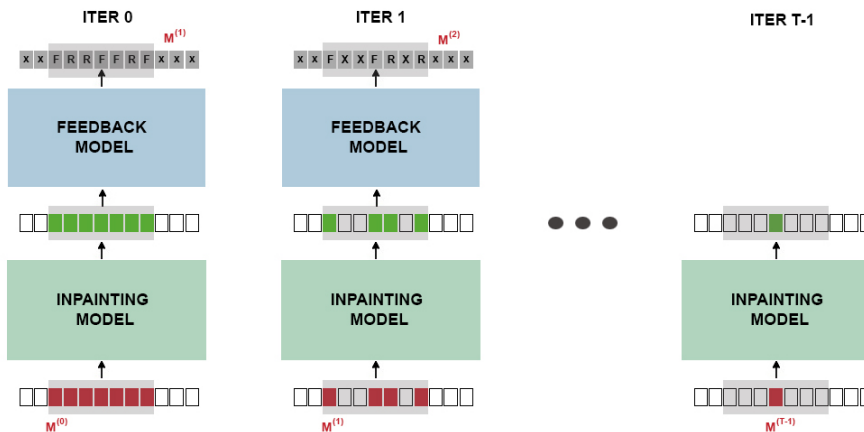


Figure 4: RefinPaint uses inpainting and feedback models to iteratively suggest changes, based on specific note feedback. It reduces the selected tokens in each iteration.

their compositions with automatic proofreading.

Our RefinPaint method is grounded in an autoregressive inpainting model to generate synthetic music tokens and a feedback model trained to distinguish between original and synthetic tokens. This differentiation is key during the sampling stage when deciding on token retention or revision. RefinPaint takes an iterative approach, integrating feedback into the inpainting model for selectively regenerating parts in each iteration, as Figure 1 shows. In contrast to Token-Critic, RefinPaint focuses on modifying a specific part of a composition using a contextual model and exposes the intermediate outputs of the autoregressive inpainting model to human inspection in each iteration.

The human-in-the-loop approach we propose allows for selecting the number of tokens to modify and revise the analysis heatmap at each iteration, as described in the following section. Through experimentation, we confirm RefinPaint’s effectiveness in inpainting and proofreading tasks, demonstrating its utility for enhancing music created by both machines and humans. Finally, we provide a companion page featuring examples¹ and the code along with the trained models of RefinPaint for reproducibility².

¹At: <https://refinpaint.github.io/>

²At: <https://github.com/ta603/RefinPaint>

2. METHODOLOGY

Our proposed methodology employs two models: an inpainting model \mathcal{I} , and a feedback model \mathcal{F} , alongside our iterative algorithm RefinPaint. Initially, \mathcal{F} identifies areas within a MIDI file that need improvement based on the specific criteria described in Section 2.2. It uses a heatmap for detailed MIDI token-level feedback, allowing one to assess the context and relevance of each note in the selected region. Then, model \mathcal{I} can regenerate the selected tokens considering the feedback, as described in Section 2.1. The methodology involves using both models iteratively with RefinPaint and encompasses three main stages: training the inpainting model (Section 2.1.1), training the feedback model (Section 2.2.1), and finally executing the iterative process for MIDI sequence generation (Section 2.3).

2.1 Inpainting model (\mathcal{I})

The inpainting model aims to predict, or fill in, missing parts of a MIDI sequence based on a given mask. We adopt an encoder-decoder architecture for sequence-to-sequence tasks, as shown in Figure 2, inspired by the PIA study for music generation [15]. This model involves an encoder converting input data into a latent representation and a decoder predicting the final output.

With an anti-causal mask, self-attention within the encoder prevents future data access, while with a causal mask, self-attention within the decoder limits access only to previous data. With an identity mask, cross-attention enforces positional alignment between the encoder and decoder outputs, which is helpful for aligned sequence tasks.

The attention mechanisms are defined as follows, where M_{type} is the mask type (anti-causal, causal, or identity):

$$\text{Attention}(Q, K, V) = \text{softmax} \left(\frac{QK^\top}{\sqrt{d_k}} \odot M_{\text{type}} \right) V. \quad (1)$$

This structure enhances the capability of the model to handle bidirectional input-output relationships, essential for inpainting, where future context influences the generation process. Furthermore, we add an extra binary embedding to the encoder input with information about the mask M_s —the tokens to regenerate—for the inpainting model.

2.1.1 Training the Inpainting Model (\mathcal{I})

The training process is outlined in Algorithm 1. A batch x is sampled from the MIDI dataset \mathcal{D} , and a random fragment M_u is chosen for each sample in x with a length determined by t_1 . It is important to note that t_1 refers to the length in terms of the token sequence, rather than the MIDI duration. Consequently, a random mask M_s , with the masking ratio controlled by $\gamma(t_2)$, is then applied to M_u . The forward pass of the model calculates the loss using the batch x , the mask M_s , and the Cross Entropy (CE) loss function to evaluate the difference between the predicted outputs and the actual labels. The model is subsequently updated via gradient descent. The function γ , a cosine scheduler, dynamically adjusts the masking ratio. It operates on a domain defined by a random variable t_2 within the interval $[0, 1]$. Specifically, for any chosen value t_2 drawn uniformly from the interval $[0, 1]$, the value undergoes a cosine transformation γ to determine the masking ratio, where $\gamma(t_2) = \cos\left(\frac{\pi t_2}{2}\right)$.

Algorithm 1 Training the Inpainting model (\mathcal{I})

Require: MIDI dataset \mathcal{D} , Inpainting model \mathcal{I}

- 1: **while** convergence **do**
- 2: $x \sim \mathcal{D}$ ▷ Sample batch
- 3: $t_1 \sim U(0.1, 0.6), t_2 \sim U(0, 1)$
- 4: $M_u \leftarrow \text{Fragment}(x, t_1)$
- 5: $M_s \leftarrow \text{Random Masking}(M_u, \gamma(t_2))$
- 6: $L \leftarrow \text{ForwardInpaintingModel}(\mathcal{I}, x, M_s)$
▷ model forward and compute loss
- 7: GradientDescent(L)
- 8: **end while**

2.2 Feedback model (\mathcal{F})

We employ an encoder-only transformer architecture for the feedback phase that classifies music tokens as fake or real. We use this output distribution to select the k most realistic tokens to retain while the others are regenerated. Unlike the encoder-decoder inpainting model, \mathcal{I} ,

this model processes the input through a parallel and bidirectional attention mechanism without employing any attention masks, thus facilitating an unrestricted analysis of the musical context. Additionally, we add an extra binary embedding to the encoder input with information about the mask M_u —the selected fragment—for the feedback model.

2.2.1 Training the Feedback model (\mathcal{F})

Algorithm 2 Training the Feedback model (\mathcal{F})

Require: MIDI dataset \mathcal{D} , Inpainting model \mathcal{I} , Feedback model \mathcal{F}

- 1: **while** convergence **do**
- 2: $x \sim \mathcal{D}$ ▷ Sample batch
- 3: $t_1 \sim U(0.1, 0.6), t_2 \sim U(0, 1)$
- 4: $M_u \leftarrow \text{Fragment}(x, t_1)$
- 5: $M_s \leftarrow \text{Random Masking}(M_s, \gamma(t_2))$
- 6: $\hat{x} \leftarrow \mathcal{I}(x, M_u)$
- 7: $L \leftarrow \text{ForwardFeedbackModel}(\mathcal{F}, \hat{x}, M_u)$
▷ model forward and compute loss
- 8: GradientDescent(L)
- 9: **end while**

After training the inpainting model \mathcal{I} , we train an encoder-only feedback model \mathcal{F} . This model aims to evaluate the output from \mathcal{I} , offering feedback on the composition quality of each music fragment denoted by M_u .

One ideal way of training \mathcal{F} would involve a vast dataset of computer- or human-generated music compositions and human experts’ revisions for inpainting and proofreading applications. Instead, we propose a more feasible synthetic training strategy, described in Algorithm 2. The inpainting model \mathcal{I} generates tokens within the selected fragment of a music piece, M_u , which we label as ‘Fake’, while we label as ‘Real’ the original unchanged tokens. We utilize these labels to instruct \mathcal{F} , following the process illustrated in Figure 3.

The training of \mathcal{F} is based on the output of \mathcal{I} . We begin by sampling a batch x from the dataset \mathcal{D} , then apply masking M_s and M_u . Model \mathcal{I} regenerates specific tokens within x , yielding a modified output \hat{x} . Model \mathcal{F} then assesses each token of \hat{x} against M_s , categorizing them as ‘Real’ or ‘Fake’. The loss L for \mathcal{F} is computed using the Binary Cross Entropy (BCE) loss function, and is minimized through gradient descent. The outcome is a heatmap for M_u , which indicates the probability of each token being ‘Real’ or ‘Fake’, determined by the sigmoid activation of the model output.

2.3 Generation of MIDI sequences (RefinPaint)

We capitalize on the strengths of the inpainting and feedback models for the iterative MIDI sequence generation. The process shown in Figure 4 begins with a MIDI sequence x introduced by the user, setting the stage for a loop that spans a predetermined number of iterations T .

Initially, the user selects the fragment to be modified $x_m^{(0)}$ and sets the initial selection rate $k = 0$ for complete inpainting. Alternatively, different values for k allow the

user to control how much of the content to keep in the selected fragment when proofreading.

In the proposed Algorithm 3, at each iteration t , the inpainting model \mathcal{I} generates a new version of the sequence \hat{x} , based on the current masked input $x_m^{(t)}$. In the human-in-the-loop scenario, the user can then adjust this generated sequence. The feedback model \mathcal{F} evaluates \hat{x} and provides a new mask $M^{(t+1)}$, which the user may also modify. This mask highlights the tokens that are deemed most realistic. The number of selected realistic tokens k follows a decreasing function γ of the iteration t , which models the increasing confidence in the tokens produced over time. Moreover, we add an extra binary embedding to the encoder input with information about the mask M —the given context—where M changes over iterations.

Refining the music sequence through each iteration aims to achieve a compositional process that closely aligns with that of a human composer so that the user intervention becomes interpretable and natural. It fosters a collaborative environment between the user and the machine and tailors the generation process to the user’s specific directives and preferences.

Algorithm 3 Generation Algorithm (RefinPaint)

Require: Inpainting model \mathcal{I} , Feedback model \mathcal{F} , masked MIDI $x_m^{(0)}$, No. masked tokens N , No. iterations T

- 1: **for** $i = 0$ to $T - 1$ **do**
 - 2: $k = \lceil \gamma \left(\frac{i}{T} \right) \cdot N \rceil$
 - 3: $\hat{x} \leftarrow \mathcal{I}(x_m^{(i)})$
 - 4: **if** $i \neq T - 1$ **then**
 - 5: $M^{(i+1)} \leftarrow \mathcal{F}(\hat{x})$
 - 6: $x_m^{(i+1)} \leftarrow k\text{-realistic tokens}(\hat{x}, M^{(i+1)}, k)$
 - 7: **end if**
 - 8: **end for**
-

3. RELATED WORK

Automatic music generation has rapidly advanced recently. Significant progress has been made [4–6], especially in solo piano compositions [3, 7, 15], through the capabilities of autoregressive models in producing coherent musical outputs. However, several challenges remain for creating successful interactions with humans [3, 11, 15–22].

Previous work has explored various approaches to generate music iteratively and allowed for partial modification—often referred to as inpainting—which enhances controllability. Among them, sequential handling of musical elements has been a common strategy, as in models like DeepBach [11] and Coconet [12]. Although these models allow for inpainting and iterative generation, they often rely on random iterations without a mechanism for discriminative feedback to guide improvements. This lack of directed refinement contrasts with the human compositional process, which typically involves iterative improvements based on evaluative feedback. Our proposed approach addresses this limitation by incorporating a feedback model that identifies areas for improvement for both

humans and machines to refine the composition.

Although it is not designed as an inpainting model, ES-Net’s approach to music generation integrates generative and discriminative capabilities in one model [13], with a feature for correcting past errors for iterative refinement. Our model differs significantly: it takes into account the context of the selected fragment, could improve any existing inpainting model, and can handle general MIDI formats. In [23], the authors propose a GAN model for piano music composition with a discriminator model that discerns real and fake compositions in the training process. However, it does not give feedback on which generated parts are good or bad and does not create compositions iteratively. Yet, the application of discriminative feedback in music generation, particularly in a manner that mimics human iterative refinement, remains largely unexplored.

Finally, inpainting models in music have seen various approaches but remain less studied compared to their counterparts in image generation [24]. They typically focus on quantized scores, with significant contributions like Gibbs sampling for Bach chorales [11] and RNN-based melodies inpainting [25]. Studies on transformers for multitrack inpainting have advanced the field, such as MMM [26], which utilizes a decoder architecture akin to GPT2 [2], and PIA [15], which uses a specialized transformer design. We chose PIA over MMM as a ground element in this work, given it is capable of working in the token level or larger contexts and inpainting multiple little fragments at the same time, similar to Token-Critic’s generator [14].

4. EXPERIMENTAL SETUP

4.1 Data preparation

Our study utilizes the Lakh MIDI dataset (LMD), an extensive collection of approximately 170k unique multi-track MIDI files, compiled by Colin Raffel for music research [27]. The dataset offers a wide variety of music, albeit with varying quality due to its internet-sourced nature. Despite this, the volume and diversity of the LMD dataset make it a valuable asset for our proofreading task. We extracted only the piano parts, totaling 120,000 tracks.

We tokenize the piano tracks using REMI (REvamped MIDI-derived events) [16], a music representation method that converts MIDI events into a structured format optimized for Transformer-based models that significantly enhances their ability to comprehend and produce music. REMI categorizes music elements into distinct event types, including timing for rhythm and note events for melody, but we exclude velocity events for simplicity. Specifically, we use a modified version of REMI tailored for handling single-track piano performances, as implemented in [28]. The dataset was split into training (hashes 0–d), validation (hash e), and testing (hash f) segments, based on each file’s MD5 hash’s leading digit, akin to previous methods [5, 6]

4.2 Model development

We train the inpainting and feedback models with the AdamW optimizer, using eighty per cent of the dataset for training and the remainder for validation. Each epoch consists of a randomly selected fragment from the training set,

512 tokens in length. We also employ an augmentation procedure that transposes the pitch tokens of a sequence by adding or subtracting up to 6 semitones. For the inpainting model, we apply a cross-entropy loss and use the maximum batch size that our system can handle; a single V100 GPU with 16GB allows for 48 samples. The encoder-decoder inpainting model comprises 12 layers: 4 encoder layers and 8 decoder layers, similar to the original PIA, with 8 heads and an embedding dimension of 512. We employ a cosine scheduler for training, with 16,000 warmup steps, reaching up to a 0.0006 learning rate. The feedback model consists of 6 layers, with an embedding dimension of 512, a dropout rate of 0.1, 8 heads, and the same cosine scheduler. Finally, we acknowledge that optimizing these models was not the main focus of this paper, so there might be better hyperparameter values.

In the particular case of proofreading without human intervention, i.e. for evaluation purposes, the final output is the iteration that maximizes the feedback model probability distribution. Using a sigmoid function, the model determines whether each token in a sequence is fake or real. By averaging the output probabilities, we calculate a global feedback score (GFS) for the sequence’s overall realism and select the best regeneration output based on it.

5. INPAINTING RESULTS

5.1 Divide and conquer with the inpainting model

We conducted an experiment to explore how the model’s inpainting performance is affected by the percentage of tokens to inpaint in a selected fragment. We hypothesize that the more tokens to inpaint, the harder the problem is, so the model performance is lower. The experiment uses the inpainting model trained as detailed in section 2.1.1, and we report its Negative Log-Likelihood (NLL) loss and perplexity of the next predicted token. The evaluation covered the entire test set, with masking ratios ranging from 1 (fully masked) to 0 (no tokens to inpaint) and a fixed 30% fragment size rate of the 512 tokens sequence. Results shown in Table 1 indicate better performance with reduced masking, confirming our hypothesis. Notably, the average Perplexity value is less than half at 0.05 compared to the 1.0 masking ratio. This finding is crucial for RefinPaint’s effectiveness as it reduces the number of tokens to be inpainted in subsequent iterations, considering the iterative process as a top-to-bottom strategy.

5.2 Objective evaluation of proofreading inpainting

This section conducts a comparative analysis between the reference inpainting output, as described in [15] (PIA), and our enhanced method. Our method applies the RefinPaint proofreading process to the initial PIA’s inpainting output over ten iterations and is referred to as ‘Ours’. For fragment sizes of 50%, 30%, and 10% of the 512-token test sequences, we computed 1,000 instances each. It is important to note that the PIA method discussed is our reimplementation, since the original code was not available.

Table 2 shows the average global feedback score (GFS), computed as explained in Section 4.2, and the number of

masking ratio	NLL	AVG PPL
0.05	0.56	0.31
0.10	0.58	0.33
0.15	0.58	0.34
0.20	0.58	0.33
0.40	0.64	0.41
0.60	0.70	0.49
0.80	0.77	0.59
1.00	0.86	0.73

Table 1: Summary of the inpainting experiment with different masking ratios. A masking ratio of 1.0 corresponds to being fully masked, and 0 indicates no masking. The standard deviation is less than 0.01 in all the experiments.

evaluations in which each algorithm outperforms the other (Wins) and in which their scores are the same (Ties). Table 3, on the other hand, focuses on the comparison between PIA and Ours, employing the NLL loss, a metric of the next token prediction in generated music. This metric, derived from an autoregressive model we trained explicitly from scratch to assess the inpainting results, is a benchmark metric in our evaluation. Similar evaluations have been employed in previous studies in natural language processing [29] and music generation [30]. Consequently, our study employs a 12-layer Transformer-based autoregressive model with REMI representation. Our goal is to assess the similarity between the distribution of musical elements in inpainted sections and those in the original dataset, including aspects such as rhythms, harmony, or melodies. A lower NLL loss indicates a more accurate prediction of the next token, reflecting a closer approximation to the dataset’s inherent musicality. Note we assess this metric over the entire output sequence.

	GFS (↑)		Wins		Ties
	PIA	Ours	PIA	Ours	
50%	0.458	0.696	0	870	130
30%	0.515	0.730	0	886	114
10%	0.650	0.803	0	891	209

Table 2: Comparison of global feedback scores (GFS) between PIA and the proposed RefinPaint methodology, Ours. Higher values indicate better performance.

	NLL (↓)		Wins		Ties
	PIA	Ours	PIA	Ours	
50%	2.01	1.97	330	541	129
30%	1.68	1.66	347	533	120
10%	1.63	1.62	321	457	222

Table 3: Comparison of Negative Log Likelihood (NLL) between PIA and the proposed RefinPaint methodology, Ours. Lower values indicate better performance.

Results in Table 2 indicate that our model’s GFS score is generally better than the baseline, suggesting that the optimization goal of the RefinPaint iterative process is met.

The PIA model never wins because this experiment selects the best GFS of all the iterations, as mentioned in Section 4.2. Although dynamic programming or genetic algorithms could enhance the process, this study uses a simpler method, focusing on the iteration with the highest GFS.

In Table 3, RefinPaint consistently achieves a slightly lower average NLL loss than PIA, suggesting that the inpainted content by RefinPaint is more consistent with the original dataset used for training. Furthermore, RefinPaint wins more evaluations than PIA across all the percentages of fragment size evaluated. This further underscores the enhanced performance of RefinPaint in producing sequences more akin to human compositions. However, comparing both tables, we acknowledge that higher GFS does not always imply a better NLL loss, calling for other types of evaluation, as addressed in the next section.

5.3 Listening test of proofreading inpainting

While computational metrics provide valuable insights into the quality of our inpainted music sections, human perception adds another perspective for evaluating musical quality and appeal. A user-based evaluation was conducted to capture a holistic view of the inpainted outputs' musicality.

For each experiment, which involved 50%, 30%, and 10% fragments of inpainted content, 15 different annotators evaluated both the first iteration of inpainted content (PIA) and the complete iterative process of RefinPaint (Ours) for ten iterations. Participants were exposed to two scenarios, Experiment 1 and Experiment 2: one from the PIA model and one from our RefinPaint model. The order in which these pairs were presented was randomized to avoid any bias. Additionally, we provided the original music fragment without the inpainted content for reference. Participants listened to both the PIA and RefinPaint versions before making their evaluations. They were asked to assess the inpainted content's quality by comparing it to the original fragment, focusing specifically on coherence and creativity. To make their choice, participants were given four options to prevent bias: 'Experiment 1,' 'Maybe Experiment 1,' 'Maybe Experiment 2,' and 'Experiment 2'.

Figure 5 shows the listening test results. Firstly, PIA got lower preference scores than RefinPaint for the different fragment size conditions. In addition, RefinPaint's performance for different fragment sizes shows that the coherence scores increase as the fragment size gets larger, even if the creativity varies. This means that as there is more to inpaint, RefinPaint gets better at being coherent. In contrast, PIA does not show such a strong trend.

The quantitative and qualitative evaluations point towards a clear trend: RefinPaint tends to yield superior inpainting results when proofreading machine inpainted sections compared to the baseline. Our methodology produces music sequences that are more consistent, perceptually closer to the original, and preferred by listeners.

6. CASE OF STUDY ON PROOFREADING AMATEUR COMPOSITIONS

We conducted an additional study to explore the proposed system's capabilities for proofreading music compositions

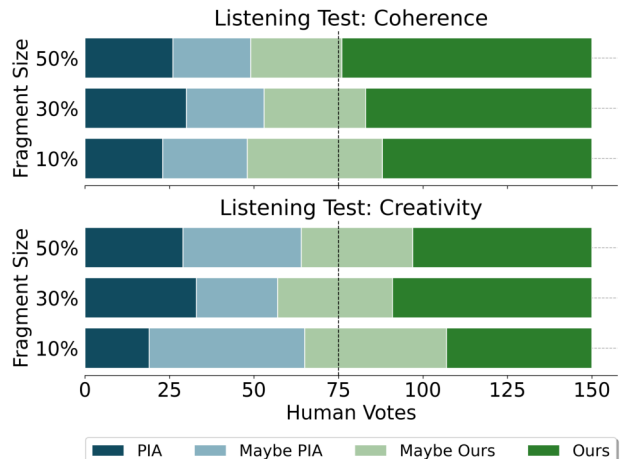


Figure 5: Results of the participants' votes for the listening test comparing PIA and RefinPaint (Ours) along different fragment sizes (50%, 30%, and 10%).

by humans. Given the intrinsic difficulties of such a study and due to practical restrictions, we limited our experiment to four amateur composers—two with classical music training and two with modern popular music training.

Participants used a straightforward proofreading interface that enables bar selection for regeneration, allowing them to choose how much of the content to keep in certain sections of their work, as described in 2.3. Additionally, we allowed the users to change the RefinPaint feedback in the selected area and experiment with the tools by conducting as many trials as they wanted.

After testing our inpainting tool on a 30-second music piece, participants responded to questions about their experience. They evaluated whether the tool (i) enhanced their original draft, (ii) sparked new ideas, (iii) could save time over manual proofreading, and (iv) was something they would use in the future. All chose "yes" for (i), (iii) and (iv) with three "yes" and one "maybe" for (ii), suggesting time efficiency as a key advantage and providing an overall positive view of the tool.

The positive feedback prompted us to showcase the proofread compositions on our companion website. Participants suggested the tool could be particularly effective in overcoming creative blocks, noting that inspiring ideas stemmed from all iterations, not just the last one. Additionally, two participants especially valued the option to alter tokens within the RefinPaint selection.

7. CONCLUSION

In conclusion, our novel approach, RefinPaint, significantly enhances music generation by identifying and improving weaker musical elements through iterative feedback. Its effectiveness in both inpainting and proofreading tasks promises a new direction for creative assistance and quality enhancement in compositions by humans and machines alike. Future work could fruitfully extend the research to multitrack compositions and explore control mechanisms for this model, such as conditioning by harmony, rhythm, genre, or other musical factors.

8. ETHICS STATEMENT

While RefinPaint can represent a significant leap forward in music composition technology, ensuring ethical deployment and use is crucial. We advocate for a future where such technologies support and enrich the creative process, complementing rather than displacing human creativity. While RefinPaint aims to democratize music creation, making it accessible and achievable for amateurs, there is a risk that professional musicians and composers could feel their roles and contributions are being undermined or replaced by machines. It is essential to strike a balance where this technology serves as a tool for enhancement and learning rather than a substitute for human creativity. Furthermore, it will be vital to establish guidelines that protect the intellectual property rights of original compositions, whether entirely human-made or AI-assisted.

9. REFERENCES

- [1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems (NeurIPS)*, 2017.
- [2] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI Blog*, vol. 1, no. 8, 2019.
- [3] C.-Z. A. Huang, A. Vaswani, J. Uszkoreit, N. Shazeer, I. Simon, C. Hawthorne, A. M. Dai, M. D. Hoffman, M. Dinculescu, and D. Eck, "Music transformer: Generating music with long-term structure," in *7th International Conference on Learning Representations, (ICLR)*, 2019.
- [4] S.-L. Wu and Y.-H. Yang, "Compose & embellish: Well-structured piano performance generation via a two-stage approach," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023.
- [5] J. Thickstun, D. Hall, C. Donahue, and P. Liang, "Anticipatory music transformer," *arXiv preprint arXiv:2306.08620*, 2023.
- [6] B. Yu, P. Lu, R. Wang, W. Hu, X. Tan, W. Ye, S. Zhang, T. Qin, and T.-Y. Liu, "Museformer: Transformer with fine-and coarse-grained attention for music generation," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.
- [7] N. Fradet, J.-P. Briot, F. Chhel, A. E. F. Seghrouchni, and N. Gutowski, "Byte pair encoding for symbolic music," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023.
- [8] D. Collins and M. Dunn, "Problem-solving strategies and processes in musical composition: Observations in real time," *Journal of Music, Technology & Education*, vol. 4, no. 1, pp. 47–76, 2011.
- [9] P. Burnard, *Musical creativities in practice*. OUP Oxford, 2012.
- [10] B. Jacob, "Algorithmic composition as a model of creativity," *Organised Sound*, vol. 1, pp. 157 – 165, 1996.
- [11] G. Hadjeres, F. Pachet, and F. Nielsen, "Deepbach: a steerable model for bach chorales generation," in *International Conference on Machine Learning (ICML)*, 2017.
- [12] C.-Z. A. Huang, T. Cooijmans, A. Roberts, A. Courville, and D. Eck, "Counterpoint by convolution," in *Proc. of the 18th Int. Society for Music Information Retrieval Conf. (ISMIR)*, 2017.
- [13] W. Chi, P. Kumar, S. Yaddanapudi, R. Suresh, and U. Isik, "Generating music with a self-correcting non-chronological autoregressive model," in *Proc. of the 21th Int. Society for Music Information Retrieval Conf. (ISMIR)*, 2020.
- [14] J. Lezama, H. Chang, L. Jiang, and I. Essa, "Improved masked image generation with token-critic," in *European Conference on Computer Vision (ECCV)*, 2022.
- [15] G. Hadjeres and L. Crestel, "The piano inpainting application," *arXiv preprint arXiv:2107.05944*, 2021.
- [16] C.-Z. A. Huang, T. Cooijmans, A. Roberts, A. Courville, and D. Eck, "Pop music transformer: Beat-based modeling and generation of expressive pop piano compositions," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020.
- [17] T. Akama, "Controlling symbolic music generation based on concept learning from domain knowledge," in *Proc. of the 20th Int. Society for Music Information Retrieval Conf. (ISMIR)*, 2019.
- [18] —, "Connective fusion: Learning transformational joining of sequences with application to melody creation," in *Proc. of the 21st Int. Society for Music Information Retrieval Conf. (ISMIR)*, 2020.
- [19] —, "A contextual latent space model: Subsequence modulation in melodic sequence," in *Proc. of the 22nd Int. Society for Music Information Retrieval Conf. (ISMIR)*, 2021.
- [20] C. Payne, "Musenet," <https://openai.com/blog/musenet>, 2019.
- [21] G. Hadjeres and L. Crestel, "Vector quantized contrastive predictive coding for template-based music generation," *arXiv preprint*, 2020.
- [22] Y.-J. Shih, S.-L. Wu, F. Zalkow, M. Muller, and Y.-H. Yang, "Theme transformer: Symbolic music generation with theme-conditioned transformer," *IEEE Transactions on Multimedia*, 2022.

- [23] A. Muhamed, L. Li, X. Shi, S. Yaddanapudi, W. Chi, D. Jackson, R. Suresh, Z. C. Lipton, and A. J. Smola, “Symbolic music generation with transformer-gans,” in *35th AAAI Conference on Artificial Intelligence*, 2021.
- [24] O. Elharrouss, N. Almaadeed, S. Al-Maadeed, and Y. Akbari, “Image inpainting: A review,” *Neural Processing Letters*, vol. 51, pp. 2007–2028, 2020.
- [25] G. Hadjeres and F. Nielsen, “Anticipation-rnn: Enforcing unary constraints in sequence generation, with application to interactive music generation,” *Neural Computing and Applications*, vol. 32, no. 4, pp. 995–1005, 2020.
- [26] J. Ens and P. Pasquier, “Mmm: Exploring conditional multi-track music generation with the transformer,” *arXiv preprint arXiv:2008.06048*, 2020.
- [27] C. Raffel, “Learning-based methods for comparing sequences, with applications to audio-to-midi alignment and matching,” Ph.D. dissertation, Columbia University, 2016.
- [28] N. Fradet, J.-P. Briot, F. Chhel, A. El Fallah Seghrouchni, and N. Gutowski, “MidiTok: A python package for MIDI file tokenization,” in *Extended Abstracts for the Late-Breaking Demo Session of the 22nd International Society for Music Information Retrieval Conference*, 2021.
- [29] A. Wang and K. Cho, “BERT has a mouth, and it must speak: BERT as a Markov random field language model,” in *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, Jun. 2019.
- [30] A. Roberts, J. Engel, C. Raffel, C. Hawthorne, and D. Eck, “A hierarchical latent vector model for learning long-term structure in music,” in *International conference on machine learning (ICML)*, 2018.

NOTEWISE EVALUATION FOR MUSIC SOURCE SEPARATION: A CASE STUDY FOR SEPARATED PIANO TRACKS

Yigitcan Özer¹ Hans-Ulrich Berendes¹ Vlora Arifi-Müller¹
Fabian-Robert Stöter² Meinard Müller¹

¹International Audio Laboratories Erlangen, Germany

²AudioShake, Inc.

{yigitcan.oezer, meinard.mueller}@audiolabs-erlangen.de

ABSTRACT

Deep learning has significantly advanced music source separation (MSS), aiming to decompose music recordings into individual tracks corresponding to singing or specific instruments. Typically, results are evaluated using quantitative measures like signal-to-distortion ratio (SDR) computed for entire excerpts or songs. As the main contribution of this article, we introduce a novel evaluation approach that decomposes an audio track into musically meaningful sound events and applies the evaluation metric based on these units. In a case study, we apply this strategy to the challenging task of separating piano concerto recordings into piano and orchestra tracks. To assess piano separation quality, we use a score-informed nonnegative matrix factorization approach to decompose the reference and separate piano tracks into notewise sound events. In our experiments assessing various MSS systems, we demonstrate that our notewise evaluation, which takes into account factors such as pitch range and musical complexity, enhances the comprehension of both the results of source separation and the intricacies within the underlying music.

1. INTRODUCTION

Music source separation (MSS) is a key task in Music Information Retrieval (MIR), involving the separation of a musical mixture into individual components like vocals, instruments, and other sound elements [1]. Deep learning techniques have significantly advanced MSS, especially in scenarios with sufficient training data. In particular, this progress is evident in popular music separation, making use of the existence of multitrack recordings inherent in the production process [2–5]. In scenarios with limited training data, systems are often trained using artificially generated mixes through synthesis techniques [6,7] or data augmentation approaches [8,9]. An example of such a sce-

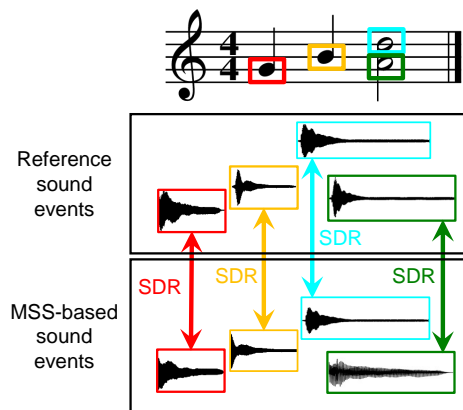


Figure 1: Illustration of the proposed evaluation method for music source separation (MSS), considering signal-to-distortion ratio (SDR) values based on notewise sound events rather than entire recordings.

nario, also addressed in this paper, is presented in [10], where the goal is to separate piano concertos into piano and orchestra tracks.

Extensive efforts have been devoted to evaluating and understanding existing MSS systems. Specifically, in the realm of popular music, evaluation campaigns like the Signal Separation Evaluation Campaign (SiSEC) [11] and the Music Demixing Challenge (MDX) [12] have significantly contributed to the comparison of current systems. In these campaigns, along with evaluations in most approaches described in the literature, one typically relies on quantitative evaluation measures such as the signal-to-distortion ratio (SDR) [13]. These measures are computed and aggregated over audio excerpts or even entire recordings, offering ease of computation and convenience for comparison. However, it is well recognized that such measures provide limited insights into the effectiveness of source separation methods [14, 15]. On the other hand, designing perceptually or musically more relevant measures is challenging, and performing listening tests is often cumbersome and infeasible.

In this paper, we introduce a novel evaluation methodology aimed at attaining a more nuanced understanding of separation quality. This involves comparing a reference signal with a separated signal, utilizing an evaluation



© Yigitcan Özer, Hans-Ulrich Berendes, Vlora Arifi-Müller, Fabian-Robert Stöter, Meinard Müller. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Yigitcan Özer, Hans-Ulrich Berendes, Vlora Arifi-Müller, Fabian-Robert Stöter, Meinard Müller, “Notewise Evaluation for Music Source Separation: A Case Study for Separated Piano Tracks”, in *Proc. of the 25th Int. Society for Music Information Retrieval Conf.*, San Francisco, United States, 2024.

metric based on musically meaningful sound units instead of the entire excerpt. To achieve this, we employ score-informed nonnegative matrix factorization (NMF) [16] to decompose signals into notewise sound events. Then, we calculate SDR values for individual units before aggregating this information in various ways (see Figure 1). This methodology draws conceptual parallels to the evaluation of tasks where automatic speech recognition (ASR) is used as a downstream task. For example, Chen et al. [17] computed word-level and utterance-level metrics to evaluate the quality of the speech separation system.

In a case study, we apply this methodology to the intricate task of separating piano concerto recordings into piano and orchestra tracks. Besides utilizing the Piano Concerto Dataset (PCD) [18], which comprises piano concerto excerpts performed by five pianists in four distinct acoustic settings, we generated piano scores for all the excerpts. We then employed music synchronization techniques [19, 20] to align these scores with all recorded excerpts. As an additional contribution to this paper, we release these annotations, thereby adding a score-based layer to the PCD collection.

In systematic experiments, we apply our evaluation methodology to effectively compare several academic and commercial source separation systems. Our approach uncovers general trends and yields insights into how separation quality is affected by factors like pitch range and musical complexity. In particular, it allows users to explore evaluations in-depth by pinpointing complex passages and challenging sound units where source separation systems tend to fail. Along these lines, we provide qualitative discussions that deepen insights into the behavior of source separation systems and the complexity of the underlying music.

The remainder of the paper is organized as follows. In Section 2, we review relevant literature on source separation and introduce the MSS models used for separating piano concertos. Subsequently, in Section 3, we elaborate on the score-based extension of PCD and outline our evaluation approach, covering NMF-based audio decomposition and notewise SDR-based metrics. In Section 4, we provide details on the experimental settings and report our empirical findings. Finally, in Section 5, we conclude and discuss potential directions for future work.

2. MUSIC SOURCE SEPARATION

As mentioned earlier, the decomposition of music recordings into individual sound components has garnered significant attention in academia and industry in recent years [1–5, 21–23]. While there is a multitude of approaches and architectures proposed in the literature, one can broadly distinguish between spectral-based, waveform-based, and hybrid models. Spectral-based models, such as Open-Unmix (UMX) [2] or Spleeter (SPL) [3], estimate the magnitude spectrograms of target musical sources given the magnitude spectrogram of an input mixture. Techniques like binary masking, soft masking, or multichannel Wiener filtering are then employed to reconstruct the separated audio

Model ID	Domain	Size (MB)	TS (Hours)
UMX	Spectrogram	34	52
SPL	Spectrogram	75	52
DMC	Waveform	510	52
HDMC	Hybrid	319	52
AudioShake	Hybrid	N/A	500+

Table 1: MSS models considered in our experiments. TS denotes the size (in hours) of the training set used.

signals [24, 25]. Waveform-based models, such as Demucs (DMC) [21], process the raw waveform of an input mixture and predict the waveforms of the individual separated sources. Hybrid models integrate complementary information from waveform- and spectrogram-based models, encompassing both spectral and temporal branches. In these architectures, latent representations are combined through the addition of shared layers to leverage the advantages offered by both domains [4, 26, 27]. Examples include the hybrid Demucs model (HDMC) introduced in [4] and a system (AudioShake) provided by the company AudioShake.

In this paper, we consider the challenging source separation scenario of decomposing piano concerto recordings into distinct piano and orchestral tracks. Piano concertos involve an intricate interplay between the piano and the entire orchestra, resulting in high spectro-temporal correlations among the constituent instruments. Additionally, the absence of multitrack data for training poses an extra challenge for data-driven source separation approaches. To overcome the lack of training data, the approach in [28] proposes generating artificial training data by superimposing randomly chosen audio patches from the solo piano repertoire (e.g., piano sonatas and etudes) and orchestral pieces without piano (e.g., symphonies). The training procedure and comparison of four different models mentioned above are described in [28], including the use of further data augmentation techniques. In our experiments, we employ four pre-trained models from the study [28], shown in Table 1. Additionally, we utilize the commercial system AudioShake, trained with over 500 hours of multitrack music recordings spanning various genres, with a focus on popular music. It is important to note that the AudioShake system has not been specifically adapted to the piano concerto scenario but is trained on mixtures where the vocal stem is usually dominant.

Finally, we want to emphasize that the implementation details and the reproducibility of the various MSS systems are not the main focus of this paper. Instead, these MSS systems and the piano concerto scenario serve as a framework for illustrating our evaluation methodology, as we will further discuss in Section 4.

3. EVALUATION APPROACH

We now introduce our novel evaluation approach, which we will apply to compare reference piano recordings and separated piano tracks. In Section 3.1, we briefly describe the PCD collection, which will serve as a test dataset, and present our score-based extensions. Then, in Section 3.2,

Room ID	Room Description	Piano	Dur	#Notes
R1	Lecture hall	Yamaha C3	180	1780
R2	Private studio	Yamaha C3X	180	2216
R3	Small concert hall	Seiler	252	2305
R4	Big concert hall	Steinway D	360	3741
Σ			972	10042

Table 2: Overview of the PCD test set, indicating the four rooms and the piano models employed, and including the duration (in seconds) and the number of notes (piano only).

we revisit the score-informed NMF approach for audio decomposition. Finally, in Section 3.3, we define the SDR-based evaluation metrics, which we use to gain a deeper understanding of the source separation results.

3.1 Piano Concerto Dataset and its Extension

The PCD collection, introduced in [18], is based on piano concerto recordings featuring five different amateur and professional pianists playing along with orchestral recordings provided by the publisher *Music Minus One*¹. Multitrack recordings with clean piano and orchestra reference tracks were produced from these sessions. The PCD consists of 81 multitrack excerpts, each lasting 12 seconds, selected from 15 piano concertos spanning the Baroque to Post-Romantic period. As summarized in Table 2, the PCD comprises excerpts recorded in four distinct acoustic settings with different grand piano models.

Our novel evaluation approach relies on synchronized score information used for notewise audio decomposition. To this end, we manually generated symbolically encoded sheet music representations using the *Sibelius* software² for the piano tracks (and piano-reduced versions of the orchestra tracks, which are not utilized in this study). We employed the *Sync Toolbox* [20]³ to automatically align the score information with the PCD audio excerpts. To ensure high synchronization accuracy, we computed these alignments in two independent ways: once based on the piano-only tracks and another time based on the piano–orchestra mixes. We then applied fusion techniques to establish the final score annotations. Additionally, expert listeners verified the final results using visual cues provided by the *Sonic Visualizer* [29] and acoustic cues using sonified score annotations overlaid with the audio excerpts. With regard to note onsets, the accuracy of the score annotations for the piano tracks can be expected to lie in the range of 20–40 ms. Additionally, we manually annotated the left-hand (LH) and right-hand (RH) notes, resulting in further musically meaningful note groupings beyond the notewise ones.

We release the symbolically encoded sheet music along with the score-based annotations of the audio excerpts, thereby adding an additional score-based layer to the PCD collection as part of the contributions of this paper.⁴

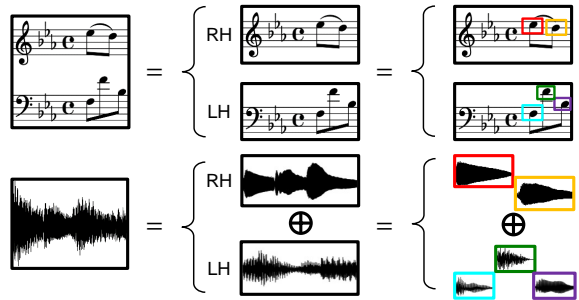


Figure 2: Illustration of the decomposition of the piano track into left-hand (LH), right-hand (RH), and individual note events as indicated by the rectangular windows.

3.2 NMF-Based Audio Decomposition

Nonnegative matrix factorization (NMF) is an algorithm for approximating a nonnegative matrix as the product of two low-ranked nonnegative matrices [30]. In the context of music processing, NMF has been widely applied to decompose a magnitude spectrogram into the product of two nonnegative matrices [31], where the columns of the first matrix encode spectral prototype patterns (called *templates*), and the rows of the second matrix encode their occurrences in time (called *activations*). Thanks to nonnegativity and multiplicative update rules, NMF facilitates the straightforward integration of prior musical knowledge, such as information from an acoustic model or a musical score. For instance, one may constrain the spectral template matrix to enforce a harmonic structure [32] or use aligned score information to constrain the activation matrix [16]. In addition to stabilizing the convergence of the NMF algorithm, such constraints also guide the factorization process to yield decompositions of musical relevance [33].

Following the approach in [34], we adopt a score-informed NMF approach to decompose a given audio signal x into its constituent notewise audio events x^m for $m \in [1:M]$ and a residual signal r such that

$$x = \sum_{m=1}^M x^m + r. \quad (1)$$

Here, we assume that we have a score representation with M denoting the number of note events, which are aligned to the audio signal. Note that this alignment does not need to be completely accurate, as it only serves to constrain the NMF algorithm, which can then improve the accuracy in the iteratively learned decomposition process. Besides applying this procedure to obtain a notewise decomposition of the audio signal, one can use the same approach to obtain a decomposition corresponding to note groups, resulting, for example, in the decomposition of the LH and RH notes, as illustrated in Figure 2.

We conclude our description of the NMF-based decomposition approach with some final remarks regarding implementation issues encountered in our experiments based on the PCD test set. Note that, in general, NMF training based on iterative update rules yields more reliable decom-

¹ www.halleonard.com/series/MMONE

² www.sibelius.com/

³ [www.github.com/meinardmueller/synctoolbox](https://github.com/meinardmueller/synctoolbox)

⁴ www.audiolabs-erlangen.de/resources/MIR/PCD

position results when applied to longer input spectrograms exhibiting a coherent template structure. Therefore, rather than applying the NMF-based decomposition to individual 12-second excerpts, we concatenated all 12-second excerpts recorded in the same room (see Table 2). This strategy is grounded on the assumption that the learned spectral templates, encoding characteristics of the piano and room acoustics, exhibit coherence within each room. Subsequently, we executed the NMF algorithm for 100 iterations on the concatenated data for four subsets with distinct room acoustics, using the same configurations and initialization approach introduced in [16]. This procedure was applied to both the reference piano recordings and the separated piano tracks generated by each MSS model. The resulting notewise decomposition results serve as the basis for our experiments, as reported in Section 4.

3.3 SDR-Based Metrics

The signal-to-distortion ratio (SDR) is a widely used metric in the evaluation of source separation performance, measuring the quality of a separated source by comparing it to the reference source in terms of signal distortion [13]. In our evaluation, when given a reference signal x and a separated signal \hat{x} , we use instead the more computationally efficient SDR metric proposed at the recent SDX challenge [35], also denoted as SDR:

$$\text{SDR}(x, \hat{x}) := 10 \log_{10} \frac{\|x\|^2}{\|\hat{x} - x\|^2}. \quad (2)$$

Rather than comparing entire excerpts, we use a localized variant referred to as $\text{SDR}_{\text{local}}$ that better accounts for significant level differences within the signal. To this end, we split the reference and separated signals into 1-second segments x_k and \hat{x}_k , respectively, defining:

$$\text{SDR}_{\text{local}} := \frac{1}{K} \sum_{k=1}^K \text{SDR}(x_k, \hat{x}_k) \quad (3)$$

In our evaluation, we have $K = 12$, as each excerpt in the PCD test set has a duration of 12 seconds.

To obtain a musically more informed evaluation metric, we exploit the decomposition as defined in Equation (1) and consider notewise SDR values:

$$\text{SDR}_{\text{note}} := \text{SDR}(x^m, \hat{x}^m), \quad (4)$$

where x^m and \hat{x}^m denote the notewise sound events of the reference signal and the separated signal, respectively. Note that, using the same score-based activation constraints in the NMF decomposition for x and \hat{x} , respectively, the lengths of x^m and \hat{x}^m are identical for a given $m \in [1:M]$.

4. EXPERIMENTS

In this section, we report on our systematically conducted experiments to highlight the potential of our notewise evaluation methodology. In this context, the piano concerto separation task, along with the five MSS systems described

Model	Piano	Orchestra
UMX	8.38 ± 4.24	3.61 ± 2.19
SPL	8.16 ± 3.99	3.46 ± 2.25
DMC	7.59 ± 4.38	2.82 ± 2.13
HDMC	9.61 ± 4.42	4.75 ± 2.31
AudioShake	12.82 ± 4.24	8.01 ± 2.97

Table 3: $\text{SDR}_{\text{local}}$ values (mean and standard deviation) averaged over all PCD excerpts for different MSS systems (see Table 1).

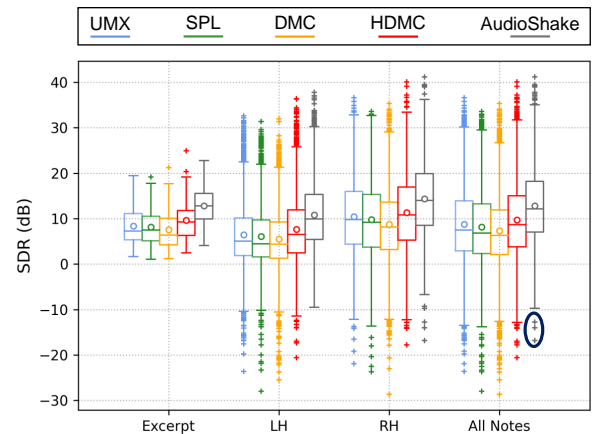


Figure 3: Comparison of different evaluation methodologies for the piano case using boxplots. The three outliers for AudioShake, indicated by the black oval, are shown in Figure 8.

in Section 2, should be considered an illustrative case study of practical relevance. When describing the various experiments, we progress from a coarse to a fine perspective. We start with a more global view of the source separation quality of the MSS systems (Section 4.1). Subsequently, we adopt a more fine-grained perspective, delving into the separation quality depending on the musical pitch (Section 4.2). Finally, we assume an excerptwise view and discuss specific examples to illustrate how separation errors may occur in musically complex situations (Section 4.3). This hierarchical discussion underscores how the notewise evaluation methodology serves as a tool, enabling users to delve into and comprehend not only the separation results but also the intricacies within the underlying music.

4.1 Global Perspective

To gain an initial understanding of the overall performance of the five MSS systems, Table 3 presents the $\text{SDR}_{\text{local}}$ values averaged across the 81 PCD excerpts for both separated piano tracks and orchestra tracks. For instance, in the piano case, DMC achieves the lowest $\text{SDR}_{\text{local}}$ value at 7.59, while HDMC shows a higher value of 9.61, and AudioShake outperforms all other models with a value of 12.82. Similar trends are evident in the separated orchestra case, although all values are notably lower compared to the piano case. Similar tendencies have been reported in [28].

In the subsequent finer-grained evaluation, we employ notewise evaluation metrics. Since we have the required

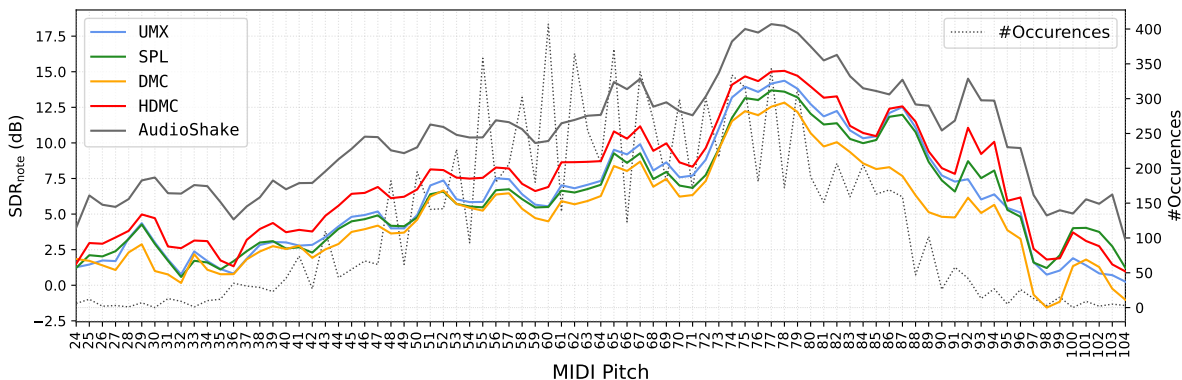


Figure 4: SDR_{note} values aggregated by pitch (specified by MIDI note number) shown for five MSS systems.

symbolic score information for the score-based NMF decomposition exclusively for the piano tracks, we confine our analysis to the piano case.⁵ Extending the evaluation methodology for the five MSS systems, Figure 3 shows boxplots that indicate the median, first quartile, third quartile, and outliers of differently computed SDR values. The first group of boxplots (Excerpt) provides the SDR_{local} values computed as in Table 3. The second (LH) and third (RH) groups show the SDR_{note} values for the left-hand and right-hand notes, respectively, and the last group (All Notes) shows the SDR_{note} values for all individual notes.

While the general trends for the five MSS systems are similar to those shown in Table 3, the different evaluation methodologies provide additional information. Firstly, being based on notewise aggregation, outliers in the SDR_{note} -based boxplots offer explicit cues worth further investigation. For instance, outliers such as the three indicated by the black oval in Figure 3 yield interesting examples for musically complex passages as further explored in Section 4.3. The boxplots in Figure 3 also facilitate a comparison of SDR_{note} values between the LH and RH notes. Notably, for all MSS systems, a better separation quality can be observed for the right hand compared to the left hand, with a difference of approximately 5 dB. Drawing from these observations, one can formulate various hypotheses regarding the relationship between source separation quality and pitch or musical complexity, as we detail in the subsequent sections. Please visit our demo webpage to find audio examples separated by five MSS models.⁶

4.2 Pitchwise Evaluation

Considering that RH typically contains higher notes than LH, one may conjecture that source separation quality depends on the pitch of the played notes. To test this hypothesis, Figure 4 provides an overview of the SDR_{note} values aggregated by pitch (specified by MIDI note number). While the overall trend regarding the MSS systems’

performances remains the same (AudioShake performing best, DMC worst, and HDMC being in between), the pitch-dependent SDR_{note} values indicate that, overall, source separation quality tends to increase for higher pitch numbers, with the highest values in the pitch range 74–80.

However, such trends, and drawing conclusions from them, need to be taken with care. For example, the curves in Figure 4 may indicate that source separation becomes more difficult for very high pitches in the range 96–104. However, these numbers lack statistical significance due to the limited occurrence (indicated by the dotted line). Also, one may assume that such pitches may rarely occur in the training material used for training the MSS systems, thus leading to poor generalizations on the test set.

4.3 Excerptwise Evaluation

Rather than source separation quality solely being a matter of pitch height, there may be other confounding factors underlying the trend. An alternative hypothesis could be that the LH (or lower-pitched) piano notes are more interwoven with the orchestral track, while the RH (or higher-pitched) piano notes stand out and can be better isolated by MSS systems. To explore aspects of musical complexity, we present in Figure 5 SDR_{note} values aggregated by excerpt (specified by PCD ID), this time focusing on the results for the two best-performing MSS systems, HDMC and AudioShake. Sorting the excerpts, e. g., based on decreasing mean values concerning AudioShake, facilitates the identification of challenging excerpts, which are depicted toward the right side of the plot.

Guided by the plot in Figure 5, let us consider some concrete examples. Examining the top three excerpts (PCD IDs 045, 042, and 024), a manual inspection reveals that these excerpts share a common characteristic of relatively low musical complexity, consisting of slower passages drawn from the second movements of piano concertos by Beethoven and Mozart. For such passages, both MSS systems achieve a good separation quality.

Next, let us examine the excerpt with the lowest SDR_{note} value. This excerpt has PCD ID 076 and corresponds to measures 18–24 of the first movement of Tchaikovsky’s Piano Concerto Op. 23, as shown in Figure 6. Evidently, this passage exhibits a high musical complexity, with both piano and orchestra playing numerous

⁵ For the orchestra, we generated only piano-reduced scores due to the considerable effort required for full scores. Additionally, automated synchronization and decomposition approaches present greater challenges for orchestral music compared to piano, extending beyond the scope of the case study presented in this paper.

⁶ www.audiolabs-erlangen.de/resources/MIR/2024-ISMIR-PianoSepEval

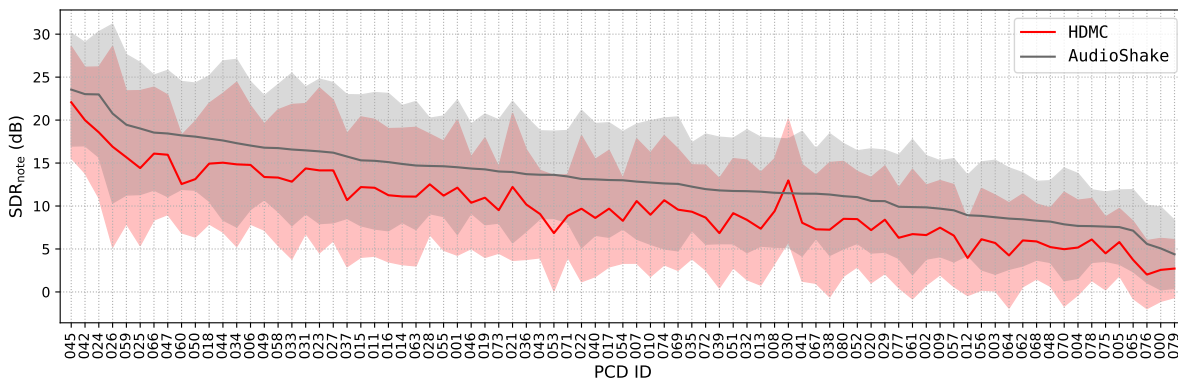


Figure 5: SDR_{note} values aggregated by excerpt (specified by PCD ID) shown for the two best-performing MSS systems, HDMC and AudioShake. The mean (solid line) and standard deviations (filled regions) are indicated. The excerpts are sorted based on decreasing mean values with regard to AudioShake.



Figure 6: Excerpt with PCD ID 079: Tchaikovsky’s Piano Concerto Op. 23, measures 18–24 of the first movement (only four measures are shown here).



Figure 7: Excerpt with PCD ID 000: Bach’s Piano Concerto BWV 1056, measures 1–8 of the first movement (only four measures are shown here).

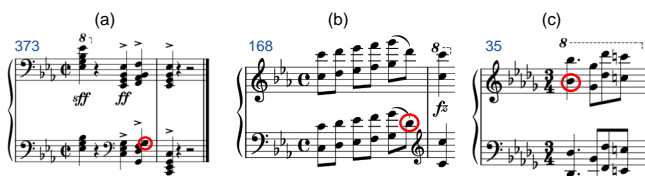


Figure 8: Musical context within the piano scores for the three notewise outliers marked in Figure 3 (here indicated by the red circles). (a) PCD ID: 052. (b) PCD ID: 061. (c) PCD ID: 077.

notes within a wide pitch range. Particularly notable are the fortissimo and broken chords in the piano part, which strongly interfere with the full orchestral sound, not to mention the effects resulting from the application of the sustain pedal. As a second concrete example, let us have a

closer look at the excerpt with PCD ID 000, also yielding a low SDR_{note} value. This excerpt corresponds to the first measures of Bach’s Piano Concerto BWV 1056, where the piano and orchestra play many notes in unison (see Figure 7). This scenario represents one of the most challenging situations for source separation models to deal with [36, 37].

Finally, we revisit the boxplots shown in Figure 3, where we marked three outliers indicating problematic notewise sound events with low SDR values, poorly separated by AudioShake. Figure 8 provides the musical context within the piano scores where these notes occur. A common feature in these examples, which is also typical in piano music in general, is the simultaneous playing of two notes that belong to the same pitch class, contributing to a rich and complex sound texture. Obviously, such instances are difficult for any MSS system, as well as the NMF algorithm to handle.

Overall, these examples show that while MSS systems like AudioShake and HDMC are capable of achieving impressive separation quality, their efficacy is highly influenced by the intrinsic characteristics of the musical pieces.

5. CONCLUSION

In this paper, we have considered a novel evaluation methodology that compares separated sounds with reference sounds on a notewise basis rather than at the excerpt level. For the challenging piano concerto scenario and employing five MSS systems, we applied this methodology in a case study focusing on the separated piano tracks. This allowed us to gain insights into the separation quality and the complexity of the underlying music. While our focus has been on the piano case, future work may involve evaluating other orchestral instruments and guitars. This could pose additional challenges not only for source separation itself but also for automated synchronization and decomposition approaches. On a meta-level, we hope that our hierarchical discussion, assuming different perspectives, also showcased the potential of musically informed evaluation methodologies, providing a basis for interdisciplinary dialogue between engineering and music experts.

Acknowledgements: This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Grant No. 328416299 (DFG MU 2686/10-2). The International Audio Laboratories Erlangen are a joint institution of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) and Fraunhofer Institute for Integrated Circuits IIS.

6. REFERENCES

- [1] E. Cano, D. FitzGerald, A. Liutkus, M. D. Plumbley, and F. Stöter, “Musical source separation: An introduction,” *IEEE Signal Processing Magazine*, vol. 36, no. 1, pp. 31–40, 2019.
- [2] F. Stöter, S. Uhlich, A. Liutkus, and Y. Mitsufuji, “Open-Unmix – A reference implementation for music source separation,” *Journal of Open Source Software*, vol. 4, no. 41, 2019. [Online]. Available: <https://doi.org/10.21105/joss.01667>
- [3] R. Hennequin, A. Khelif, F. Voituret, and M. Mousallam, “Spleeter: A fast and efficient music source separation tool with pre-trained models,” *Journal of Open Source Software (JOSS)*, vol. 5, no. 50, p. 2154, 2020. [Online]. Available: <https://doi.org/10.21105/joss.02154>
- [4] A. Défossez, “Hybrid spectrogram and waveform source separation,” in *Proceedings of the ISMIR 2021 Workshop on Music Source Separation*, Online, 2021.
- [5] Y. Luo and J. Yu, “Music source separation with Band-Split RNN,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 1893–1901, 2023.
- [6] S. Sarkar, E. Benetos, and M. Sandler, “EnsembleSet: A new high quality dataset for chamber ensemble separation,” in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Bengaluru, India, 2022, pp. 625–632.
- [7] S. Sarkar, L. Thorpe, E. Benetos, and M. Sandler, “Leveraging synthetic data for improving chamber ensemble separation,” in *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2023, pp. 1–5.
- [8] S. Uhlich, M. Porcu, F. Giron, M. Enenkl, T. Kemp, N. Takahashi, and Y. Mitsufuji, “Improving music source separation based on deep neural networks through data augmentation and network blending,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, New Orleans, Louisiana, USA, March 2017, pp. 261–265.
- [9] H. Kim, J. Park, T. Kwon, D. Jeong, and J. Nam, “A study of audio mixing methods for piano transcription in violin-piano ensembles,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Rhodes Island, Greece, 2023, pp. 1–5.
- [10] Y. Özer and M. Müller, “Source separation of piano concertos with test-time adaptation,” in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Bengaluru, India, 2022, pp. 493–500.
- [11] F. Stöter, A. Liutkus, and N. Ito, “The 2018 Signal Separation Evaluation Campaign,” in *Proceedings of the International Conference on Latent Variable Analysis and Signal Separation (LVA/ICA)*, ser. Lecture Notes in Computer Science, vol. 10891. Springer, 2018, pp. 293–305.
- [12] Y. Mitsufuji, G. Fabbro, S. Uhlich, F.-R. Stöter, A. Défossez, M. Kim, W. Choi, C.-Y. Yu, and K.-W. Cheuk, “Music demixing challenge 2021,” *Frontiers in Signal Processing*, vol. 1, 2022.
- [13] E. Vincent, R. Gribonval, and C. Févotte, “Performance measurement in blind audio source separation,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 4, pp. 1462–1469, 2006.
- [14] E. Cano, D. FitzGerald, and K. Brandenburg, “Evaluation of quality of sound source separation algorithms: Human perception vs quantitative metrics,” in *Proceedings of the European Signal Processing Conference (EUSIPCO)*, 2016, pp. 1758–1762.
- [15] M. Torcoli, T. Kastner, and J. Herre, “Objective measures of perceptual audio quality reviewed: An evaluation of their application domain dependence,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 1530–1541, 2021.
- [16] S. Ewert and M. Müller, “Using score-informed constraints for NMF-based source separation,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Kyoto, Japan, 2012, pp. 129–132.
- [17] Z. Chen, T. Yoshioka, L. Lu, T. Zhou, Z. Meng, Y. Luo, J. Wu, X. Xiao, and J. Li, “Continuous speech separation: Dataset and analysis,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2020, pp. 7284–7288.
- [18] Y. Özer, S. Schwär, V. Arifi-Müller, J. Lawrence, E. Sen, and M. Müller, “Piano Concerto Dataset (PCD): A multitrack dataset of piano concertos,” *Transactions of the International Society for Music Information Retrieval (TISMIR)*, vol. 6, no. 1, pp. 75–88, 2023.
- [19] S. Ewert, M. Müller, and P. Grosche, “High resolution audio synchronization using chroma onset features,” in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Taipei, Taiwan, 2009, pp. 1869–1872.

- [20] M. Müller, Y. Özer, M. Krause, T. Prätzlich, and J. Driedger, “Sync Toolbox: A Python package for efficient, robust, and accurate music synchronization,” *Journal of Open Source Software (JOSS)*, vol. 6, no. 64, pp. 3434:1–4, 2021.
- [21] A. Défossez, N. Usunier, L. Bottou, and F. R. Bach, “Music source separation in the waveform domain,” 2019. [Online]. Available: <http://arxiv.org/abs/1911.13254>
- [22] Y. Luo and N. Mesgarani, “Conv-TasNet: Surpassing ideal time–frequency magnitude masking for speech separation,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 8, pp. 1256–1266, 2019.
- [23] D. Stoller, S. Ewert, and S. Dixon, “Wave-U-net: A multi-scale neural network for end-to-end audio source separation,” in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Paris, France, 2018, pp. 334–340.
- [24] N. Q. K. Duong, E. Vincent, and R. Gribonval, “Underdetermined reverberant audio source separation using a full-rank spatial covariance model,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 7, pp. 1830–1840, 2010.
- [25] A. Liutkus and R. Badeau, “Generalized Wiener filtering with fractional power spectrograms,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brisbane, Australia, April 2015, pp. 266–270.
- [26] M. Kim, W. Choi, J. Chung, D. Lee, and S. Jung, “KUIELab-MDX-Net: A two-stream neural network for music demixing,” in *Proceedings of the ISMIR 2021 Workshop on Music Source Separation*, Online, 2021.
- [27] S. Rouard, F. Massa, and A. Défossez, “Hybrid transformers for music source separation,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Rhodes Island, Greece, 2023.
- [28] Y. Özer and M. Müller, “Source separation of piano concertos using musically-motivated augmentation techniques,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP)*, vol. 32, pp. 1214–1225, 2024.
- [29] C. Cannam, C. Landone, and M. B. Sandler, “Sonic Visualiser: An open source application for viewing, analysing, and annotating music audio files,” in *Proceedings of the International Conference on Multimedia*, Florence, Italy, 2010, pp. 1467–1468.
- [30] D. D. Lee and H. S. Seung, “Algorithms for non-negative matrix factorization,” in *Proceedings of the Neural Information Processing Systems (NIPS)*, Denver, Colorado, USA, November 2000, pp. 556–562.
- [31] P. Smaragdis and J. C. Brown, “Non-negative matrix factorization for polyphonic music transcription,” in *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2003, pp. 177–180.
- [32] S. A. Raczynski, N. Ono, and S. Sagayama, “Multi-pitch analysis with harmonic nonnegative matrix approximation,” in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Vienna, Austria, September 2007, pp. 381–386.
- [33] S. Ewert, B. Pardo, M. Müller, and M. Plumbley, “Score-informed source separation for musical audio recordings: An overview,” *IEEE Signal Processing Magazine*, vol. 31, no. 3, pp. 116–124, April 2014.
- [34] J. Driedger, H. Grohganz, T. Prätzlich, S. Ewert, and M. Müller, “Score-informed audio decomposition and applications,” in *Proceedings of the ACM International Conference on Multimedia (ACM-MM)*, Barcelona, Spain, 2013, pp. 541–544.
- [35] G. Fabbro, S. Uhlich, C.-H. Lai, W. Choi, M. Martínez-Ramírez, W. Liao, I. Gadelha, G. Ramos, E. Hsu, H. Rodrigues, F. Stöter, A. Défossez, Y. Luo, J. Yu, D. Chakraborty, S. Mohanty, R. Solovyev, A. Stempkovskiy, T. Habruseva, N. Goswami, T. Harada, M. Kim, J. H. Lee, Y. Dong, X. Zhang, J. Liu, and Y. Mitsufuji, “The sound demixing challenge 2023 – music demixing track,” *arXiv*, 2024.
- [36] J. J. Burred, “From sparse models to timbre learning: New methods for musical source separation,” Ph.D. dissertation, Technische Universität Berlin, Berlin, Germany, 2009.
- [37] C.-B. Jeon, H. Moon, K. Choi, B. S. Chon, and K. Lee, “Medleyvox: An evaluation dataset for multiple singing voices separation,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2023, pp. 1–5.

AUTOMATIC ESTIMATION OF SINGING VOICE MUSICAL DYNAMICS

Jyoti Narang^{b*}

Nazif Can Tamer^{b*}

Viviana de la Vega[‡]

Xavier Serra^b

^b Music Technology Group, Universitat Pompeu Fabra, Barcelona, Spain

[‡] Escuela Superior de Música de Cataluña (ESMUC), Barcelona, Spain

jyoti.narang@upf.edu, nazifcan.tamer@upf.edu,
vivianadelavega@gmail.com, xavier.serra@upf.edu

ABSTRACT

Musical dynamics form a core part of expressive singing voice performances. However, automatic analysis of musical dynamics for singing voice has received limited attention partly due to the scarcity of suitable datasets and a lack of clear evaluation frameworks. To address this challenge, we propose a methodology for dataset curation. Employing the proposed methodology, we compile a dataset comprising 509 musical dynamics annotated singing voice performances, aligned with 163 score files, leveraging state-of-the-art source separation and alignment techniques. The scores are sourced from the OpenScore Lieder corpus of romantic-era compositions, widely known for its wealth of expressive annotations. Utilizing the curated dataset, we train a multi-head attention based CNN model with varying window sizes to evaluate the effectiveness of estimating musical dynamics. We explored two distinct perceptually motivated input representations for the model training: log-Mel spectrum and bark-scale based features. For testing, we manually curate another dataset of 25 musical dynamics annotated performances in collaboration with a professional vocalist. We conclude through our experiments that bark-scale based features outperform log-Mel-features for the task of singing voice dynamics prediction. The dataset along with the code is shared publicly for further research on the topic.

1. INTRODUCTION

Musical dynamics, such as *piano* and *forte* [1], are key elements in adding expressiveness to the singing voice [2]. They enhance overall performance and facilitate the conveyance of the desired emotional impact [3]. Despite extensive research on the singing voice, the analysis of dynamics in this context has received limited attention for several reasons. Firstly, annotating dynamics is an expensive process that requires repeated listening to audio tracks

to accurately identify the dynamics category. Secondly, unlike other musical features such as pitch or tempo, the categorization of dynamics is not clearly defined, and even the same annotator may interpret a piece differently on multiple listens. Finally, a significant challenge for modern deep learning applications is the lack of reliable, existing dynamics based annotated datasets that can be used for the development of automatic analysis systems [4].

Despite the challenges of dynamics-based annotations for the singing voice, investigating dynamics in singing performances is worthwhile. On one hand, dynamics are a key component of expressivity in a music performance [5, 6]. On the other hand, dynamics are also an integral part of the music writing tradition [1, 7]. The use of dynamics in Western classical music evolved significantly from the Baroque period to the Romantic era. Particularly during the Romantic era, when expressivity became prominent, the annotation of dynamics alongside the score became widespread and accepted as part of the composition process. Composers frequently utilized symbols such as *forte*, *piano*, *crescendo*, and *diminuendo* to convey their desired variations in musical dynamics, and adhering to the dynamics instructions given by the composers became an important part of a Classical music performance.

While dynamics is a musical concept, its automatic estimation for music performance analysis relies on properties derived from audio signals. The audio characteristic most similar to musical dynamics is loudness or perceptual intensity. However, the mapping of musical dynamics to audio-based features from Music Information Retrieval (MIR) technologies is still not clearly understood. Extensive research exists on dynamics and tempo as expressive dimensions for Western classical piano performances [8]. However, unlike piano, there are almost no publicly available dynamics-based annotated datasets for the singing voice, which hinders the development of such technologies for the vocal performance analysis.

In this work, we propose to take advantage of the existing OpenScore Lieder corpus to curate a dataset of vocal performances with dynamics annotations, using state-of-the-art source separation and alignment as intermediate steps¹. Furthermore, we curate a dataset of 25 other performances of different genres annotated manually by a professional Classical vocalist to test the model. At the end,

*These authors contributed equally to this work.



¹ <https://github.com/MTG/SingWithExpressions>.
git

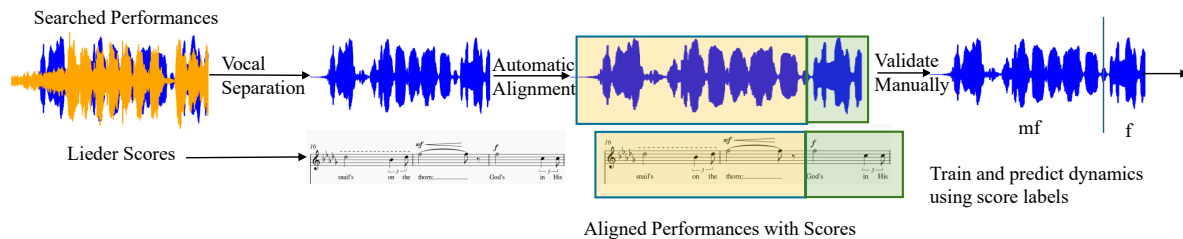


Figure 1: Data Preparation Pipeline: Corresponding to the Lieder scores from OpenScore Lieder Corpus, we apply Vocal Separation followed by Automatic Alignment. Finally, we validate the aligned score-performance data using Visualizations

we study the relationship between score based musical dynamics to perceptually motivated audio features [9] like log-Mel and bark-scale based features, testing the model with different analysis window-size, and genres of the test dataset.

Figure 1 illustrates the overall pipeline of the task. Using the meta-data information of the repository accompanying Lieder corpus, we start with searching for corresponding performances on YouTube. Further, we apply vocal separation on the performance to get vocals. Thereafter, using state-of-art alignment techniques, we align the corresponding score with the performance. At this stage, to test the accuracy of the alignment process, we develop visualization to filter out performances with mismatched aligned scores. Using the aligned score and performance data, we train a model for estimating dynamics based markings for an unknown performance.

The rest of this paper is structured as follows. In section 2, we cover the related works. Section 3 describes the dataset and the curation process. In section 4, we describe the experiments conducted with the curated data, followed by discussion and future work.

2. RELATED WORK

Although musical dynamics has been a topic of investigation in several studies [6, 7, 10, 11], especially for the case of piano [8, 12–14] there remains a notable gap in research concerning standalone musical dynamics analysis for the case of singing voice, particularly from an MIR perspective. Despite this gap, dynamics form a fundamental aspect of analysis within the interconnected fields of singing voice synthesis [15] and voice pedagogy [5].

In Singing Voice Synthesis (SVS) systems, dynamics play a crucial role in conveying expressive nuances [16]. Typically, dynamics are modelled as measures of energy in the signal [15–17] at the frame level. However, while there exists a close correlation between energy of the signal and musical dynamics, the influence of other parameters, such as pitch and timbre [10], remains largely unexplored. Understanding the relationship between pitch, timbre and dynamics could lead to more realistic representations of musical expression in SVS systems.

Bous and Roebel [4] explore the relationship between musical dynamics and timbral characteristics of the singing

voice, employing mel-spectrogram features. Their experiment involves modifying the singing voice dynamics using a neural auto-encoder to transform voice levels. Effectiveness is assessed through evaluating perceived changes in voice level in the transformed recordings. However, a significant challenge arises as there is currently no reliable labels to determine the perceived changes in musical dynamics corresponding to "voice-level" changes as proposed in the system.

Narang et al. [18] utilize perceptually-motivated *some* scale, comparing loudness curves of different professional renditions and student renditions for "musical dynamics" comparison following the methodology outlined by Kosta et al. [12] for comparing musical dynamics in piano. However, the study encountered limitations due to the lack of dynamics annotated datasets for evaluation.

While there are some aspects of the research on Vocal Pedagogy [5] that has been utilized for the case of singing voice research from an MIR perspective, for example, Phonation mode [19] dataset or VocalSet [20] (which also contains some singing voice dynamics annotations but confined to vowel renditions), research outcomes of the vocal pedagogy remain largely unexplored by the MIR community. One direction is the role of voice source in singing voice, or how the positioning of the diaphragm affects vocal characteristics [21]. A study on vocal dynamics can help infer the voice source characteristics that can directly aid in vocal pedagogy.

3. DATASET

Dynamics are considered to be the most commonly manipulated parameter of an expressive performance and research investigations show that professionals or experts have much better control in expressive parameters in comparison to novice performers [6]. Further, songs from the 19th century Romantic era of Western classical music are widely known to be rich in expressive parameters. Drawing inspiration from this notion, we curate a dataset comprising professional renditions of 19th-century songs sourced from the OpenScore Lieder corpus [22]. Notably, composers often embed numerous dynamic markings within their scores, laying a foundational framework conducive to the analysis of dynamics.

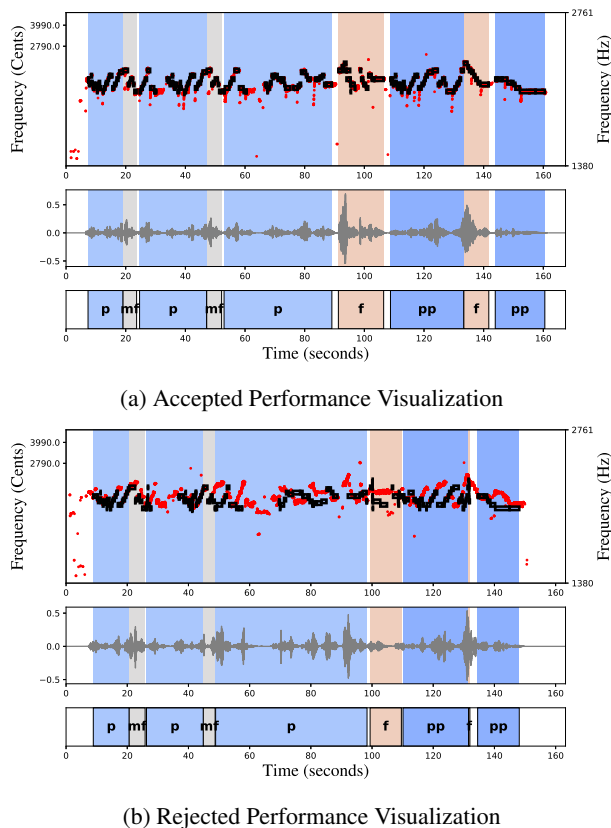


Figure 2: Example visualization after automatic alignment on "The Shepherds Song" by Edward Elgar; For each sub-figure: red dots represent f_0 using crepe, black dots represent note-information from the score (top), audio waveform (middle), dynamics information from the aligned score after automatic alignment (bottom)

3.1 Training Dataset Curation Process

3.1.1 Score Sources

Lieder Scores is a comprehensive collection of over 1200 19th century songs encoded over several years [22]. Within the Lieder dataset, we capitalize on two specific resources to facilitate our data curation process:

- The GitHub repository of Lieder provides MSCX files along with batch-conversion script to convert to MusicXML, enabling further processing with tools such as music21 [23]
- In the metadata section of the Lieder scores, a comprehensive compilation of composers, score names, and their respective MuseScore IDs is provided. This rich metadata serves as a valuable resource during the performance collection stage, enabling efficient querying and selection of performances.

3.1.2 Filtering Criteria for Scores

From all the batch-converted MusicXML files, we filter all scores, focusing on those with more than 3 dynamics annotations, and containing only 3 streams of score data: vocal, piano left hand and piano right hand.

3.1.3 Performance Sources

For the identified scores with greater than 3 dynamics markings, we search for multiple corresponding performances on YouTube using the query term obtained from the meta-data information of the scores. We curate multiple performances of similar pieces with the intention of extracting general dynamics based expressive patterns from professional singers. Our aim is to glean insights into varied interpretations, as there is no singular correct rendition of a performance that strictly adheres to the score. Subsequently, we carefully listen to each performance, specifically selecting those featuring vocals accompanied solely by piano. It is to be noted that not all composers have available performance data; thus, our selection process initially prioritizes renowned figures such as Schubert, Schumann, Brahms or Debussy, and ones with greater than 10 dynamics annotations. Once having exhaustively searched for performances of these well known composers, we proceed to search for lesser known composers following similar criteria. We automate the download process by utilizing YouTubeDL batch download to acquire the identified performances. The method yields a final list of 970 performances comprising identified composers, performances, and their respective MusicXML score files with dynamics based annotations.

3.1.4 Filtering Criteria for Performances

Following the filtration of scores and the manual curation of performance links, we advance to filtering performances suitable for the dynamics learning process. This process includes the following steps:

Source Separation Singing voices typically aren't presented in isolation. Even for solo performances, piano accompaniment is part of the performance. However, for our analysis, we require solo vocal renditions to accurately discern variations in performance dynamics. The initial step involves isolating the vocal component from the vocal-piano mix. This process, known as source separation, entails breaking a mixture into its constituent components, and significant research has been dedicated to separation of vocals from the mix. We use Demucs v2 [24] to extract the vocals for the chosen songs. The robustness of using vocals resulting from source separation as an intermediate step was examined with the MusDB dataset [25].

Automatic Alignment To ensure that our curated performances can effectively serve as the basis for dynamics analysis, it's essential to achieve a basic alignment with the scores. Our approach to label creation draws inspiration from the methodology outlined by Tamer et al. [26, 27], who leverage Dynamic Time Warping (DTW) based music synchronization techniques [28] for creating pseudo labels in the realm of Violin transcription. Additionally, the concept of utilizing audio-to-score alignment as a pre-processing step for curating datasets in a semi-automatic manner for musicological endeavours was introduced in works by Weiss et al. [29], with a focus on the curation of Schubert's Winterreise dataset. While the works by Weiss et al. utilize MIDI-to-score alignment, we have chosen to

conduct the alignment using musicXML scores. This decision stems from the fact that dynamics information such as *piano*, *forte*, *crescendo*, and *diminuendo* can be less reliable in the process of MIDI conversion.

Manual Filtering using Visualizations The alignment stage yields a score with time information mapped to the corresponding performance files. Subsequently, we develop a visualization process utilizing fundamental frequency (f0) data extracted from performance files using CREPE [30] to validate the alignment between time-aligned performance and score files. Figure 2 showcases a sample visualization from the dataset. Figure 2a illustrates a performance that was accepted, and Figure 2b depicts a performance that we manually excluded during the selection process. The performance in Figure 2b was rejected because f0 curve from crepe (red dots) do not align with note-information from score (black rectangles) after automatic alignment, and hence the final labels lose reliability. The end result of this step is a comprehensive dataset of 509 performances for 163 aligned score files, which can be used to extract precise note-level expressive information from the score using tools like music21 [23].

3.1.5 Dynamics based Labels Extraction from Aligned Score Files

The aligned score files consist of all score-based information crucial for dynamics prediction. In this stage, we process the musical dynamics labels extracted using music21. Our approach adheres to the following principle: consecutive notes in the aligned audio are assumed to maintain similar dynamics unless there is a change in dynamics annotation in the score. When encountering labels like *sfz* or *sf* for a note, the value of the label of the consecutive note is assigned to be the dynamic value of the note preceding *sf* or related categories. This process results in a note-level mapping of 13 musical dynamics categories: *pppp*, *ppp*, *pp*, *p*, *mp*, *mf*, *f*, *ff*, *fff*, *ffff*, *sf*, *crescendo*, *diminuendo* directly extracted from the score. It is to be noted that we consolidate accent related categories, such as *sf*, *sfz* into a single category. Additionally, while we focus on musical dynamics for our task, the aligned score-performance data holds potential for various other Music Information Retrieval (MIR) tasks related to singing voice, including transcription, synthesis, or pedagogy.

3.2 Test Dataset Curation Process

For testing, we curated performances from a diverse selection of genres, ranging from operatic pop to theatre, R&B, or jazz, which lie outside the typical classical music domain. We collaborated with a Classical Vocalist, possessing over a decade of experience, to identify artists renowned for their wide vocal range. Once identified, we created reference scores for selected performances by these professional artists. The distribution of the genres in the selected pieces is as follows: pop(13), rock(12), jazz(3), soul(5), R&B(5), theatre(2) and other miscellaneous genres(5) including categories such as "post-disco", "acoustic" or "progressive rock", amongst others.

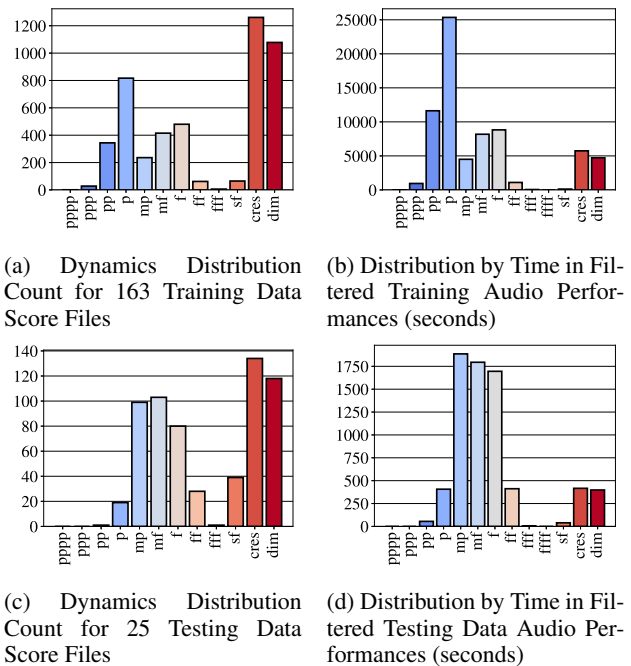


Figure 3: Dynamics Distribution across Train and Test Performances

3.2.1 Annotation Methodology

This section details the annotation methodology for dynamics-related markings of selected pieces as outlined by the musician: In the first listening, the piece's starting dynamic value is determined according to the dynamic markings such as: *pp*, *p*, *mp*, *mf*, *f*, *ff*, creating a reference point for each piece. This phase captures the most prominent features, recognizing that notation conveys more than mere amplitude. Subsequent listenings entail adding details, both in terms of dynamics and articulation of the text and musical phrases. Increased attention reveals additional layers of variation, often unnoticed during the first listening. In the third listening, decisions are made based on unification criteria. If different notations were used for the same musical effect in similar portions of the piece (e.g., different verses), the notation that best represents the musical intent is selected and unified with the rest. Rarely, genuine differences may exist between similar sections, in which case they are left distinct. In the final listening, no further notations are added. Instead, a mental musical reading of the entire work, from beginning to end, is undertaken. This involves elaborating on the interpretation following the written notations while simultaneously comparing it with the rendition produced by the artist.

3.2.2 Processing Methodology for Test Dataset

The processing methodology followed for the test dataset is similar to that of the training dataset, i.e., we apply source separation followed by automatic alignment to fetch the annotated labels using curated reference scores and performances.

Table 1: Results with Mel and Bark Features. Temporal resolution refers to the final feature rate after downsampling.

Seq Length	Temporal Resolution	Perceptual Feature	Acc	Acc(± 1)	Acc(± 2)
4096	17.4 ms	log-Mel	6.95	38.46	63.02
10000	29 ms	log-Mel	11.35	42.55	68.38
4096	16 ms	Bark	20.44	59.17	82.24
10000	30 ms	Bark	20.96	60.71	84.78

3.3 Dataset Statistics

Audio Statistics: The total duration of all the performances for the training dataset is 25.91 hours. The total duration of test files is 1.614 hours. The distribution of the labels as identified in dataset section 3 is illustrated in Figure 3. We observe that Lieder scores follow a relatively uniform distribution of dynamics with large number of dynamics annotations centered on a ‘*piano*’. And for the test dataset, the distribution curve is largely gaussian with majority of the distribution centered around *mp* and *mf*, which is not surprising considering the nature of pop music and mixing and mastering effects added to the final renditions.

Performance Count Per Piece: Although a single performer can deviate from the annotated score dynamics, having multiple performers per piece can help the model learn the general patterns closer to composer’s intention. To leverage this effect, we collect performances with an average count of 3.12 performances per piece (std: 2.13), with a maximum of 12 performances for a piece by Robert Schumann. The average performance duration was observed to be 9.54 minutes (std: 9.36 minutes), with a maximum of 74.27 minutes for a piece by Franz Schubert and a minimum of 1.01 minutes for a piece by Peter Warlock.

4. EXPERIMENTS AND RESULTS

For the experiments outlined in this section, we utilize the curated dataset of Classical vocal performances for training and the dataset created in collaboration with the Classical vocalist for testing. We convert the note-level dynamics labels spanning from *pianissississimo* (*pppp*) to *fortissississimo* (*ffff*) into framewise labels encompassing 10 dynamics classes, and train and test our models for estimating the frame-wise dynamics. Thus, we consider dynamics estimation as a 10-class classification problem operating at the granularity of individual frames.

Input Representations: For model inputs, we consider two perceptually-motivated loudness features that are extracted after isolating the vocal tracks using DemucsV2 [24]. As our first input representation, we consider log-Mel features, which are commonly used in many audio and music processing tasks. These features are extracted using the librosa [31] library from audio sampled at 44.1 kHz using a hop size of 5.8 ms. As our second representation, we consider the specific loudness in Bark critical bands, which was previously studied in the context of piano dynamics [12] and singing voice loudness analysis [18]. The 240 dimensional Bark features are extracted using the MoSQUITo library [32, 33], following the Zwicker

loudness calculation method for time-varying signals [34] as specified in the ISO.532-1:2017 standard. The extraction process adheres to the default settings of a 48 kHz audio sampling rate and a 2 ms hop size.

Alongside these different input representations, we also study the effect of input sequence length and rate. To that end, we experiment with sequence lengths of 4096 and 10000. Since the original input representations have different temporal resolutions, we employ various downsampling rates to ensure that the models receive comparable feature rates during analysis. In our study with short context (4096 frames) dynamics modeling, we downsample the Bark features by 8 to operate at 16 ms, and downsample the log-Mel features by 3 to operate at 17.4 ms. For modeling dynamics detection using longer contexts (10000 frames), we downsample the log-Mel features by 5 to achieve a temporal resolution of 29 ms, and downsample the Bark features by 15 to achieve a comparable resolution of 30 ms.

Model Architecture and Training: For the frame-level estimation of dynamics, we employ a multi-scale Convolutional Neural Network (CNN) with self-attention² [35], originally introduced for the closely related task of frame-wise playing technique detection. In our implementation, the network receives input features with a fixed sequence length and outputs probabilities for 10 dynamics classes, with the class having the highest probability taken as the estimate. During training, we utilize the Adam optimizer with a learning rate of 0.002, aiming to reduce the Cross Entropy loss between the predicted dynamics classes and the aligned dynamics labels. We report our results on training the same network for different input representations, sequence lengths, and feature rates.

Metrics: One big challenge in the experimentation with musical dynamics is the subjectivity and relativity in its evaluation. For instance, one piece may span dynamics ranging from *pp* to *f*, and another piece may span dynamics ranging from *p* to *ff*. However, the measured loudness values of performances derived from both music pieces might be similar, as both sets of labels indicate a transition from relatively "soft" to "loud" dynamics. Therefore, the mapping between perceived performed dynamics and labeled musical dynamics may not be absolute. To address this challenge, we present the results in terms of exact match (Acc), relaxed accuracy 1 (Acc ± 1), and relaxed accuracy 2 (Acc ± 2). Relaxed accuracy denotes that estimates are not penalized for a mismatch of 1 or 2 classes, respectively.

²Based on the modified version of <https://github.com/LiDCC/GuzhengTech99/blob/main/function/model.py>

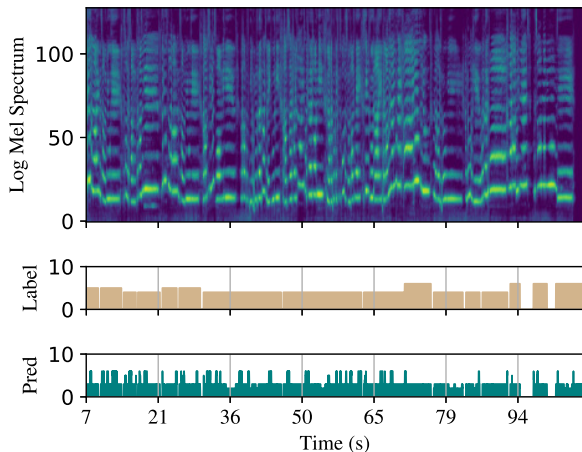


Figure 4: Model input and outputs for the log-Mel spectrum features. log-Mel-spectrogram (top), annotated labels by musician(middle), model estimates (bottom)

4.1 Results

The results are summarised in Table 1. Despite the subjectivity of the task, we observe that the most confusion lie within the ± 1 or ± 2 range with significantly higher relaxed accuracies. Furthermore, we see that bark-based features outperform log-Mel features for the task. The highest relaxed accuracy ± 2 is achieved with bark-based features, indicating the models ability to differentiate between upper and lower bounds of dynamics. For example, a fortissimo is not classified to be a piano in almost 85% of the cases. An example prediction using log-Mel features and bark-based features for a theatre song "sound of music" is presented in Figures 4 and 5 respectively.

The effect of larger and smaller temporal contexts can also be seen in Table 1. Providing larger temporal contexts results in better performance for dynamics estimation. This effect is more prominent for log-Mel features compared to the Bark features. We found that the best performing model is the one with the entire song frames included in the context window i.e., the sequence length is long enough to encapsulate the whole song.

5. DISCUSSION

One of the primary challenges in predicting musical dynamics lies in the fact that performance information is available through recordings, which is a result of mixing and mastering. Consequently, the loudness information captured in recordings may diverge from performers original intentions. However, we contend that despite the influence of mixing and mastering, it is possible for musicians as well as non-musicians to infer whether a performer is singing softly, loudly or even shouting independent of raw loudness levels. Our approach leverages perceptually motivated features that encapsulate timbral characteristics, which have the potential to enhance musical dynamics estimation while remaining agnostic to variations in loudness levels.

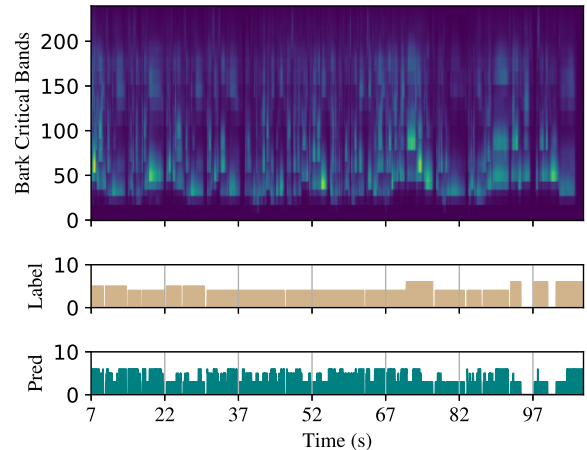


Figure 5: Model input and outputs for the bark based features: bark-critical-bands (top), annotated labels by musician (middle), model estimates (bottom)

While the labels are created semi-automatically, there are potential discrepancies due to performers not adhering strictly to the score or editors creating alternative versions of the score different from the one curated in the dataset.

Additionally, we have framed the dynamics estimation at an absolute level, with the expectation that the model will learn the variations in relative markings given a large amount of data. However, musical dynamics at any given time in a performance depend on the context rather than the absolute value of measured loudness [36]. Additionally, addressing class imbalance remains a significant challenge.

On software front, while MuseScore offers extensive annotation capabilities, some categories cannot be accurately modeled. To mitigate this, musicians often use note-level "TextExpressions" in MuseScore to add additional information. During our experimentation, we encountered terms like "sempre piano," "poco dolce," and "calando" that musicians add to the score. While we were able to mitigate challenges with some labels, achieving comprehensive coverage requires further collaboration with vocalists to refine the target labels.

6. CONCLUSION AND FUTURE WORK

We've developed a methodology for large-scale dataset curation focused on singing voice. The semi-automatically curated dataset serves as a valuable resource for tasks such as transcription, expression analysis, synthesis, and vocal pedagogy. It currently includes 509 performances aligned with 163 score files from 25 composers. Using this dataset, we trained a CNN with multi-head attention for dynamics prediction and found that bark-scale-based features outperform log-Mel features. To test the model, we curated score-performance dataset manually in collaboration with a Classical vocalist. Future work involves integrating pitch features with loudness features to enhance prediction accuracy, improving the model to address class imbalance, and expanding the dataset to include more composers.

7. ACKNOWLEDGMENTS

We would like to thank Ajay Srinivasamurthy for his invaluable feedback. IA y Música: Cátedra en Inteligencia Artificial y Música" (TSI-100929-2023-1), funded by the Secretaría de Estado de Digitalización e Inteligencia Artificial, and the European Union-Next Generation EU, under the program Cátedras ENIA 2022 para la creación de cátedras universidad-empresa en IA.

8. REFERENCES

- [1] B. Patterson, "Musical dynamics," *Scientific American*, vol. 231, no. 5, pp. 78–95, 1974.
- [2] D. Fabian, R. Timmers, and E. Schubert, *Expressiveness in music performance: Empirical approaches across styles and cultures*. Oxford University Press, USA, 2014.
- [3] R. Miller, *On the Art of Singing*. Oxford University Press, 1996. [Online]. Available: <https://books.google.es/books?id=Sv1EAQAACAAJ>
- [4] F. Bous and A. Roebel, "Analysis and transformation of voice level in singing voice," in *Proceedings of ICASSP*. Rhodes Island, Greece: IEEE, 2023, pp. 1–5.
- [5] J. Sundberg, "Perceptual aspects of singing," *Journal of voice*, vol. 8, no. 2, pp. 106–122, 1994.
- [6] L. Bishop, F. Bailes, and R. T. Dean, "Performing musical dynamics: How crucial are musical imagery and auditory feedback for expert and novice musicians?" *Music Perception: An Interdisciplinary Journal*, vol. 32, no. 1, pp. 51–66, 2014.
- [7] A. Berndt and T. Hähnel, "Modelling musical dynamics," in *Proceedings of the 5th Audio Mostly Conference: A Conference on Interaction with Sound*, 2010, pp. 1–8.
- [8] G. Widmer and W. Goebel, "Computational models of expressive music performance: The state of the art," *Journal of new music research*, vol. 33, no. 3, pp. 203–216, 2004.
- [9] A. Friberg, E. Schoonderwaldt, A. Hedblad, M. Fabiani, and A. Elowsson, "Using listener-based perceptual features as intermediate representations in music information retrieval," *The Journal of the Acoustical Society of America*, vol. 136, no. 4, pp. 1951–1963, 2014.
- [10] A. Elowsson and A. Friberg, "Predicting the perception of performed dynamics in music audio with ensemble learning," *The Journal of the Acoustical Society of America*, vol. 141, no. 3, pp. 2224–2242, 2017.
- [11] K. Kosta, R. Ramírez, O. F. Bandtlow, and E. Chew, "Mapping between dynamic markings and performed loudness: a machine learning approach," *Journal of Mathematics and Music*, vol. 10, no. 2, pp. 149–172, 2016.
- [12] K. Kosta, O. F. Bandtlow, and E. Chew, "Dynamics and relativity: Practical implications of dynamic markings in the score," *Journal of New Music Research*, vol. 47, no. 5, pp. 438–461, 2018.
- [13] D. Jeong and J. Nam, "Note Intensity Estimation of Piano Recordings by Score-Informed NMF," in *2017 AES International Conference on Semantic Audio*, Erlangen, Germany, Jun. 2017.
- [14] L. Marinelli, A. Lykartsis, S. Weinzierl, and C. Saitis, "Musical dynamics classification with CNN and modulation spectra," in *Proceedings of the 17th Sound and Music Computing Conference*, Torino, Italy, 2020, pp. 193–199.
- [15] M. Blaauw and J. Bonada, "A neural parametric singing synthesizer modeling timbre and expression from natural songs," *Applied Sciences*, vol. 7, no. 12, p. 1313, 2017.
- [16] M. Umbert, J. Bonada, M. Goto, T. Nakano, and J. Sundberg, "Expression control in singing voice synthesis: Features, approaches, evaluation, and challenges," *IEEE Signal Processing Magazine*, vol. 32, no. 6, pp. 55–73, 2015.
- [17] T. Nakano and M. Goto, "Vocalistner2: A singing synthesis system able to mimic a user's singing in terms of voice timbre changes as well as pitch and dynamics," in *Proceedings of ICASSP*. Prague, Czech Republic: IEEE, 2011, pp. 453–456.
- [18] J. Narang, M. Miron, A. Srinivasamurthy, and X. Serra, "Analysis of musical dynamics in vocal performances using loudness measures," in *Proceedings of the 25th International Conference on Digital Audio Effects (DAFx 2022)*. DAFx, 2022, pp. 33–39.
- [19] P. Proutskova, C. Rhodes, T. Crawford, and G. Wiggins, "Breathy, resonant, pressed—automatic detection of phonation mode from audio recordings of singing," *Journal of New Music Research*, vol. 42, no. 2, pp. 171–186, 2013.
- [20] J. Wilkins, P. Seetharaman, A. Wahl, and B. Pardo, "Vocalset: A singing voice dataset," in *Proceedings of the 19th International Society for Music Information Retrieval Conference (ISMIR 2018)*, Paris, France, Sep. 2018, pp. 468–474.
- [21] J. Sundberg, "Research on the singing voice in retrospect," *TMH-QPSR*, vol. 45, no. 1, pp. 11–22, 2003.
- [22] M. R. H. Gotham and P. Jonas, "The OpenScore Lieder Corpus," in *Music Encoding Conference Proceedings 2021*, S. Münnich and D. Rizo, Eds. Humanities Commons, 2022, pp. 131–136.
- [23] M. S. Cuthbert and C. Ariza, "music21: A toolkit for computer-aided musicology and symbolic music

- data,” in *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR 2010)*, Utrecht, Netherlands, Aug. 2010, pp. 637–642.
- [24] A. Défossez, “Hybrid spectrogram and waveform source separation,” in *Proceedings of the ISMIR 2021 Workshop on Music Source Separation*, Nov. 2021.
- [25] J. Narang, M. Miron, X. Lizarraga Seijas, and X. Serra, “Analysis of musical dynamics in vocal performances,” in *Proceedings of the 15th International Symposium on Computer Music Multidisciplinary Research (CMMR 2021)*, Tokyo, Japan, Nov. 2021, pp. 99–108.
- [26] N. C. Tamer, P. Ramoneda, and X. Serra, “Violin etudes: a comprehensive dataset for f0 estimation and performance analysis,” pp. 517–524, 2022.
- [27] N. C. Tamer, Y. Özer, M. Müller, and X. Serra, “High-resolution violin transcription using weak labels,” in *Proceedings of the 24th International Society for Music Information Retrieval Conference (ISMIR 2023)*, Milan, Italy, Nov. 2023, pp. 223–230.
- [28] M. Müller, Y. Özer, M. Krause, T. Prätzlich, and J. Driedger, “Sync toolbox: A python package for efficient, robust, and accurate music synchronization,” *Journal of Open Source Software*, vol. 6, no. 64, p. 3434, 2021.
- [29] C. Weiß, F. Zalkow, V. Arifi-Müller, H. Grohganz, H. V. Kooops, A. Volk, and M. Müller, “Schubert Winterreise dataset: A multimodal scenario for music analysis,” *ACM Journal on Computing and Cultural Heritage (JOCCH)*, 2020, in press.
- [30] J. W. Kim, J. Salamon, P. Li, and J. P. Bello, “Crepe: A convolutional representation for pitch estimation,” in *Proceedings of ICASSP*, Calgary, Canada, 2018, pp. 161–165.
- [31] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, “librosa: Audio and music signal analysis in python,” in *Proceedings of the 14th python in science conference*, vol. 8. Citeseer, 2015, pp. 18–25.
- [32] G. F. Coop, “Mosquito,” Feb. 2024. [Online]. Available: <https://doi.org/10.5281/zenodo.10629475>
- [33] R. San Millán-Castillo, E. Latorre-Iglesias, M. Glesser, S. Wanty, D. Jiménez-Caminero, and J. M. Álvarez-Jimeno, “Mosquito: an open-source and free toolbox for sound quality metrics in the industry and education,” in *INTER-NOISE and NOISE-CON Congress and Conference Proceedings*, vol. 263, no. 5. Institute of Noise Control Engineering, 2021, pp. 1164–1175.
- [34] E. Zwicker, H. Fastl, U. Widmann, K. Kurakata, S. Kuwano, and S. Namba, “Program for calculating loudness according to din 45631 (iso 532b),” *Journal of the Acoustical Society of Japan (E)*, vol. 12, no. 1, pp. 39–42, 1991.
- [35] D. Li, M. Che, W. Meng, Y. Wu, Y. Yu, F. Xia, and W. Li, “Frame-level multi-label playing technique detection using multi-scale network and self-attention mechanism,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.
- [36] T. Nakamura, “The communication of dynamics between musicians and listeners through musical performance,” *Perception & psychophysics*, vol. 41, pp. 525–533, 1987.

JOINT AUDIO AND SYMBOLIC CONDITIONING FOR TEMPORALLY CONTROLLED TEXT-TO-MUSIC GENERATION

Or Tal^{*1,2} Alon Ziv^{*1} Itai Gat²
Felix Kreuk² Yossi Adi^{1,2}

¹The Hebrew University of Jerusalem

²Meta, FAIR Team

{or.tal1, alon.ziv1}@mail.huji.ac.il

ABSTRACT

We present JASCO, a temporally controlled text-to-music generation model utilizing both symbolic and audio-based conditions. JASCO can generate high-quality music samples conditioned on global text descriptions along with fine-grained local controls. JASCO is based on the Flow Matching modeling paradigm together with a novel conditioning method that allows for both locally (e.g., chords) and globally (text description) controlled music generation. Specifically, we apply information bottleneck layers in conjunction with temporal blurring to extract relevant information with respect to specific controls. This allows the incorporation of both symbolic and audio-based conditions in the same text-to-music model. We experiment with various symbolic control signals (e.g., chords, melody), as well as with audio representations (e.g., separated drum tracks, full-mix). We evaluate JASCO considering both generation quality and condition adherence using objective metrics and human studies. Results suggest that JASCO is comparable to the evaluated baselines considering generation quality while allowing significantly better and more versatile controls over the generated music. Samples are available on our demo page <https://pages.cs.huji.ac.il/adiyoss-lab/JASCO>

1. INTRODUCTION

Conditional music generation has shown a great improvement in recent years, specifically in the task of *text-to-music* generation [1–6]. Such advancements in music generation hold great potential to empower content creators, advertisers, and video game designers. Though presenting highly realistic music samples, most of the prior work is focused on global conditioning only. Such methods mainly consider textual descriptions or melody in the form

*Equal contribution

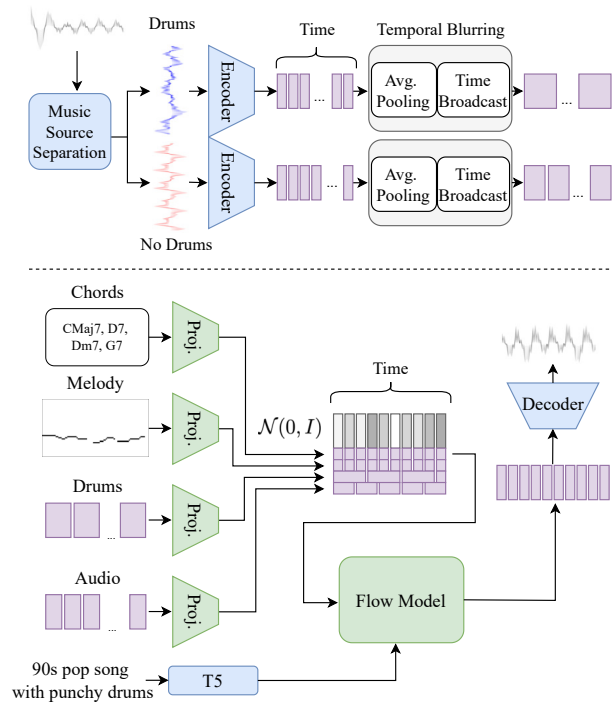


Figure 1. Top figure presents the temporal blurring process, showcasing source separation, pooling and broadcasting. Bottom figure presents a high level presentation of JASCO. Conditions are first being projected to low dimensional representation and are concatenated over the channel dimensions. Green blocks have learnable parameters while blue block are frozen.

of spectral features [3]. However, when considering music production, global controls may not be enough. During the creative process, professional musicians often use chords, melodies, or audio prompts, at the local level, rather than global descriptions. As a result, current models may be limited in their relevancy for music creators.

More recently, several works study text-to-music generation using temporally aligned controls. The authors in [7] suggest adding symbolic beat and dynamics conditions on top of the previously explored melody conditioning. The authors in [8] further explore musical structure conditioning, such as A-part and B-part. Unlike these works, the proposed method provides local controls considering both

symbolic representation and raw audio together with a global textual description. When considering music editing, the authors in [9] propose leveraging chord progression to guide the generation process towards the harmony of the inputs signal. For that, the authors extract an internal representation from stemmed data using a pre-trained chord classification model. The proposed method is different as we focus on generating full musical pieces rather than editing a given one. Specifically, we allow symbolic chord progression conditioning during inference time.

In this work, we present JASCO, a locally controlled Joint Audio and Symbolic COnditioning text-to-music model. JASCO uses time-aligned controls, namely audio prompts, melodies and chord progressions, comprised of either symbolic signals or raw waveforms. We relieve the need for either studio quality stemmed data or supervised datasets by using off-the-shelf pre-trained models to automatically extract the relevant information. JASCO is based on the Flow-Matching [10] modeling paradigm. Figure 1 provides a high level description of the proposed method. Results suggest that JASCO achieves comparable performance in terms of generation quality w.r.t the evaluated baselines while allowing significantly richer set of controls. The main contributions of this work are as follows: (i) We introduce a simple yet effective approach for audio conditioning with high temporal-adherence. (ii) We offer specific evaluation metrics to measure the alignment and accuracy of our suggested controls. (iii) We provide a thorough analysis on the components composing JASCO and compare to several baselines.

2. BACKGROUND

Audio Representation. Modern audio generative models mostly operate on a latent representation of the audio, commonly obtained from a compression model [11–13]. Compression models such as [14] employ Residual Vector Quantization (RVQ) which results in several parallel streams. Each stream is comprised of discrete tokens originating from different learned codebooks.

Specifically, the authors in [14] introduced EnCodec, a convolutional auto-encoder with a latent space quantized using RVQ [15], and an adversarial reconstruction loss. Given a reference audio signal $x \in \mathbb{R}^{D \cdot f_s}$ with D the audio duration and f_s the sample rate, EnCodec first encodes it into a continuous latent tensor $z \in \mathbb{R}^{D \cdot f_r \times N_{\text{enc}}}$ with a frame rate $f_r \ll f_s$ and $N_{\text{enc}} = 128$. Then, z is quantized into $q \in \{1, \dots, N\}^{D \cdot f_r \times K}$, with K being the number of codebooks used in RVQ and N being the codebook size. After quantization, we are left with K discrete token sequences, each of length $T = D \cdot f_r$, representing the audio signal. In RVQ, each quantizer encodes the quantization error left by the previous quantizer, thus quantized values for different codebooks are in general dependent, where the first codebook holds most of the information. Finally, the quantized representation is decoded back to a time domain signal using the decoder network applied to the sum of the representations learned by the different codebooks.

In JASCO, we use the continuous tensor z as the latent representation, while leveraging the discrete representation q for audio conditioning.

Flow Matching. The Flow Matching modeling paradigm [10] was recently found to provide impressive results on image [10], speech [16] and environmental sound generation [17]. More specifically, Conditional Flow Matching (CFM) is a novel training technique for Continuous Normalizing Flow models [18], that captures the continuous transformation paths of samples from a basic prior distribution, usually standard normal $\mathcal{N}(0, 1)$, to their counterparts in a target data distribution, \mathcal{S} . The position on this path is denoted by a time parameter t , starting from the prior state at $t = 0$ and ending at the data state at $t = 1$.

In this work, we focus on Optimal Transport (OT) paths as defined in [10]. The model is trained to predict the vector field of the continuous latent audio variable z , given t and a set of conditions \mathbf{Y} . Formally, the model minimizes the regression loss

$$\mathcal{L}_{\text{CFM}}(\theta; z_0, z_1, t | \mathbf{Y}) = \|v_\theta(z, t | \mathbf{Y}) - (z_1 - (1 - \sigma_{\min}) \cdot z_0)\|^2, \quad (1)$$

where $z_0 \sim \mathcal{N}(0, I)$ is a sampled noise, $z_1 \sim \mathcal{S}$ is the latent representation of a data sample, and

$$z = (1 - (1 - \sigma_{\min}) \cdot t) \cdot z_0 + t \cdot z_1, \quad (2)$$

is an interpolation between the noise and the data sample. For numerical stability, we use a small value $\sigma_{\min} = 10^{-5}$ in both terms. During inference we follow an iterative process, starting with the prior noise $z \leftarrow z_0 \sim \mathcal{N}(0, 1)$ and with $t = 0$. In each step, we translate the estimated vector field $v_\theta(z, t | \mathbf{Y})$ into an updated latent sequence z , and gradually converge toward the data distribution.

3. METHOD

Given a textual description, and a set of temporal conditions - such as melody, chord progression or drum recording, our goal is to produce high-quality samples that are musically aligned with the given controls, while complying to the arrangement description provided in the text.

JASCO tackles the aforementioned problem by a CFM model, operating on the continuous latent space of EnCodec. JASCO is conditioned on low-dimensional embeddings of melody, chords and audio signals, together with a T5 embedding of the textual description. All local controls are concatenated to the model’s input across the feature dimension, while text is being passed via cross attention. To diminish timbre-related information, JASCO further applies temporal blurring to the audio-based controls, as well as band-pass filtering. See Figure 1 for a visual description, and Section 3.1 for detailed information.

3.1 Temporal Controls

Symbolic. We use Chordino¹ chord progression model to extract an integer categorical chord label sequence, and a

¹ <https://github.com/ohollo/chord-extractor>

pretrained multi-F0 classifier [19] to obtain melody scores per time step. We resample all features to match EnCodec’s frame rate using nearest-interpolation for chords and linear-interpolation for melody. For Chords, we use a learned embedding table to map the raw integer sequence, denoted as \mathbf{c}_{crd} , to its corresponding condition matrix in shape $T \times d_{\text{crd}}$. For Melody, we zero out values with a score lower than a pre-defined threshold (0.5). Then, we select the maximal non-zero score per time step from the remaining values, and set it to 1 while setting the rest to 0. This yields a binary matrix $\mathbf{c}_{\text{mld}} \in \{0, 1\}^{D \cdot f_r^{\text{mld}} \times N_{\text{mld}}}$. Finally, we linearly project the binary matrix and obtain the melody condition representation in shape $T \times d_{\text{mld}}$. We use $N_{\text{mld}} = 53$ (corresponding to G2-B7 notes), and $d_{\text{crd}} = d_{\text{mld}} = 16$.

Audio. We consider general audio and separated drum stems. We use a pretrained source separation model [20], to extract the drum stem from a source audio. We pass the waveform through EnCodec to obtain the corresponding quantized discrete representation \mathbf{q} . We then convert the first token stream back to its continuous latent representation, using EnCodec’s first codebook while discarding all other streams, yielding $\mathbf{c}_{\text{aud}}, \mathbf{c}_{\text{drm}} \in \mathbb{R}^{T \times N_{\text{enc}}}$. We chose to use only the first codebook stream to further discard timbre information, stressing the forced information-bottleneck further. Following that, we apply temporal blurring to the reconstructed latent. First, we apply average pooling using non-overlapping windows along the temporal axis. Then, we broadcast the signal to its original temporal dimension. Finally, we linearly project the blurred condition to a low dimensional feature space and obtain the condition matrix. For general audio, we use a window size of 5 and output dimension of 1, while for drums we use a window size of 3 and output dimension of 2.

Inpainting and Outpainting. In/Out-painting is the task of filling in a masked region, where in/out refers to the masked segment position in the sequence, be it at the middle (in) or at the end (out). Following prior work [5], we add in/out-painting as an additional condition to the model. We randomly choose between inpainting/outpainting, and mask a random segment of 40-90% from the reference waveform. Then, we use the raw EnCodec latent representation of the masked waveform $\mathbf{c}_{\text{iop}} \in \mathbb{R}^{T \times N_{\text{enc}}}$ as the condition, with no learned projection.

3.2 Model and Optimization

Similarly to prior work [17], our CFM model consists of a Transformer, with U-Net-like residual connections. We replace the standard residual addition with channel-wise concatenation followed by a linear projection. We use learned convolutional positional encoding [21] as well as symmetric bi-directional ALiBi self-attention biases [22]. We use a model scale of 330M parameters, with 24 Transformer layers, 16 attention heads, embedding dimensionality of 1024 and a feed-forward dimension of 4096.

We train our model using the \mathcal{L}_{CFM} objective as defined in Section 2. We further experiment with non-uniform loss

weighting as function of t , and find the following formulation to produce the best overall sample quality:

$$\mathcal{L}_{\text{WeightedCFM}} = \sum_{\substack{t \sim \mathcal{U}(0,1) \\ \mathbf{z}_0 \sim \mathcal{N}(0,1) \\ \mathbf{z}_1 \sim \mathcal{S}}} (1+t) \cdot \mathcal{L}_{\text{CFM}}(\theta; \mathbf{z}_0, \mathbf{z}_1, t | \mathbf{Y}), \quad (3)$$

where $\mathbf{Y} = \{\mathbf{c}_{\text{crd}}, \mathbf{c}_{\text{mld}}, \mathbf{c}_{\text{aud}}, \mathbf{c}_{\text{drm}}, \mathbf{c}_{\text{iop}}\}$. We provide an ablation study for this scheme in Section 5.

3.3 Inference

During inference, as in [10], we use *dopri5* [23], an off-the-shelf numerical ODE solver, to iteratively solve for \mathbf{z} given the estimated vector field v_θ . Specifically, at each iteration the solver determines the increment to the time parameter t , resulting in a dynamic scheduling for the inference process. The process halts when an acceptance criterion is met, defined by an error approximation of the solver and a tolerance parameter provided by the user.

Multi-Source Classifier Free Guidance. We employ classifier-free guidance (CFG) [24] for the conditional vector field estimation $v_\theta(\mathbf{z}, t | \mathcal{Y})$. Since our set of conditioning signals combines both global and local concepts, we further experiment with multi source CFG. While prior work [25] suggest a separate evaluation for each condition, we evaluate the model considering all and partial conditions. During each inference step, we obtain an estimated vector field for each set of conditions $\mathcal{Y} \in \{\{\text{local}\}, \{\text{text}\}, \{\text{local}, \text{text}\}\}$. The resulting CFG formulation then follows:

$$\text{CFG}(v_\theta, \mathbf{z}, t) = (1 - \sum_{c \in \mathcal{Y}} \alpha_c) v_\theta(\mathbf{z}, t) + \sum_{c \in \mathcal{Y}} \alpha_c v_\theta(\mathbf{z}, t | c). \quad (4)$$

When following the standard CFG setup ($\alpha_{\text{text}} = \alpha_{\text{local}} = 0$), we observe that the model adheres to the temporal condition while ignoring instrumentation information provided in the text prompt. To increase text influence on guidance, we set a positive weight to the text-only term $\alpha_{\text{text}} > 0$. We found that $\alpha_{\text{text}} = 0.5, \alpha_{\text{local}} = 0, \alpha_{\text{local}, \text{text}} = 1.5$ offer a good trade-off between audio quality, text alignment and temporal controls adherence.

4. EXPERIMENTAL SETUP

Implementation Details. We follow the same experimental setup as in [3, 6], and use a training dataset consisting of 20K hours of licensed music from the Shutterstock² and Pond5³ data collections with 25K and 365K instrument-only music tracks, respectively. We additionally include a set of proprietary data consisting of 10K high-quality tracks. All datasets are sampled at 32kHz, paired with textual descriptions. We present results on the MusicCaps benchmark [1], comprising 5.5K 10-second samples together with an in-domain test set of 528 tracks.

² shutterstock.com/music

³ pond5.com

We use the official EnCodec model provided by [3, 12], with a frame rate of 50 Hz, and 4 codebooks, each with a size of 2048. For text representation we use a pretrained T5 model [26]. For melody extraction we use the pretrained deep salience multi-F0 detector⁴, for chords extraction we use Chordino, while for drum track extraction we use the Hybrid Demucs model [27].

All single condition models were trained with 40% condition dropout, and in the multi-condition experiments we train the models with 20% condition dropout for all conditions. In the remaining 80% we set 50% dropout for each of the conditions independently excluding the in/out-painting, for which we set 70% dropout.

We experiment with multi-source CFG coefficients in $(\alpha_{\text{text}}, \alpha_{\text{local}}, \alpha_{\text{text,local}}) \in \{0.0, 0.5\} \times \{0.0, -0.5\} \times \{1.5, 2.0\}$ and report the best overall configuration. All models were trained for 500k steps over audio segments of 10 seconds, with a batch size of 336. We use Adam [28] optimizer with linear learning rate warm-up up to a peak of 10^{-4} during the first 5k steps, followed by a linear decay, and a gradient clipping with a norm threshold of 0.2.

4.1 Evaluation Metrics

We perform a thorough empirical evaluation, using both objective metrics and human studies. We evaluate JASCO on several temporal alignment aspects, namely harmonic matching, rhythmic alignment and melody preservation. Additionally, we measure audio quality and text adherence.

Objective Evaluations. We evaluate our method with widely used metrics, namely Fréchet Audio Distance (FAD), Kullback-Leiber Divergence (KL) and CLAP score (CLAP), as well as more specific metrics designed to quantify the adherence of our suggested controls. We report FAD [29] using the official tensorflow implementation where a low FAD score indicates that the generated audio is associated with higher quality. Following [3, 12], we use an audio classifier [30] to compute the KL-divergence over the probabilities of the labels between the original and the generated music. The generated music is expected to share similar concepts with the reference music when the KL is low. Last, CLAP score [31, 32] is computed between the track description and the generated audio, measuring audio-text alignment. We use the official pretrained CLAP model⁵. To evaluate melody compatibility, similar to [3] we use a cosine similarity metric on either a simple quantized chroma representation, or multi-octave melody representation obtained from a pretrained multi-F0 classifier [19]. For beat adherence, as in [7] we evaluate the onset F1 score using *mir eval*⁶ considering a 50ms tolerance margin around classified onsets in the reference signal. Lastly, to evaluate chord progression, we use the Chordino model to extract the chord progression from both the reference and the generated signals and compute the intersection over union (IOU) score between the two.

Model	FAD↓	CLAP↑	Mel Sim.↑	Mel Acc.↑
MusicGen	5.90	0.29	0.61	44.0
MusicControlNet	10.81	0.22	-	47.1
JASCO	6.05	0.26	0.67	49.1

Table 1. Melody conditioning evaluation over MusicCaps. We evaluated MusicGen with 300M parameters.

Human Study. We request raters to evaluate three aspects of given audio samples: (i) overall quality; (ii) similarity to text description; and (iii) adherence to either melody or rhythmic pattern from a reference recording. Raters were instructed to rate the recordings on a scale between 0-100 where higher is better. Raters were recruited using the Amazon Mechanical Turk platform. We evaluate randomly sampled files, where each sample was evaluated by at least 5 raters. We use the CrowdMOS package [33] to filter noisy annotations and outliers. We remove annotators who did not listen to the full recordings, annotators who rate the reference recordings less than 90, and the rest of the recommended recipes from [33]. Similarly to [3], for a fair comparison, all samples are normalized at -14dB LUFS [34]. Overall, we 179 raters evaluated the generation quality, 121 raters evaluated the text relevancy, 159 raters evaluated the adherence to rhythm patterns using drum conditioning, and 142 raters evaluated melody conditioning.

5. RESULTS

Melody Conditioning. We start by evaluating the proposed method considering melody conditioning. We compare JASCO to MusicGen [3] and MusicControlNet [7]. For a fair comparison, we train MusicGen (300M) on 10 second music segments using Audioscraft⁷ repository, considering text and melody conditions. For comparison compatibility with [7] we compute melody accuracy score on both JASCO and MusicGen. We experiment with melody conditioning using the commonly used 12-bins chroma representation which is octave invariant. Results are presented in Table 1.

Results suggest that JASCO surpasses the evaluated baselines w.r.t melody adherence. When considering melody accuracy, JASCO provides better alignment to the conditioning melody. Notice, we hypothesize this is due to the conditioning method: both MusicGen and MusicControlNet inject conditions as an additive bias (i.e., cross-attention and zero-convolutions), this is in contrary to JASCO which follows the concatenation approach for melody conditioning (see Section 6 for more experiments).

Local Controls. We train a single-condition variant for each observed condition-type as well as two multi-condition models. Under the multi-condition setup, we train models with Drums tracks passed through a Band-Pass-Filter (BPF) over 200-800 Hz frequency range, and

⁴ github.com/rabitt/ismir2017-deepsalience

⁵ github.com/LAION-AI/CLAP

⁶ github.com/craffel/mir_evaluators

⁷ <https://github.com/facebookresearch/audiocraft/blob/main/docs/MUSICGEN.md>

Local Controls				Objective metrics (MusicCaps / Internal dataset)						
Aud	Drum	Crds	Mld	Mld (clf) sim. ↑	Mld sim. ↑	Onset F1 ↑	Crds IOU ↑	FAD ↓	KL ↓	CLAP ↑
-	-	-	-	0.13 / 0.13	0.09 / 0.09	0.34 / 0.41	0.09 / 0.07	6.04 / 0.90	1.46 / 0.70	0.27 / 0.36
✓	-	-	-	0.33 / 0.34	0.38 / 0.47	0.62 / 0.81	0.23 / 0.27	4.47 / 0.86	0.92 / 0.81	0.30 / 0.31
no drm	-	-	-	0.21 / 0.22	0.38 / 0.31	0.62 / 0.58	0.23 / 0.18	5.68 / 0.92	1.79 / 0.75	0.19 / 0.33
-	✓	-	-	0.13 / 0.13	0.09 / 0.10	0.62 / 0.73	0.09 / 0.08	5.85 / 0.94	1.68 / 0.78	0.23 / 0.35
-	BPF	-	-	0.13 / 0.13	0.10 / 0.10	0.45 / 0.74	0.10 / 0.07	6.31 / 1.61	1.52 / 0.65	0.26 / 0.37
-	-	✓	-	0.21 / 0.25	0.22 / 0.29	0.24 / 0.13	0.59 / 0.61	7.23 / 0.95	1.16 / 0.68	0.28 / 0.36
-	-	-	✓	0.67 / 0.64	0.41 / 0.35	0.37 / 0.57	0.31 / 0.27	6.96 / 1.05	1.32 / 0.63	0.27 / 0.35
-	BPF	✓	✓	0.68 / 0.69	0.44 / 0.46	0.63 / 0.66	0.50 / 0.53	6.42 / 1.15	1.22 / 0.50	0.28 / 0.37
no drm	BPF	✓	✓	0.71 / 0.68	0.50 / 0.55	0.54 / 0.75	0.51 / 0.55	4.78 / 0.80	0.93 / 0.41	0.30 / 0.37

Table 2. Objective local controls experiment, observing all suggested controls w.r.t a zero hypothesis (no local controls).

Model	Cond.	Q	T	M	D
Reference	-	92.7±0.66	93.7±0.8	96.3±0.6	97.1±0.6
MusicGen	T	84.4±0.8	84.5±0.9	81.5±1.3	82.1±1.0
JASCO	T	83.3±0.7	80.3±1.3	79.7±1.5	81.5±1.1
MusicGen	T & M	84.7±0.7	82.5±1.1	83.6±1.1	82.7±0.9
JASCO	T & M	84.1±0.7	81.2±1.2	89.3±0.7	80.6±1.2
JASCO	T & D	85.5±0.8	84.1±1.1	81.9±1.4	89.5±0.7

Table 3. Human evaluation results. Observing general quality (Q), text match (T) melody match (M) and drums match (D). Evaluated on a 0-100 scale (higher is better).

Audio condition excluding drums. This was found to better disentangle Drums and Audio conditions in preliminary experiments, and allows users to provide different drum beats than the one presented in the Audio. When applying Audio/Drums conditions, we evaluate Melody, Onset F1, and Chord IoU using the reference audio as a condition, while for the computation FAD, KL, and CLAP scores we use a randomly selected audio from the test set.

As there are no open-source relevant baselines available, we compare the proposed method against a text-only condition model. We perform experiments using both the open source MusicCaps dataset, and an internal proprietary dataset, highlighting our model performance on diverse, high quality recordings. Table 2 summarizes the results.

Results depict a systematic improvement considering local control adherence. For instance, chords conditioning shows apparent improvement in Chords IOU metric, improving from 0.09/0.07 to 0.59/0.61. In addition, in spite of being evaluated with randomly selected audio conditions, FAD, KL, CLAP scores mostly remain comparable w.r.t to the baseline. This highlights JASCO’s disentangling property as local controls metrics improve while text adherence and audio quality metrics stay roughly the same.

The lower section of the table presents multi-control setup results. This section draws a similar trend to the single control setups, allowing for multiple controls while preserving FAD, KL, CLAP. This highlights JASCO’s ability to incorporate multiple controls simultaneously with no significant penalty to quality and text alignment.

Human Study. Lastly, we perform a human study in or-

der to validate both quality and text alignment as well as local control adherence. We evaluate JASCO vs MusicGen considering: (i) text only; and (ii) both text and melody. We additionally, provide results of the proposed method with text and drums conditions. Results seen on Table 3, indicate that JASCO achieve similar generation quality as MusicGen across all setups. As of text relevancy, MusicGen reaches superior performance to the proposed method, however, when considering melody conditioning, JASCO reaches significantly better scores. Lastly, when conditioned on drums, JASCO provides the best rhythmic pattern similarity scores. This highlights JASCO’s ability to provide better controls over the generated music without sacrificing quality and text alignment. Interestingly, after including melody or drums conditions, as expected, the relevant metrics are improving (i.e., melodic and rhythmic similarity) while the quality and text adherence remain comparable to the unconditioned model.

6. ANALYSIS

Condition Injection Method. We compare the proposed method to two widely used condition injection methods proposed in prior work. Specifically, we perform a controlled experiment in which we evaluate cross-attention as used in MusicGen, and zero-convolution as used in MusicControlNet, considering the same training configuration.

Results shown in Table 4 suggest that the temporal adherence using the concatenation method performs the best overall. This can be seen in both higher Chord IoU, as well as better FAD and KL, where CLAP was 0.36 for all methods. Additionally, the concatenation method allows training from scratch as opposed to zero-convolutions, in which we start from a pretrained model) without a significant increase in the number of trainable parameters.

Flow vs. Diffusion. Most of prior work on music generation is mainly based on Diffusion models [2, 4, 5, 35]. In this experiment we evaluate, under controlled settings, both Diffusion (v-Diffusion) and Flow Matching modeling approaches for music generation. We report FAD, KL, and CLAP scores. Results are depicted in Figure 2. As can be seen, the Flow Matching approach is superior across all metrics, with the biggest gap observed in FAD.

Conditioning	Chord IOU \uparrow	FAD \downarrow	KL \downarrow
Concat	0.6	1.19	0.71
Cross Attn.	0.59	1.61	0.73
Zero Conv	0.26	1.64	0.74

Table 4. Ablation for conditioning method. evaluated on internal dataset. All models started from a text-to-music pretrained checkpoint and trained for 500K steps.

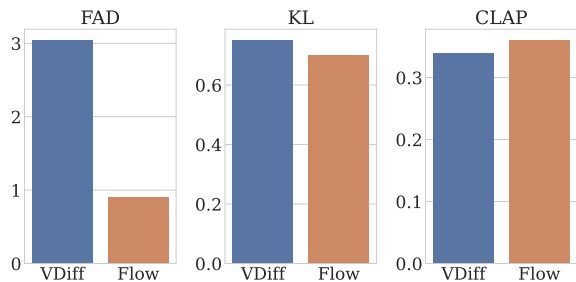


Figure 2. Comparison of v-Diffusion vs Flow Matching. We report FAD, KL, and CLAP on the internal dataset.

The Effect of Weighted Loss. Finally, we evaluate the effect of the proposed modification to the loss function as presented in Equation (3). We compare the proposed objective function against the loss as describe in Equation (1), considering FAD, KL, and CLAP scores in Table 5. Results suggest the new objective function modification improves the generation quality. It provides significantly better FAD while having comparable KL and CLAP scores.

7. RELATED WORK

Flow Matching for Audio Generation. Flow Matching [10] was recently studied for speech generation. A notable work in this context presented VoiceBox [16], a Flow Matching model, operating on spectrograms, for text-guided multilingual speech generation. More recently, AudioBox [17] was presented, in which self-supervised infilling objectives were leveraged to improve the generalization capabilities of VoiceBox. Similar to our model, AudioBox operates on the continuous latent representations of EnCodec [14]. Though the scope of audio modalities was extended in AudioBox to both speech and environmental sounds, applying a Flow Matching approach for music generation remained less explored.

Temporally Controlled Music Generation. Recent work offered several forms of temporally restrictive controls for music generation. Melody conditioned text-to-music was studied in MusicLM [1], in which a melody embedding was trained using a dedicated dataset consists of multiple cover versions of musical tracks paired with aligned singing and humming performances. In MusicGen [3] and Music ControlNet [7], the need for supervised data was relieved, and instead an unsupervised melody extraction was performed using the argmax note of the audio chromagram. Audio-to-audio setups were studied for drum gener-

Weighted loss schedule	FAD \downarrow	KL \downarrow	CLAP \uparrow
$w(t) = 1$	1.73	0.71	0.38
$w(t) = 1 + t$	0.99	0.73	0.37

Table 5. Ablation for loss weighting method. Evaluated on internal dataset. All models were trained for 500K steps.

ation conditioned on drumless track [36], accompaniment generation given singing voice [37], and single instrument generation given partial mix [25] [9]. Recently, generation conditioned on multiple symbolic controls was studied in Music ControlNet [7], a spectrogram diffusion text-to-music model, fine-tuned using the ControlNet scheme [38], to generation with melody, beat and dynamics controls. In DITTO [8], inference time optimization was explored, for tiding a text-to-music diffusion model to perform several tasks including inpainting, outpainting, loop generation, melody and dynamics conditioned generation, as well as conditioning on musical structures. In [39], classifier guidance was used to perform music inpainting, outpainting and style transfer given a pretrained unconditional latent diffusion model. Inpainting was further explored in [5], [40], and [41]. Style transfer was explored also in [42] and [9].

8. DISCUSSION

In this work we present JASCO, a temporally controlled text-to-music generation model, supporting both audio and symbolic conditioning. JASCO is based on the Flow Matching modeling paradigm operating over a dense music latent representation. Through extensive experimentation we empirically show JASCO generates high-fidelity samples that can be conditioned on global textual description together with harmony, melody, rhythmic patterns, and overall musical style. Results suggest JASCO provides comparable generation quality to the evaluated baselines while allowing significantly better control over generation.

Limitations. The main limitations of the proposed approach are: (i) Similarly to previous diffusion-based text-to-music models, the length of the generated samples is relatively short (~ 10 seconds) compared to the auto-regressive alternative. Although this can be extrapolated with overlaps, it may limit the capability of the model in capturing global structure in the generated music; (ii) although generating the whole sequence at once, generation time is slower than auto-regressive alternatives, while not supporting streaming capabilities.

Future work. For future work we intend to support additional controls, such as music dynamics, musical structure, etc. together with editing options, e.g., add or replace specific instrument in a given recording. We believe such a research direction, and specifically the proposed approach, holds great potential in empowering musicians, creators, and producers which require richer set of controls during their creative process.

9. ETHICAL STATEMENT

The use of large-scale generative models raises several ethical concerns. To mitigate at some of them, we first made sure all the data used for training our models was obtained legally through an agreement with Shutterstock. Another issue is the potential lack of diversity in the dataset, which predominantly consists of western-style music. However, we believe that the proposed method is not tied to any specific genera and can help expand the scope of applications to new datasets.

Moreover, generative models could potentially create an unbalanced competitive environment for artists, a problem that is yet to be solved. We are firm believers in the power of open research to provide all participants with equal opportunities to access these models. By introducing more sophisticated controls, like chords and rhythmic patterns as suggested in this work, we aspire to make these models beneficial for both amateurs and professional musicians.

10. REFERENCES

- [1] A. Agostinelli, T. I. Denk, Z. Borsos, J. Engel, M. Verzetti, A. Caillon, Q. Huang, A. Jansen, A. Roberts, M. Tagliasacchi, M. Sharifi, N. Zeghidour, and C. Frank, “Musiclm: Generating music from text,” 2023.
- [2] Q. Huang, D. S. Park, T. Wang, T. I. Denk, A. Ly, N. Chen, Z. Zhang, Z. Zhang, J. Yu, C. Frank, J. Engel, Q. V. Le, W. Chan, Z. Chen, and W. Han, “Noise2music: Text-conditioned music generation with diffusion models,” 2023.
- [3] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Défossez, “Simple and controllable music generation,” 2023.
- [4] F. Schneider, O. Kamal, Z. Jin, and B. Schölkopf, “Moûsai: Text-to-music generation with long-context latent diffusion,” 2023. [Online]. Available: <https://arxiv.org/abs/2301.11757>
- [5] P. Li, B. Chen, Y. Yao, Y. Wang, A. Wang, and A. Wang, “Jen-1: Text-guided universal music generation with omnidirectional diffusion models,” *arXiv preprint arXiv:2308.04729*, 2023.
- [6] A. Ziv, I. Gat, G. L. Lan, T. Remez, F. Kreuk, A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, “Masked audio generation using a single non-autoregressive transformer,” *arXiv preprint arXiv:2401.04577*, 2024.
- [7] S.-L. Wu, C. Donahue, S. Watanabe, and N. J. Bryan, “Music controlnet: Multiple time-varying controls for music generation,” 2023.
- [8] Z. Novack, J. McAuley, T. Berg-Kirkpatrick, and N. J. Bryan, “Ditto: Diffusion inference-time t-optimization for music generation,” 2024.
- [9] B. Han, J. Dai, W. Hao, X. He, D. Guo, J. Chen, Y. Wang, Y. Qian, and X. Song, “Instructme: An instruction guided music edit and remix framework with latent diffusion models,” 2023.
- [10] Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel, and M. Le, “Flow matching for generative modeling,” 2023.
- [11] Z. Borsos, R. Marinier, D. Vincent, E. Kharitonov, O. Pietquin, M. Sharifi, D. Roblek, O. Teboul, D. Grangier, M. Tagliasacchi *et al.*, “Audiolm: a language modeling approach to audio generation,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.
- [12] F. Kreuk, G. Synnaeve, A. Polyak, U. Singer, A. Défossez, J. Copet, D. Parikh, Y. Taigman, and Y. Adi, “Audiogen: Textually guided audio generation,” *arXiv preprint arXiv:2209.15352*, 2022.
- [13] D. Yang, J. Yu, H. Wang, W. Wang, C. Weng, Y. Zou, and D. Yu, “Diffsound: Discrete diffusion model for text-to-sound generation,” *arXiv preprint arXiv:2207.09983*, 2022.
- [14] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, “High fidelity neural audio compression,” 2022.
- [15] N. Zeghidour, A. Luebs, A. Omran, J. Skoglund, and M. Tagliasacchi, “Soundstream: An end-to-end neural audio codec,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2021.
- [16] M. Le, A. Vyas, B. Shi, B. Karrer, L. Sari, R. Moritz, M. Williamson, V. Manohar, Y. Adi, J. Mahadeokar, and W.-N. Hsu, “Voicebox: Text-guided multilingual universal speech generation at scale,” 2023.
- [17] A. Vyas, B. Shi, M. Le, A. Tjandra, Y.-C. Wu, B. Guo, J. Zhang, X. Zhang, R. Adkins, W. Ngan, J. Wang, I. Cruz, B. Akula, A. Akinyemi, B. Ellis, R. Moritz, Y. Yungster, A. Rakotoarison, L. Tan, C. Summers, C. Wood, J. Lane, M. Williamson, and W.-N. Hsu, “Audiobox: Unified audio generation with natural language prompts,” 2023.
- [18] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud, “Neural ordinary differential equations,” 2019.
- [19] R. M. Bittner, B. McFee, J. Salamon, P. Q. Li, and J. P. Bello, “Deep salience representations for f0 estimation in polyphonic music,” in *International Society for Music Information Retrieval Conference*, 2017. [Online]. Available: <https://api.semanticscholar.org/CorpusID:4531539>
- [20] A. Défossez, “Hybrid spectrogram and waveform source separation,” in *Proceedings of the ISMIR 2021 Workshop on Music Source Separation*, 2021.
- [21] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, “wav2vec 2.0: A framework for self-supervised learning of speech representations,” 2020.

- [22] O. Press, N. A. Smith, and M. Lewis, “Train short, test long: Attention with linear biases enables input length extrapolation,” 2022.
- [23] J. R. Dormand and P. J. Prince, “A family of embedded runge-kutta formulae,” *Journal of computational and applied mathematics*, vol. 6, no. 1, pp. 19–26, 1980.
- [24] J. Ho and T. Salimans, “Classifier-free diffusion guidance,” 2022.
- [25] J. D. Parker, J. Spijkervet, K. Kosta, F. Yesiler, B. Kuznetsov, J.-C. Wang, M. Avent, J. Chen, and D. Le, “Stemgen: A music generation model that listens,” 2024.
- [26] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, “Exploring the limits of transfer learning with a unified text-to-text transformer,” *The Journal of Machine Learning Research*, 2020.
- [27] S. Rouard, F. Massa, and A. Défossez, “Hybrid transformers for music source separation,” 2022.
- [28] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2017.
- [29] K. Kilgour, M. Zuluaga, D. Roblek, and M. Sharifi, “Fr\`echet audio distance: A metric for evaluating music enhancement algorithms,” *arXiv preprint arXiv:1812.08466*, 2018.
- [30] K. Koutini, J. Schlüter, H. Eghbal-zadeh, and G. Widmer, “Efficient training of audio transformers with patchout,” *arXiv preprint arXiv:2110.05069*, 2021.
- [31] Y. Wu, K. Chen, T. Zhang, Y. Hui, T. Berg-Kirkpatrick, and S. Dubnov, “Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation,” in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.
- [32] R. Huang, J. Huang, D. Yang, Y. Ren, L. Liu, M. Li, Z. Ye, J. Liu, X. Yin, and Z. Zhao, “Make-an-audio: Text-to-audio generation with prompt-enhanced diffusion models,” in *International Conference on Machine Learning*. PMLR, 2023, pp. 13 916–13 932.
- [33] F. Ribeiro, D. Florêncio, C. Zhang, and M. Seltzer, “Crowdmos: An approach for crowdsourcing mean opinion score studies,” in *2011 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2011, pp. 2416–2419.
- [34] T. Sugimoto, “Loudness-level-chasing algorithm for multiformat live audio production,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 1290–1304, 2022.
- [35] S. Forsgren and H. Martiros, “Riffusion-stable diffusion for real-time music generation. 2022,” URL <https://riffusion.com/about>.
- [36] Y.-K. Wu, C.-Y. Chiu, and Y.-H. Yang, “Jukedrummer: Conditional beat-aware audio-domain drum accompaniment generation via transformer vq-vae,” 2022.
- [37] C. Donahue, A. Caillon, A. Roberts, E. Manilow, P. Esling, A. Agostinelli, M. Verzetti, I. Simon, O. Pietquin, N. Zeghidour, and J. Engel, “Singsong: Generating musical accompaniments from singing,” 2023.
- [38] L. Zhang, A. Rao, and M. Agrawala, “Adding conditional control to text-to-image diffusion models,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 3836–3847.
- [39] M. Levy, B. D. Giorgi, F. Weers, A. Katharopoulos, and T. Nickson, “Controllable music production with diffusion models and guidance gradients,” 2023.
- [40] H. F. Garcia, P. Seetharaman, R. Kumar, and B. Pardo, “Vampnet: Music generation via masked acoustic token modeling,” 2023.
- [41] L. Lin, G. Xia, Y. Zhang, and J. Jiang, “Arrange, in-paint, and refine: Steerable long-term music audio generation and editing via content-based controls,” 2024.
- [42] N. Mor, L. Wolf, A. Polyak, and Y. Taigman, “A universal music translation network,” 2018.

DIFF-A-RIFF: MUSICAL ACCOMPANIMENT CO-CREATION VIA LATENT DIFFUSION MODELS

Javier Nistal¹ Marco Pasini² Cyran Aouameur¹
Maarten Grachten¹ Stefan Lattner¹

¹Sony Computer Science Laboratories, Paris, France

²Queen Mary University of London, UK

javier.nistal@sony.com

ABSTRACT

Recent advancements in deep generative models present new opportunities for music production but also pose challenges, such as high computational demands and limited audio quality. Moreover, current systems frequently rely solely on text input and typically focus on producing complete musical pieces, which is incompatible with existing workflows in music production. To address these issues, we introduce Diff-A-Riff, a Latent Diffusion Model designed to generate high-quality instrumental accompaniments adaptable to any musical context. This model offers control through either audio references, text prompts, or both, and produces 48kHz pseudo-stereo audio while significantly reducing inference time and memory usage. We demonstrate the model’s capabilities through objective metrics and subjective listening tests, with extensive examples available on the accompanying website.¹

1. INTRODUCTION

Deep generative modeling has recently made significant strides, greatly expanding the toolbox for synthesizing visual and auditory art [1–6] and signaling a new era of enhanced creative expression. These technologies promise more intuitive, high-level control over digital creations, yet their deployment in music production comes with inherent challenges. Generative music systems frequently rely solely on text inputs for control and typically focus on generating complete musical pieces rather than individual sounds or instruments. This approach can limit their integration into existing musical workflows and may compromise the artist’s control over the final product. Furthermore, the computational demands of these advanced models often necessitate access to specialized hardware or online services. Additionally, they often fail to meet professional audio standards, such as true stereo output at 48 kHz.

¹ sonycompslparis.github.io/diffariff-companion/



© J. Nistal, M. Pasini, C. Aouameur, M. Grachten, and S. Lattner. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** J. Nistal, M. Pasini, C. Aouameur, M. Grachten, and S. Lattner, “Diff-A-Riff: Musical Accompaniment Co-creation via Latent Diffusion Models”, in *Proc. of the 25th Int. Society for Music Information Retrieval Conf.*, San Francisco, United States, 2024.

In this paper, we introduce Diff-A-Riff, a novel Latent Diffusion Model designed for generating single-instrument accompaniments. A distinct feature of our approach is the ability to condition on musical audio contexts. This specific form of control crucially allows the music to dynamically adapt to the artist’s style, enabling a more personalized creation process. Additionally, the model supports conditioning using joint text-and-audio embeddings from CLAP [7], which can be derived from either textual descriptions or audio references, providing versatile input options for directing the music generation.

At the core of Diff-A-Riff are two pivotal technological elements. First, the efficiency of a Consistency Autoencoder with a high compression rate enhances the system’s performance in terms of inference time and memory usage [8]. Second, the model employs the expressiveness of Elucidated Diffusion Models (EDMs), known for their robust handling of complex data distributions and improved efficiency in model parameterization and inference [9].

We validate Diff-A-Riff through comprehensive evaluations, assessing its performance in ablation studies using objective metrics, and we compare it to other models and estimate context adherence using subjective listening tests. The results, detailed in Sections 5 and 6, demonstrate that our model not only achieves state-of-the-art audio quality (statistically not distinguishable from real audio) but also effectively adapts to various conditional settings confirming its potential for practical applications in music production.

The paper is organized as follows: after a review of related work in Section 2 and background in Section 3, we describe our methodology in Section 4. We then present our results in Section 5 and conclude with a discussion and potential future research directions in Section 6.

2. RELATED WORK

End-to-end models. The landscape of generative models for music has undergone transformative advancements in recent years. End-to-end Autoregressive Models (AMs) have traditionally been at the forefront of sound fidelity, diversity, and long-term coherence [10, 11]. Nonetheless, their high computational demands render AMs unsuitable for music production settings (i.e., sample rate ≥ 44.1 kHz, stereo). In contrast, Generative Adversarial Networks [12]

and Variational Autoencoders [13] exhibit exceptional generation speed at high sampling rates [14–16], positioning them as valuable assets for commercial music production technologies [14, 17]. However, these strategies typically require simple datasets with reduced diversity [14, 15], and often restrict generation to fixed lengths [16, 17]. Recently, diffusion models showed a balanced equilibrium between generation quality, diversity, and efficiency [18–21]. Nevertheless, these rely on an iterative denoising process that, while faster than AMs, still demands long and heavy computations.

Latent models. To address the challenges inherent in end-to-end modeling, generative models have recently pivoted towards operating on compressed representation spaces learned via autoencoders [3–6, 22]. By doing so, generative systems can allocate representational capacity separately for learning immediate auditory characteristics of sound and longer-term music structure. Additionally, they facilitate the interpretation and integration of multi-modal control data, such as text [3, 5, 6], audio [23, 24], or melody [4]. Within this evolved framework, AMs leverage discrete representation spaces crafted through vector-quantized variational autoencoders [25, 26], resulting in faster models with better long-term structure [3, 4]. Recent developments have equipped AMs with parallel decoding using masked token modeling techniques [23, 27, 28], enabling sample rates as high as 44.1 kHz with acceptable inference speed.

Latent Diffusion Models (LDMs) also operate on compressed representation spaces, which are typically continuous [5, 6, 29]. This evolution has catalyzed the development of various LDMs capable of generating high-resolution musical audio with long-term structure [5, 6, 29, 30]. Notably, some works can generate audio at sampling rates as high as 48 kHz [6] and stereo [29, 30]. Other works like Stable Audio [5] improve inference efficiency, enabling the generation of 44.1 kHz sampling rate and stereo audio at an unprecedented speed.² Following this spirit, our system leverages a pre-trained Consistency Autoencoder [8] which enables Diff-A-Riff to function within a highly compressed representation space, allowing faster generation than previous systems. Further, our LDM employs the framework of Elucidated Diffusion Models (EDMs) [9, 20], a departure from the Denoising Diffusion Implicit Models (DDIMs) [31] used in previous approaches [5, 6, 29].

Control mechanisms. As evidenced by the state-of-the-art, text prompts currently serve as the most common interface for users to guide audio generative models [3–6]. To facilitate finer control, Jukedrummer [32] and Music ControlNet [33] utilize time-varying controls such as rhythmic and dynamic envelopes and melodic lines. By exploiting the semantic properties of multi-modal text-and-audio spaces, recent works propose zero-shot solutions to music editing via latent space manipulations [34] and inversion methods [35]. Alternative approaches to control pretrained models include inference-time optimization [36] or guidance [37]. Another method for influencing audio output involves conditioning on audio signals, a technique pri-

marily used in style transfer and accompaniment tasks. In style transfer, the objective is to emulate specific aspects of the source audio, e.g., melody [4], timbre [24]. For accompaniment, the focus is on generating musical content that complements or enhances the conditioning audio [23, 24, 38–40]. Recent works attempting joint music generation and source separation also exhibit compositional capabilities such as accompaniment generation without requiring paired data [41, 42]. Inspired by these control mechanisms, our system introduces conditioning on audio and textual features derived from CLAP [7] alongside audio signals that serve as music context, widening the scope of generative capabilities, e.g., accompaniment generation, text-driven generation, and style transfer.

3. BACKGROUND

In this section, we provide a brief overview of Consistency Models and Denoising Diffusion Models. For an in-depth explanation, we encourage the reader to review the corresponding references.

Consistency Models (CMs) [43, 44] are a novel class of generative models that can produce high-quality samples in a single forward pass without adversarial training. CMs learn a mapping between noisy and clean data samples via a probability flow Ordinary Differential Equation (ODE) [31]. Given a noise level t , the consistency function f transforms a noisy sample $x_t \sim p_t(x)$ to a clean sample $x \sim p_{data}(x)$ by mapping $f(x_t, t) \mapsto x$. This consistency function is approximated by a neural network $f_\theta(x_t, t)$ with parameters θ . It must satisfy the boundary condition $f_\theta(x, t_{\min}) = x$ and is trained by minimizing the discrepancy between its output and a teacher CM at adjacent noise levels t_i and t_{i+1} .

Denoising Diffusion Models (DDMs) [45] are generative models originally inspired by the concept of thermodynamic diffusion [46]. DDMs first add noise to data in a *forward* diffusion process and then use a neural network to *reverse* this process by removing the noise iteratively. The forward diffusion process is detailed by a Stochastic Differential Equation (SDE), introducing noise to the original data x_0 over T steps, resulting in a noisy version x_T . This process is defined by $dx_t = f(x_t, t)dt + g(t)dB_t$. Here, dB_t is the increment of a Wiener process (the random noise), $f(x_t, t)$ is the drift term, $g(t)$ is the diffusion term, and t represents the diffusion time step. The reverse process aims to reconstruct the original data from its noisy version by removing the noise. This is achieved by modeling the score of the data distribution, i.e., the gradient of the log probability density function of the noisy data with respect to the data itself, $\nabla_x \log p(x|t)$. The reverse process is described by another SDE, which guides the denoising $dx_t = [f(x_t, t) - g(t)^2 \nabla_x \log p(x_t|t)]dt + g(t)dB_t$, where a neural network g_θ , with parameters θ , is trained to estimate this score function, i.e., $g_\theta(x_t, t) \approx \nabla_x \log p(x_t|t)$. During inference, by performing this process iteratively, we can progressively transform pure noise inputs into data points following the training data distribution.

² 95 seconds of audio in 8 seconds on an A100 GPU

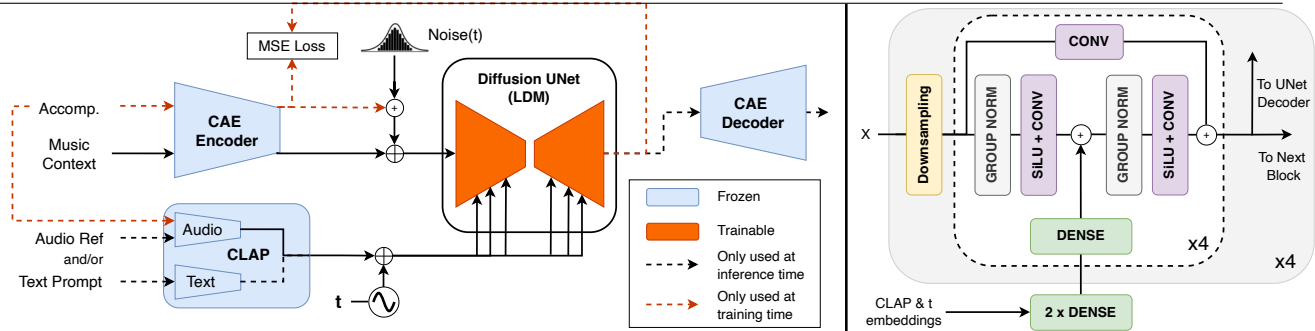


Figure 1. Overview of Diff-A-Riff. **Left:** The CAE Encoder transforms the music context into a compressed representation, concatenated with a noisy sample, and further processed through a multi-scale U-Net. At each scale, conditional CLAP and time-step embeddings are integrated through a feature-wise linear transformation. The generated latent sequence is decoded via the CAE Decoder. We highlight frozen components in blue and trainable elements in orange. Text prompting is only used at inference. **Right:** The encoder architecture comprises four down-sampling blocks with four convolutional and group norm layers with skip connections. The decoder mirrors this architecture.

4. METHODOLOGY

4.1 Dataset

We train our model on a proprietary dataset comprising 12,000 multi-track recordings of diverse music genres (e.g., pop/rock, R&B, rap, country). Each multi-track has various instrument tracks, including bass, guitars, pianos, vocals, and more. We resample each track to 48 kHz, convert it to mono and segment it into overlapping windows of approximately 10 seconds with a 3-second hop size. In training, we randomly select a target *accompaniment* track (excluding vocals) and construct the music *context* by mixing a random subset of the remaining tracks for each segment. We apply this data segmentation and sampling strategy offline to obtain 1M training pairs of audio segments. Following the same methodology, we derive a validation set from 1,200 multi-tracks.

4.2 Diff-A-Riff

4.2.1 Consistency Autoencoder

In this work, we employ a consistency model-based Autoencoder (CAE). We use it pre-trained and freeze its parameters to train a generative model on its latent embeddings (see Fig. 1). The CAE encodes audio samples into a continuous representation space with a $64\times$ compression ratio. It operates on complex Short-Time Fourier Transform (STFT) spectrograms, with real and imaginary components as separate channels. The architecture uses convolutional residual blocks interleaved with down/up-sampling layers. The *CAE Encoder* produces 64-dimensional encodings in the range $(-1, 1)$ with a sample rate of 12 Hz for 48 kHz input audio. The model has ~ 58 million parameters and is trained following the consistency training framework [44]. For a detailed description of the architecture and training procedure, we refer the reader to the original reference [8].

4.2.2 Latent Diffusion Model

We train a Latent Diffusion Model (LDM) on the latent space learned by the CAE. The proposed LDM follows

the framework of Elucidated Diffusion Models (EDMs) [9], a departure from DDIMs [31] for improved model parametrization and inference. The architecture follows DDPM++ [47], an upgraded version of the originally proposed Diffusion Probabilistic Model [45]. We only adapt the network’s input dimensionality to that of the CAE’s latent space (64 channels, see Sec. 4.2.1). Also, we increase the dimensionality of the conditional embedding input with that of CLAP [7] (i.e., 512 dimensions). Our UNet is composed of four down/up-sampling blocks with convolutional layers and skip connections, both for the encoder and the decoder (see Fig. 1 Right). Self-attention is employed in the penultimate resolution layer. We use 512 base channels and double their number at each resolution block. Additionally, the model relies on two dense layers to project the concatenation of CLAP embeddings and the sinusoidal denoising step embeddings into a joint representation. The resulting embedding is used in all down/up-sampling blocks to condition the denoising process as illustrated in Fig. 1.

4.2.3 Training

Fig. 1 provides a high-level overview of Diff-A-Riff’s setup. Given a pair of input *context* and target *accompaniment* audio segments, the model is trained to reconstruct the *accompaniment* given the *context* and a CLAP embedding derived from a randomly selected sub-segment of the target itself. This prevents the model from relying on CLAP for temporal alignment. In order to use Classifier-Free Guidance (see Sec. 4.3.1) and allow the model to optionally operate unconditionally, we drop the audio context and clap embeddings both with a 50% probability. We train Diff-A-Riff over 1M iterations using a batch size of 256 (2 weeks on a single RTX 3090 GPU). We use AdamW [48] as the optimizer and a base learning rate of 10^{-4} . We use a learning rate schedule with an initial warm-up phase and a reduce-in-plateau process that decreases the learning rate to a minimum value of 10^{-6} . We keep an Exponential Moving Average (EMA) on the weights with a momentum of 0.9999. The resulting model has 500M parameters (including the CAE and not CLAP) and occupies 3 GB of memory.

4.3 Evaluation

Objective comparison of Diff-A-Riff with existing state-of-the-art models [4–6] is challenging as these are generally trained to perform a substantially different task (generation of fully mixed music with no accompaniment conditioning). Even though StemGen [23] and SingSong [39] are trained to generate accompaniments, their implementation and pretrained weights are not publicly available. Therefore, we focus on subjective listening tests to compare with available music generation models. Additionally, we perform objective evaluations to analyze different inference parameters (e.g., number of diffusion steps, conditioning information; see Section 4.3.1) to understand which configurations of our proposed model perform the best on our set of metrics. We then apply the gained insights in order to generate the samples that are proposed in the user studies. In the following sections, we describe how we generate the samples used for evaluation, the objective metrics, the listening test methodology, and the baselines we compare against.

4.3.1 Inference Configurations

In this section, we describe the inference configurations that we use for the objective and subjective evaluations.

Conditioning Signals: Different conditioning signals are evaluated. $CLAP_A$ refers to the audio-derived CLAP embeddings, which are obtained by using CLAP to encode real audio from a track of the evaluation set. We can also condition the model on text-derived CLAP embeddings, despite them never being fed during training to the model, since CLAP offers a joint embedding space for both modalities. Because our dataset does not contain audio/text pairs, we create $CLAP_T$ embeddings by asking ChatGPT to write text descriptions of single-stem tracks. Finally, *Context* refers to the conditioning signal obtained by solely encoding the music context into the CAE Encoder.

Classifier-Free Guidance (CFG) [49] allows to improve generation quality by increasing the influence of conditioning signals in the sampling process. Given the guidance strength CFG, we implement guidance as $x_{t-1} = f_{\theta}(x_t) + \text{CFG} \cdot (f_{\theta}(x_t, c) - f_{\theta}(x_t))$. At inference time, we can use different guidance strengths for *Context* and $CLAP$ embeddings, denoted as $\text{CFG}_{\text{Context}}$ and CFG_{CLAP} , respectively.

Number of Diffusion Steps: At inference time, the number of denoising steps T allows to trade between audio quality and generation speed.

Pseudo-Stereo Generation: We generate pseudo-stereo audio by denoising until a given diffusion time step, and then by independently concluding a stochastic denoising process twice, one for each audio channel. We define the *stereo width* as the proportion of denoising steps used for stereo generation over the total number of steps. In the user study, we set this parameter to 0.4.

4.3.2 Objective Metrics

We evaluate Diff-A-Riff through objective metrics to assess various aspects of the generated audio. These include the standard *Squared Maximum Mean Discrepancy* (MMD2) [50] and *Fréchet Audio Distance* (FAD) [51] for audio quality as well as *Density* and *Coverage* [52] for evaluating fidelity and diversity. To study the system’s responsiveness to text prompts, we employ the *Clap Score* (CS) [53], which calculates the cosine similarity between text and audio embeddings. In order to evaluate the alignment of the generated accompaniment with the context, we employ the *Audio Prompt Adherence* (APA) [54], a metric based on FAD tailored to evaluate accompaniment systems. All metrics are calculated by averaging five batches of 500 candidate samples. We use CLAP [7] as the embedding space for metrics that compare distributions (like MMD2 and FAD) using a reference set of 5,000 real audio examples.

4.3.3 Listening Tests

Subjective Audio Quality (SAQ): We perform a Mean Opinion Score (MOS) test to assess audio quality. Participants were presented with 5-second audio segments from real data as well as generations from the baselines and the proposed system. Their task is to rate the audio quality of these segments on a 5-level Likert scale ranging from poor (1) to excellent quality (5). For all items (real data, Diff-A-Riff, and baselines), we compare both complete music pieces as well as solo instruments.

We generate solo instruments with Diff-A-Riff by conditioning the model on text or audio-derived CLAP embeddings ($CLAP_A$ or T) and without an input context (for a fair comparison with the baselines, which do not rely on contextual audio inputs). Despite the model not being trained for this task, we can also generate *complete* music pieces using $CLAP$ and *Context* embeddings, following an iterative approach: First, we create sets of $CLAP_A$ or T embeddings as described in 4.3.1. Then, from an initially empty context, we generate new tracks from those $CLAP$ embeddings, iteratively summing the resulting generation into the input context from which we derive the next *Context* embeddings.

We compare Diff-A-Riff against three state-of-the-art text-to-music baselines: AudioLDM2 [6], MusicGen [4], and Stable Audio [5].³ For each baseline, we generate 20 5-second excerpts of complete music and solo instruments using text prompts generated by ChatGPT.

Subjective Audio Prompt Adherence (SAPA): We also conduct a subjective assessment of audio-prompt adherence. Participants are provided with a reference 10-second music segment and are asked to rate the compatibility of five distinct accompaniments on a scale from 0 (indicating no adherence) to 100 (perfect adherence), according to harmonic, rhythmic, and overall music style compatibility.

³ We use the open-source AudioLDM2 model ‘AudioLDM2-48kHz’ operating at 48 kHz. For MusicGen, we use the open-source model ‘MusicGen-large’ operating at 32 kHz, and for Stable Audio, we use their public API.

	Cond. Signal	↓ MMD2 ^a	↓ FAD	↑ Coverage	↑ Density	↑ APA	↑ CS
<i>Real</i>	Original acc.	0.00	0.02	0.18	1.03	0.93	1.00
<i>Lower bound</i>	-	64.32 ^b	1.67 ^b	0.00 ^b	0.00 ^b	0.11 ^c	-0.07 ^c
<i>Diff-A-Riff</i> _{T=30} ^{CFG=1.25}	<i>CLAP</i> _A + <i>Context</i>	0.22	0.03	0.17	1.00	0.92	-
	<i>CLAP</i> _T + <i>Context</i>	3.96	0.17	0.05	0.32	0.20	0.25
	<i>Context</i> only	4.87	0.24	0.05	0.37	0.23	-
	<i>CLAP</i> _A only	0.36	0.03	0.14	0.84	-	-
	<i>CLAP</i> _T only	4.38	0.20	0.05	0.41	-	0.24
	No Conditioning	6.70	0.27	0.03	0.25	-	-
<i>Diff-A-Riff</i> _{T=10} ^{CFG=1}	<i>CLAP</i> _A + <i>Context</i>	1.50	0.06	0.09	0.54	0.54	-
	<i>CLAP</i> _T + <i>Context</i>	6.23	0.19	0.03	0.17	0.00	0.23
	<i>Context</i> only	6.59	0.25	0.03	0.24	0.00	-
	<i>CLAP</i> _A only	1.57	0.06	0.09	0.52	-	-
	<i>CLAP</i> _T only	6.26	0.22	0.03	0.20	-	0.22
	No Conditioning	7.67	0.28	0.03	0.20	-	-

^a × 10⁻⁴, ^b obtained from white noise, ^c obtained by using a random accompaniment from the dataset

Table 1. Objective metrics using two configurations, *Diff-A-Riff*_{T=30}^{CFG=1.25} and *Diff-A-Riff*_{T=10}^{CFG=1}, and different conditional settings (see Sec. 4.3.1). We compare against higher bounds obtained from the *real* validation set, and lower bounds obtained from random *noise* or random pairs (*Real*, *Random acc.*). Some cells are empty for APA and CS in the case of context and text-free generation respectively.

The reference segments are derived from music pieces within the evaluation set by summing all tracks in a multitrack, excluding one track, which is reserved to serve as the original accompaniment. Each accompaniment is presented mixed with the reference segment, with slight panning applied to the right to aid in distinguishing between them. The five accompaniments include the original accompaniment, a randomly selected one from the evaluation set, and three generated by our model under different conditional setups: (*CLAP*_A + *Context*), (*CLAP*_T + *Context*) and (*Context* only). To remove a potential bias toward better-quality audio, the original and random segments are encoded and decoded through the CAE.

For both studies, we used the GoListen platform [55]. All audio segments are normalized to a loudness of -20 dB LUFS and not cherry-picked. Sample questions are available on the accompanying website.

5. RESULTS & DISCUSSION

5.1 Objective Evaluation

Fig. 2 shows the MMD2 score of our model as a function of the number of denoising steps, with each line corresponding to a different conditional setting (see Sec. 4.3.1), all without classifier-free guidance (CFG). Note that MMD2 compares the distributions of embeddings of generated and real audio in CLAP’s latent space. This means it indicates not only audio quality but also how well the distributions of generated instrument types and timbres match the test data distribution. For this reason, when using audio CLAP conditioning (*CLAP*_A), the results are considerably better, as we force the timbre distribution to be equal to the test data distribution (by using embeddings of that distribution as conditioning). However, the improvement of the results when increasing the number of denoising steps can be con-

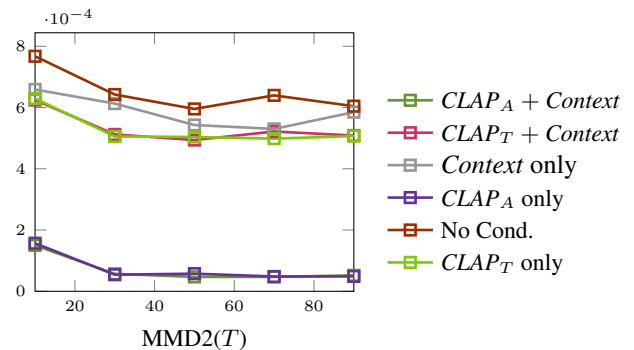


Figure 2. MMD2 as a function of the number of denoising steps *T* for various conditional settings (see Sec. 5.1).

sidered independent of the timbre distribution.

Based on the results described above we perform a grid search over diffusion steps (*T*) and multi-source classifier-free guidance strength ($CFG_{<source>}$). $T = 30$ steps and $CFG_{Context} = CFG_{CLAP} = 1.25$ yields the best results. We denote this configuration as *Diff-A-Riff*_{T=30}^{CFG=1.25} in Tab. 1. In addition, we compare to a specific configuration that achieves real-time performance⁴ on a CPU with acceptable quality using $T = 10$ steps and $CFG = 1$, denoted as *Diff-A-Riff*_{T=10}^{CFG=1}. For reference, we also include metrics computed on *real* data and a *lower bound*, calculated from white noise in the case of quality metrics or random real pairs for input adherence metrics (APA and CS). The overall trend suggests that dense conditioning information helps the model in all metrics: audio quality, *coverage* and *density*, as well as APA. Also, the results could suggest a slight dependency of the model on *CLAP*_A embeddings, with metrics close to real data, even in the absence of music context. Only *density* exhibits a minor drop without con-

⁴ 95 seconds of audio in 73 seconds on an AMD EPYC 7502P

text conditioning, suggesting that generating from a silent context leads to samples that sometimes fall in low-density regions of the CLAP space.

The impact of Classifier-free Guidance (CFG) on MMD2 can be estimated by comparing Fig. 2, that displays results without CFG, with Tab. 1, where CFG was used. In Fig. 2, the metrics for both ($CLAP_A + Context$) and ($CLAP_A$ only) converge towards an MMD2 of 0.5, while the corresponding values in Tab. 1 show a reduction to about half this figure. For ($CLAP_T + Context$) and ($CLAP_T$ only), the MMD2 drops from approximately 5 to about 4 with CFG, and for *Context only*, it decreases from around 6 to approximately 5.

Finally, we calculate the Clap Score (CS) for text-conditioned generation ($CLAP_T + Context$, $CLAP_T$ only). We compare $Diff\text{-}A\text{-}Riff_{T=30}^{CFG=1.25}$ against random pairs of text and real audio (CS=-0.07), suggesting that the model is only somewhat responsive to text prompts (CS=0.25).

5.2 Subjective Evaluation

Table 2 presents the outcomes of the Subjective Audio Quality (SAQ) test based on 74 users who each rated 32 audio segments, resulting in 2368 ratings. In this test, we compare results against leading baselines (see Sec. 4.3.3) and *real* audio data. Our analysis includes a comparison between these benchmarks and the audio generated by Diff-A-Riff in both mono (ch=1) and pseudo-stereo (ch=2) formats (see Sec. 4.3.1). Results show that the pseudo-stereo samples generated by Diff-A-Riff received ratings that are statistically indifferent from *real* audio ratings (p-value=0.79), indicating that participants found the audio quality of the generations indistinguishable from real data. This outcome is particularly remarkable given that Diff-A-Riff was not explicitly trained on complete musical pieces nor stereo music generation, but is still competitive to other models. Further, it highlights the influence of stereo imaging on the perceived audio quality.

Tab. 3 shows the results for the Subjective Audio Prompt Adherence (SAPA) listening test based on 35 users, each rating 25 accompaniments. The results include ratings scored by *real* accompaniments, *random* accompaniments, and the various conditional settings described in Sec. 4.3.1. Following the trend of previous results, the default setting ($CLAP_A + Context$) scores the closest to *real* accompaniments, suggesting that the model can effectively adapt to the context under this setting. When conditioned on (*Context only*), Diff-A-Riff is rated worse but still significantly better than *random* accompaniments. Further, for ($CLAP_T + Context$), the accompaniments are rated the lowest. A reason could be that the overall quality is worse because CLAP embeddings of text prompts have not been shown during training. Another problem could be that the randomly chosen text prompt is incompatible with the provided music context (e.g., "A drum machine with electronic textures" with an acoustic blues context), which reduces perceived adherence due to conflicting styles.

Overall, SAPA results are interesting given that APA (see Tab. 1) suggested rather pessimistic results for

	SR/Ch	Params	RTF ^a	Solo	Songs
Real data	44.1/2	-	-	3.5 ± 0.2	3.8 ± 0.2
MusicGen	32/1	3.3B	0.4	3.1 ± 0.2	3.2 ± 0.2
StableAudio	44.1/2	1B	11.8	2.5 ± 0.2	3.0 ± 0.2
AudioLDM2	48/1	712M	0.4	2.6 ± 0.2	2.0 ± 0.2
Diff-A-Riff	48/2 48/1	500M	13.5 (0.57) 19 (1.3)	3.4 ± 0.1 2.8 ± 0.1	3.8 ± 0.1 3.2 ± 0.1

^aNVIDIA A100 (CPU : AMD EPYC 7502P)

Table 2. Comparison of Diff-A-Riff to baselines. We include sampling rate in kHz and number of channels (*SR/Ch*), the total number of parameters *Params* (without CLAP), the Real Time Factor (*RTF*, the ratio of generated time over inference time, for 95 second-long audios) on GPU (and CPU for our model), as well as the SAQ32 values and 95% confidence intervals for the subjective audio quality assessment of *Solo* instruments and complete *Songs*.

	Cond. Signal	SAPA
<i>Real</i>	-	70.1 ± 4.5
<i>Random</i>	-	12.3 ± 3.0
Diff-A-Riff	$CLAP_A + Context$	62.4 ± 4.4
	$CLAP_T + Context$	37.6 ± 4.2
	<i>Context only</i>	42.3 ± 4.3

Table 3. Results for SAPA (see Sec. 4.3.3). The table includes results of Diff-A-Riff using different conditional settings, with 95% confidence intervals.

($CLAP_T + Context$) and (*Context only*). This could potentially be attributed to APA's sensitivity to audio quality and timbre differences between reference and candidate sets.

5.3 Control Mechanisms

In the accompanying website, we show examples of *Diff-A-Riff* generations for different inference settings (see Sec. 4.3.1). We also showcase other controls that naturally emerge from the denoising process, such as *in/out-painting* or the generation of *variations* and *loops*, as well as controls derived from the manipulation of CLAP embeddings, e.g., text-audio *Interpolations*.

6. CONCLUSION

This work introduced Diff-A-Riff, a Latent Diffusion Model capable of generating instrumental accompaniments adapted to a user-provided musical audio context. It can be controlled based on style audio references, text prompts, or both. We also proposed a simple method for producing pseudo-stereo audio. By exploiting the efficiency of a Consistency Autoencoder, Diff-A-Riff can generate 48 kHz sample rate pseudo-stereo audio with unprecedented speed and quality. Through extensive objective and subjective evaluation, we showed that our model achieves state-of-the-art audio quality, adapts to various conditional settings, and generates content that adheres to pre-existing musical audio contexts. We believe this work represents a significant step towards AI-assisted music production tools that prioritize artist-centric interactions, enriching the landscape of human-machine music co-creation.

7. ETHICS STATEMENT

Sony Computer Science Laboratories is committed to exploring the positive applications of AI in music creation. We collaborate with artists to develop innovative technologies that enhance creativity. We uphold strong ethical standards and actively engage with the music community and industry to align our practices with societal values. Our team is mindful of the extensive work that songwriters and recording artists dedicate to their craft. Our technology must respect, protect, and honour this commitment.

Diff-A-Riff supports and enhances human creativity and emphasises the artist’s agency by providing various controls for generating and manipulating musical material. By generating a stem at a time, the artist remains responsible for the entire musical arrangement.

Diff-A-Riff has been trained on a dataset that was legally acquired for internal research and development; therefore, neither the data nor the model can be made publicly available. We are doing our best to ensure full legal compliance and address all ethical concerns.

8. REFERENCES

- [1] U. Singer, A. Polyak, T. Hayes, X. Yin, J. An, S. Zhang, Q. Hu, H. Yang, O. Ashual, O. Gafni, D. Parikh, S. Gupta, and Y. Taigman, “Make-A-Video: Text-to-video generation without text-video data,” in *Proc. of the 11th International Conference on Learning Representations, ICLR, 2023*.
- [2] J. Betker, G. Goh, L. Jing, TimBrooks, J. Wang, L. Li, LongOuyang, JuntangZhuang, JoyceLee, YufeiGuo, WesamManassra, PrafullaDhariwal, CaseyChu, YunxinJiao, and A. Ramesh, “Improving image generation with better captions,” in *CoRR, 2023*.
- [3] A. Agostinelli, T. I. Denk, Z. Borsos, J. H. Engel, M. Verzetti, A. Caillon, Q. Huang, A. Jansen, A. Roberts, M. Tagliasacchi, M. Sharifi, N. Zeghidour, and C. H. Frank, “MusicLM: Generating Music From Text,” in *CoRR, 2023*.
- [4] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Défossez, “Simple and controllable music generation,” in *Advances in Neural Information Processing Systems 36 NeurIPS, 2023*.
- [5] Z. Evans, C. Carr, J. Taylor, S. H. Hawley, and J. Pons, “Fast timing-conditioned latent audio diffusion,” in *CoRR, 2024*.
- [6] H. Liu, Q. Tian, Y. Yuan, X. Liu, X. Mei, Q. Kong, Y. Wang, W. Wang, Y. Wang, and M. D. Plumbley, “AudioLDM 2: Learning holistic audio generation with self-supervised pretraining,” in *arXiv, 2023*.
- [7] B. Elizalde, S. Deshmukh, M. A. Ismail, and H. Wang, “CLAP learning audio concepts from natural language supervision,” in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP, 2023*.
- [8] M. Pasini, S. Lattner, and G. Fazekas, “Music2latent: Consistency autoencoders for latent audio compression,” in *Proc. of the International Society for Music Information Retrieval (ISMIR), 2024*.
- [9] T. Karras, M. Aittala, T. Aila, and S. Laine, “Elucidating the design space of diffusion-based generative models,” in *Proc. of the 36th Conference on Neural Information Processing Systems (NeurIPS), 2022*.
- [10] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu, “WaveNet: A generative model for raw audio,” in *Proc. of the 9th ISCA Speech Synthesis Workshop, 2016*.
- [11] S. Mehri, K. Kumar, I. Gulrajani, R. Kumar, S. Jain, J. Sotelo, A. C. Courville, and Y. Bengio, “SampleRNN: An unconditional end-to-end neural audio generation model,” in *Proc. of 5th International Conference on Learning Representations, ICLR, 2017*.
- [12] I. J. Goodfellow, J. Pouget-Abadie *et al.*, “Generative adversarial nets,” in *Advances in Neural Information Processing Systems 27*, Dec. 2014, pp. 2672–2680.
- [13] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” in *2nd International Conference on Learning Representations (ICLR), Apr. 2014*.
- [14] A. Caillon and P. Esling, “RAVE: A variational autoencoder for fast and high-quality neural audio synthesis,” in *CoRR, 2021*.
- [15] M. Pasini and J. Schlüter, “Musika! Fast infinite waveform music generation,” in *Proc. of the 23rd International Society for Music Information Retrieval Conference, ISMIR, 2022*.
- [16] J. Nistal, S. Lattner, and G. Richard, “DrumGAN: Synthesis of drum sounds with timbral feature conditioning,” in *Proc. of the 21st International Society for Music Information Retrieval Conference, ISMIR, 2020*.
- [17] J. Nistal, C. Aouameur, I. Velarde, and S. Lattner, “DrumGAN VST: A plugin for drum sound analysis/synthesis with autoencoding generative adversarial networks,” in *Proc. of International Conference on Machine Learning ICML, Workshop on Machine Learning for Audio Synthesis, MLAS, 2022*.
- [18] S. Rouard and G. Hadjeres, “CRASH: raw audio score-based generative modeling for controllable high-resolution drum sound synthesis,” in *Proc. of the 22nd International Society for Music Information Retrieval Conference, ISMIR, 2021*.
- [19] Q. Huang, D. S. Park, T. Wang, T. I. Denk, A. Ly, N. Chen, Z. Zhang, Z. Zhang, J. Yu, C. H. Frank, J. H. Engel, Q. V. Le, W. Chan, and W. Han, “Noise2Music: Text-conditioned music generation with diffusion models,” in *CoRR, 2023*.

- [20] G. Zhu, Y. Wen, M. Carboneau, and Z. Duan, “EDM-Sound: Spectrogram based diffusion models for efficient and high quality audio synthesis,” in *CoRR*, 2023.
- [21] F. Schneider, “ArchiSound: Audio generation with diffusion,” in *CoRR*, 2023.
- [22] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, and I. Sutskever, “Jukebox: A generative model for music,” in *CoRR*, 2020.
- [23] J. D. Parker, J. Spijkervet, K. Kosta, F. Yesiler, B. Kuznetsov, J. Wang, M. Avent, J. Chen, and D. Le, “StemGen: A music generation model that listens,” in *CoRR*, 2023.
- [24] M. Pasini, M. Grachten, and S. Lattner, “Bass accompaniment generation via latent diffusion,” in *IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP*. IEEE, 2024.
- [25] N. Zeghidour, A. Luebs, A. Omran, J. Skoglund, and M. Tagliasacchi, “SoundStream: An end-to-end neural audio codec,” in *IEEE ACM Trans. Audio Speech Lang. Process.*, 2022.
- [26] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, “High fidelity neural audio compression,” in *CoRR*, 2022.
- [27] H. F. García, P. Seetharaman, R. Kumar, and B. Pardo, “VampNet: Music generation via masked acoustic token modeling,” in *Proc. of the 24th International Society for Music Information Retrieval Conference, ISMIR*, 2023.
- [28] A. Ziv, I. Gat, G. L. Lan, T. Remez, F. Kreuk, A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, “Masked audio generation using a single non-autoregressive transformer,” in *CoRR*, 2024.
- [29] F. Schneider, O. Kamal, Z. Jin, and B. Schölkopf, “Moûsai: Text-to-music generation with long-context latent diffusion,” in *CoRR*, 2023.
- [30] P. Li, B. Chen, Y. Yao, Y. Wang, A. Wang, and A. Wang, “JEN-1: text-guided universal music generation with omnidirectional,” in *CoRR*, 2023.
- [31] J. Song, C. Meng, and S. Ermon, “Denoising diffusion implicit models,” in *Proc. of the 9th International Conference on Learning Representations, ICLR*, 2021.
- [32] Y. Wu, C. Chiu, and Y. Yang, “Jukedrummer: Conditional beat-aware audio-domain drum accompaniment generation via transformer VQ-VAE,” in *Proc. of the 3rd International Society for Music Information Retrieval Conference, ISMIR*, 2022.
- [33] S.-L. Wu, C. Donahue, S. Watanabe, and N. J. Bryan, “Music controlnet: Multiple time-varying controls for music generation,” in *CoRR*, 2023.
- [34] Y. Zhang, Y. Ikemiya, G. Xia, N. Murata, M. A. M. Ramírez, W. Liao, Y. Mitsufuji, and S. Dixon, “Musicmagus: Zero-shot text-to-music editing via diffusion models,” *CoRR*, 2024.
- [35] H. Manor and T. Michaeli, “Zero-shot unsupervised and text-based audio editing using DDPM inversion,” *CoRR*, 2024.
- [36] Z. Novack, J. J. McAuley, T. Berg-Kirkpatrick, and N. J. Bryan, “DITTO: diffusion inference-time t-optimization for music generation,” *CoRR*, 2024.
- [37] M. Levy, B. D. Giorgi, F. Weers, A. Katharopoulos, and T. Nickson, “Controllable music production with diffusion models and guidance gradients,” *CoRR*, 2023.
- [38] M. Grachten, S. Lattner, and E. Deruty, “BassNet: A variational gated autoencoder for conditional generation of bass guitar tracks with learned interactive control,” in *Applied Sciences*, 2020.
- [39] C. Donahue, A. Caillon, A. Roberts, E. Manilow, P. Esling, A. Agostinelli, M. Verzetti, I. Simon, O. Pietquin, N. Zeghidour, and J. H. Engel, “Singsong: Generating musical accompaniments from singing,” in *CoRR*, 2023.
- [40] S. Lattner and M. Grachten, “High-level control of drum track generation using learned patterns of rhythmic interaction,” in *2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, WASPAA*. IEEE, 2019.
- [41] E. Postolache, G. Mariani, L. Cosmo, E. Benetos, and E. Rodolà, “Generalized multi-source inference for text conditioned music diffusion models,” *CoRR*, 2024.
- [42] G. Mariani, I. Tallini, E. Postolache, M. Mancusi, L. Cosmo, and E. Rodolà, “Multi-source diffusion models for simultaneous music generation and separation,” *CoRR*.
- [43] Y. Song, P. Dhariwal, M. Chen, and I. Sutskever, “Consistency models,” in *International Conference on Machine Learning, ICML*, 2023.
- [44] Y. Song and P. Dhariwal, “Improved techniques for training consistency models,” *arXiv preprint arXiv:2310.14189*, 2023.
- [45] J. Ho, A. Jain, and P. Abbeel, “Denoising diffusion probabilistic models,” in *Advances in Neural Information Processing Systems, NeurIPS*, 2020.
- [46] J. Sohl-Dickstein, E. A. Weiss *et al.*, “Deep unsupervised learning using nonequilibrium thermodynamics,” in *Proceedings of the 32nd International Conference on Machine Learning, ICML*, 2015.
- [47] T. Karras, M. Aittala, T. Aila, and S. Laine, “Score-based generative modeling through stochastic differential equations,” in *Proc. of the International Conference on Learning Representations (ICLR)*, 2021.

- [48] I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” in *7th International Conference on Learning Representations, ICLR*, 2019.
- [49] J. Ho and T. Salimans, “Classifier-free diffusion guidance,” *arXiv preprint arXiv:2207.12598*, 2022.
- [50] M. Binkowski, D. J. Sutherland, M. Arbel, and A. Gretton, “Demystifying MMD gans,” in *Proc. of the 6th International Conference on Learning Representations, ICLR*, 2018.
- [51] K. Kilgour, M. Zuluaga, D. Roblek, and M. Sharifi, “Fréchet audio distance: A reference-free metric for evaluating music enhancement algorithms,” in *Proc. of the 20th Conference of the International Speech Communication Association (InterSpeech)*, 2019.
- [52] M. F. Naeem, S. J. Oh, Y. Uh, Y. Choi, and J. Yoo, “Reliable fidelity and diversity metrics for generative models,” in *Proc. of the 37th International Conference on Machine Learning, ICML*, 2020.
- [53] R. Huang, J. Huang, D. Yang, Y. Ren, L. Liu, M. Li, Z. Ye, J. Liu, X. Yin, and Z. Zhao, “Make-an-audio: Text-to-audio generation with prompt-enhanced diffusion models,” in *Proc. of the International Conference on Machine Learning, ICML*, 2023.
- [54] M. Grachten and J. Nistal, “Audio Prompt Adherence: A measure for evaluating musical accompaniment systems,” in *CoRR*, 2024.
- [55] D. Barry, Q. Zhang, P. W. Sun, and A. Hines, “Go listen: An end-to-end online listening test platform.” *Journal of Open Research Software*, vol. 9, no. 1, 2021.

EXPLORING INTERNET RADIO ACROSS THE GLOBE WITH THE MIRAGE ONLINE DASHBOARD

Ngan V.T. Nguyen

Elizabeth A.M. Acosta

Tommy Dang

David R.W. Sears

University of Science, VNU-HCMUS Texas Tech University Texas Tech University Texas Tech University

nvtngan@hcmus.edu.vn

liz.acosta@ttu.edu

tommy.dang@ttu.edu

david.sears@ttu.edu

ABSTRACT

This study presents the *Music Informatics for Radio Across the GlobE* (MIRAGE) online dashboard, which allows users to access, interact with, and export metadata (e.g., artist name, track title) and musicological features (e.g., instrument list, voice type, key/mode) for 1 million events streaming on 10,000 internet radio stations across the globe. Users can search for stations or events according to several criteria, display, analyze, and listen to the selected station/event lists using interactive visualizations that include embedded links to streaming services, and finally export relevant metadata and visualizations for further study.

1. INTRODUCTION

Despite its scholarly neglect relative to television, film, and print [1], radio's convergence with the internet has extended its reach via web browsers and smartphone apps, enabling the medium to persist as a central site of culture and daily life for communities around the world [2,3]. The recent resurgence of pirate and community radio stations on the internet alongside national and multinational networks also reflects internet radio's lower production costs relative to short-wave terrestrial (e.g., FM or AM) radio [4], resulting in a diverse range of both standardized and specialized programming [5–7].

And yet, the volume and scope of much of the research in fields like radio studies has been freighted heavily towards the Global North [1]. In doing so, the research program just described attempts to situate listeners within a particular musical tradition (e.g., western classical or popular music), rather than within a particular geographic environment (e.g., El Paso, Texas) where myriad musical traditions might co-exist. As a result, music's vast global marketplace has yet to receive sustained scholarly attention in the MIR community.

To address this issue, this study presents the development release (v0.2) of the *Music Informatics for Radio Across the GlobE* (MIRAGE) online dashboard, which

allows users with potentially little training in computational methods to access, interact with, and export metadata (e.g., artist name, track title) and musicological features (e.g., instrument list, voice type, key/mode) for 1 million events streaming on 10,000 internet radio stations across the globe. To that end, Section 2 summarizes previous research on the development of digitized music corpora and cultural databases. Next, Section 3 presents the MIRAGE-MetaCorpus, Section 4 introduces the MIRAGE online dashboard, and Section 5 offers a potential use case. Finally, Section 6 discusses limitations and future directions for the MIRAGE project.

2. PREVIOUS RESEARCH

In recent years, researchers in music theory, music information retrieval (MIR), and radio/media studies have developed digitized music corpora and cultural databases that represent data in machine-readable symbolic and audio formats.

In computational music theory, heavily curated corpora (100s of songs) like the McGill Billboard and Rolling Stone-200 data sets include expert annotations for musical parameters like harmony, meter, and melody, for example, but remain restricted to Anglophone popular music traditions [8,9]. What is more, the limited size of symbolic corpora makes comparative research especially difficult [10].

In MIR, corpora like the Million Song data set address issues of scale while avoiding copyright infringement by providing researchers with publicly available metadata and musicological features protected under fair use for a large collection of songs hosted on commercial music-streaming services like last.fm [11]. Nevertheless, the size, scope, and format of these projects require extensive training in distant-reading (i.e., computational) methods [12–14]. As a result, MIR corpora sometimes eschew the kinds of musical engagements favored by scholars in humanities disciplines using close-reading methodologies. Finally, the projects referenced above do not include information about the geographic location of the music encountered by listeners in everyday life.

Finally, in radio/media studies, researchers routinely employ interview and survey methodologies to explore radio stations across the globe [15, 16], in some cases by selecting samples from radio-station directories hosted online [17]. The now defunct ComFM, for example, included a catalogue of web-radio stations classified according to



Continent	Radio Stations	Sovereign States
Africa	392 (4%)	39 (58%)
Asia	653 (7%)	38 (59%)
Europe	5,161 (52%)	49 (60%)
North America	2,243 (22%)	33 (67%)
Oceania	222 (2%)	5 (17%)
South America	1,329 (13%)	13 (87%)
TOTAL	10,000	177

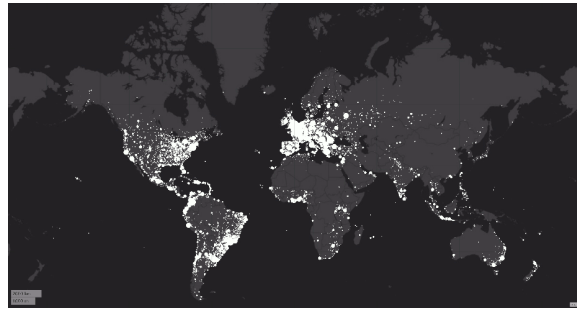


Figure 1. Descriptive statistics (left) and geographic map (right) of the radio stations in MIRAGE-MetaCorpus. The size of each bubble represents the number of stations at that location.

geographic region and type of programming. Other current internet radio directories like radio.co and internet-radio.com offer searchable databases consisting of several thousand stations, but they do not permit users to access or export the entire database for further analysis.

The MIRAGE online dashboard addresses these issues by offering a global archive of the musical traditions encountered on internet radio. For this reason, the dashboard’s database could serve MIR tasks like music recommendation and genre classification, but the dashboard itself also allows researchers with potentially little training in computational methods to select and analyze a subset of events or stations (i.e., to develop their own sub-corpora). Finally, like previous MIR projects [11], the MIRAGE online dashboard avoids copyright infringement by including publicly available metadata and musicological features protected under fair use while enabling users to stream recordings using embedded links to commercial services like Spotify and YouTube.

3. MIRAGE METACORPUS

The core database for the MIRAGE online dashboard is MIRAGE-MetaCorpus, which currently consists of metadata and musicological features for 1 million events that streamed on 10,000 internet radio stations across the globe. In this context, an ‘event’ could represent a musical work of some kind, or a radio program like a podcast or a call-in show.

3.1 Collecting MetaData

Following [12], data collection consisted of three stages: station-list and event-list collection (*Stage 1*), station-list review (*Stage 2*), and event-list parsing (*Stage 3*).

3.1.1 Stage 1: Collecting Station/Event Lists

Toward Stage 1, the research team collected metadata for an initial list of internet radio stations and then monitored the station streams to obtain additional metadata from the stream encoder. To that end, we monitored radio stations in real time on Radio Garden,¹ a streaming service with an open-access application programming interface (API)

that allows users to select and play publicly available radio streams using an interactive representation of the globe.

Between the months October to January 2022-2023, a random sample of 10,000 stations from the initial station list was monitored throughout the 24-hour day – but avoiding each ten-minute period at the top and bottom of the hour when advertising is most frequent – in order to obtain additional metadata from the stream encoder for 100 events from each station, resulting in an initial list of 1 million events. The monitoring algorithm also excluded an event if the stream description did not include metadata, or if the metadata featured advertising terms or reflected a station blackout period (e.g., ‘advert’, ‘commercial’, ‘unknown’, ‘blackout’, etc.).

During event-list collection, additional metadata for each location in the initial station list was also included from the Natural Earth map data set,² which provides public-domain vector and map raster data along with accompanying metadata.

Shown in Figure 1, the selected station list represents 177 of the globe’s 305 sovereign states. As a random sample of Radio Garden’s station list (i.e., the *Radio Garden sample*), this release of the MIRAGE-MetaCorpus (v0.2) therefore reflects the prevalence of internet radio stations across the globe on the Radio Garden streaming service.

3.1.2 Stage 2: Reviewing Stations

Toward Stage 2, a team of six human annotators began reviewing station-level metadata from the Radio Garden API and radio-station stream encoder in 2023-2024. For each station, an annotator reviewed the station’s website url, station name, city, and country for incorrect/missing spelling, capitalization, punctuation, and diacritics. Next, the list of genres, formats, and terrestrial (FM/AM) station frequencies (if applicable) were reviewed and/or included using information on the station website. Finally, the annotator reviewed the corresponding event list for each station to determine the percentage of events that featured reliable stream-description metadata (i.e., artist name, track title).

Currently, the research team has reviewed over 6,000 stations and plans to complete station-list review by 2025.

¹ <https://radio.garden>.

² <https://www.naturalearthdata.com/>.

<location>	
<city> ^{ab}	Johor Bahru
<country> ^{ab}	Malaysia
<country_GDP> ^b	863 Billion
<coordinates> ^{ab}	103.6545°, 1.4783°
<station>	
<name> ^{cd}	Best FM
<form> ^{cd}	Simulcast (FM 104.1)
<format> ^d	Adult Contemporary
<genre> ^{cd}	pop, Indonesian pop
<website> ^{cd}	http://www.bestfm.com.my
<event>	
<time@station> ^c	12/28/2022 9:37
<description> ^c	Aisha Retno – Sutera
<reliability> ^e	1
<artist>	
<name> ^f	Aisha Retno
<type> ^f	musical artist
<gender> ^f	female
<country> ^f	Malaysia
<genre> ^f	pop
<instruments> ^f	piano, voice
<track>	
<title> ^f	Sutera
<duration> ^f	03:18
<year_released> ^f	2022
<key> ^f	C minor
<language> ^f	Malay

Table 1. Left: Selected variables from the encoding scheme for MIRAGE-MetaCorpus, expressed in pseudocode. Metadata were obtained from the following sources: ^a Radio Garden API; ^b Natural Earth map data set; ^c Internet Radio Station Stream Encoder; ^d Annotator Review; ^e Monitoring/Matching Algorithm; ^f Online Music Libraries. Right: An example of the metadata for an event in MIRAGE-MetaCorpus.

3.1.3 Stage 3: Parsing Events

Toward Stage 3, additional metadata were collected for each event using the Spotify and WikiData online music libraries.³ Specifically, the team queried each API using each event’s stream description. The obtained list of matching queries was then filtered using a normalized edit distance measure. Query lists featuring more than one matching entry based on normalized edit distance were then ranked by release date, and the track with the oldest release date was selected.

3.2 MetaData Variables

Each event in MIRAGE-MetaCorpus includes metadata for 100 variables obtained from the Radio Garden API (RG), the Natural Earth map data set (NE), the internet radio station stream encoder (SE), annotator review (AR), or using the online music libraries WikiData (WD), MusicBrainz (MB), Spotify (SP), Musixmatch (MX), YouTube (YT), Genius (GE), and AZlyrics (AZ). Shown in Table 1, these metadata reflect information about each event’s location, station, event, artist, and track. For

³ <https://open.spotify.com>; <https://www.wikidata.org>.

example, location metadata includes variables like the city, country, and geographic coordinates of the monitored event, as well as demographic data like the country’s population and GDP. Station metadata includes its name, form (a webcast stream, or a stream simulcast on the internet and terrestrial radio frequencies), formats (e.g., Top 40), and the station’s website url. Event metadata includes variables like the local time when the station was monitored and the event’s identifying metadata, such as the name of the artist and title of the recording. Finally, artist and track metadata include variables like the name and type of the artist, and if the artist is a group, a list of the group’s members and their demographic information (their listed genders, sexual orientations, and ethnicities), the group’s country of origin by birth and/or citizenship, the title and duration of the track, and its year of release.

3.3 MetaData Access & Export

Users may access the complete MIRAGE-Metacorp with the online dashboard.⁴ In addition, public-domain metadata from MIRAGE-MetaCorpus are available for download in an open-access repository on Zenodo [18], which includes both the complete data set and a subset of the data set for which the metadata obtained from the station’s stream encoder and the corresponding metadata provided by online music libraries was deemed a reliable match (i.e., where the normalized edit distance measure between the two metadata character strings was $\geq .90$ on a 0–1 scale).

4. MIRAGE ONLINE DASHBOARD

The MIRAGE online dashboard is an open-access web application that enables users to effortlessly navigate and engage with radio-station metadata and musicological features at various levels of detail. The dashboard’s layout consists of fully interactive panes displaying relevant information from MIRAGE-MetaCorpus. The dashboard is also compatible with multiple platforms and operating systems, so users may access and interact with the dashboard from any internet-connected device.

The complete technology stack of the dashboard includes Node.js for the server, MongoDB for the database, and React for the front end. This integration of technology guarantees a smooth user experience and effective data processing. Moreover, the dashboard may be tailored to accommodate individual users’ distinct requirements and inclinations, rendering it a versatile instrument for analyzing radio stations. What is more, incorporating these technologies enables instantaneous data updates and interactive functionalities, thereby boosting the overall user experience over subsequent versions of the dashboard. In addition, the MIRAGE dashboard offers sophisticated search and filtering tools and the ability to export metadata and visualizations in URL, CSV, PNG, and SVG formats, allowing users to study and share data easily.

⁴ The MIRAGE online dashboard is available at <https://pearl-laboratory.github.io/mirage-mc/>.

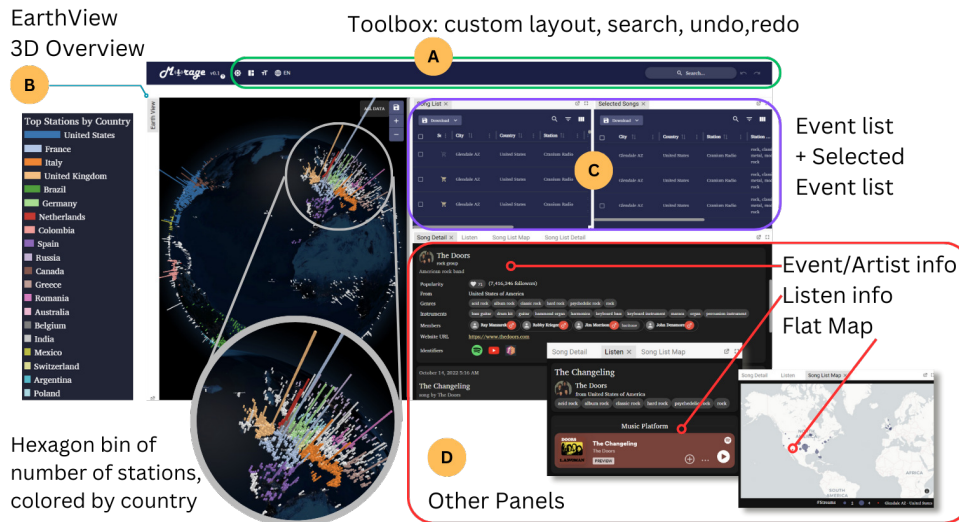


Figure 2. An overview of the MIRAGE online dashboard (v0.2).

4.1 Structure & Processing

Shown in Figure 2, the MIRAGE dashboard’s layout is divided into two groups: a toolbox on the top (A) and data-visualization panels below (B-D), making it easy for users to navigate and analyze information. The toolbox at the top includes options for language preference, panel-display customization, and searching. The data visualization panels show the data in various formats, such as charts, graphs, and tables, for straightforward interpretation and analysis. The panels can also dock to allow the user to create a customized layout, or open to another window (or undock) to permit a more detailed view suitable for multiple-screen presentations.

Shown in Figure 3, the database is partitioned into five tables: location, station, event, artist, and track. The data are structured in this manner to facilitate convenient retrieval and examination of each category while minimizing duplication. In this way, the database allows for easy filtering and sorting based on specific criteria, enhancing the overall efficiency of data analysis. Additionally, partitioning of data into separate tables helps to prevent errors and inconsistencies in data entry and manipulation.

4.2 Layout

4.2.1 Earth-View Panel

The 3D interactive Earth-view (or ‘globe’) panel visualizes the number of stations across the globe. Shown in Figure 2, each hexagonal-shaped vertical bar identifies the locations where radio stations reside. The height of each bar represents the number of stations at that geographic location, and the bars are also color-coded by country. The Earth-view panel is also linked to the event-list panel such that when a user selects a specific location on the Earth view, the event-list panel automatically filters (i.e., restricts) the station- and event-level metadata to the selected location.

In this way, users may compare the number of stations in various regions and discern any recurring patterns or trends.

4.2.2 Event-List Panels

Once users have selected a specific location on the interactive Earth-view panel or using the search function on the toolbox, they can retrieve metadata for the top 1,000 most recent entries in the event-list panel. Users can also select and add events to the selected event-list panel for further analysis and/or export, enabling users to revise their search parameters without losing selected metadata. The event-list panels also allow users to download the contents of either table in CSV format, or obtain a URL to share the

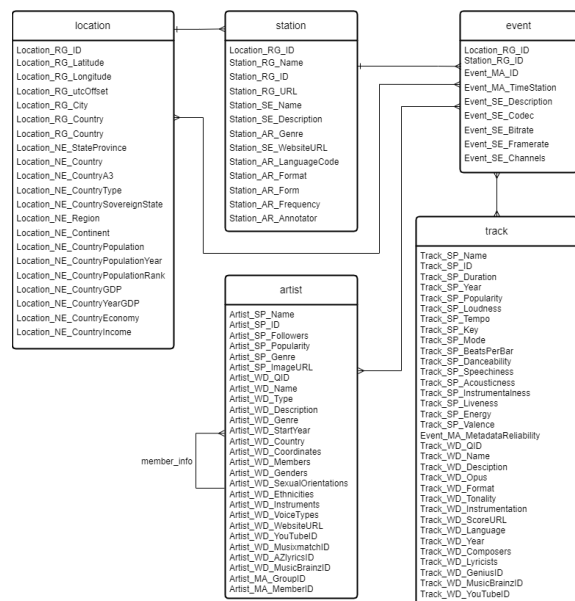


Figure 3. Database tables and connections for the MIRAGE online dashboard.

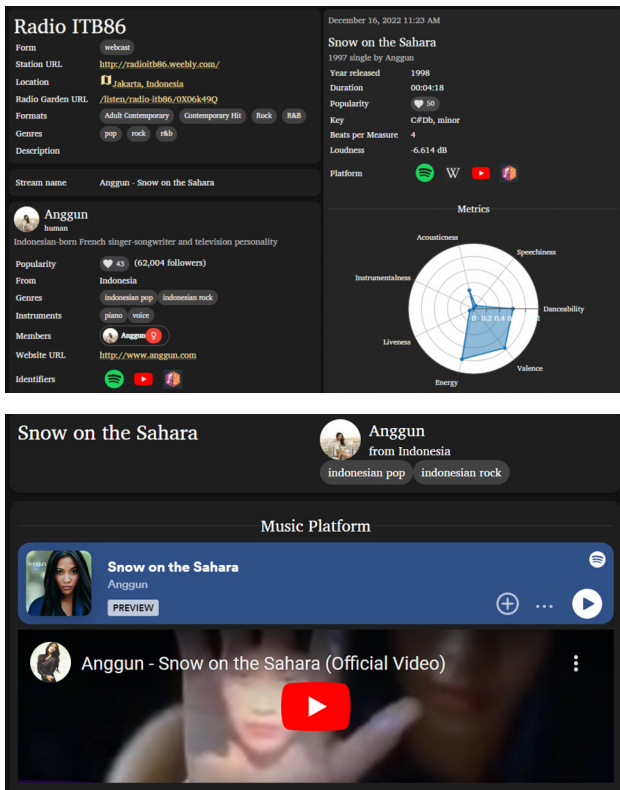


Figure 4. Top: Examples of the event-detail (top) and listen (bottom) panels in the MIRAGE online dashboard.

results of their most recent search with another user.

4.2.3 Map Panel

The map panel enables users to readily visualize the events’ geographic distribution in the event-list panel. Each dot on the map reflects the precise position of a particular event, and the size of the dot represents the number of events at that position. If the user selects an event from the event-list panel, that event will be represented by a red dot in the map panel. In this way, the map panel offers users a distinctive method for visualizing the geographical variety of the events in the event-list panel.

4.2.4 Event-Detail & Listen Panels

Shown in Figure 4, the event-detail panel displays the currently selected event from the event-list panel. The content is categorized into four sections: radio-station metadata (e.g., name, location, formats, url, etc.), event metadata (i.e., stream description), artist details (e.g., name(s), gender(s), group affiliations, instrument(s), etc.), and track metadata (e.g., track title, duration, key/mode, etc.). Figure 4, for example, presents all available metadata for Indonesian singer Anggun’s “Snow on the Sahara,” which streamed on Radio ITB86 in Jakarta, Indonesia on December 16, 2022. Note that users can obtain a list of demographic (e.g., gender, nationality, etc.) and musicological (e.g., list of instruments, vocal type, associated genres, etc.) information about Anggun, review additional metadata about the song itself (year released, language, the song’s lyricist(s), etc.), and finally navigate to other web-

sites and online music libraries using the provided hyperlinks.

Finally, the listen panel allows users to stream available recordings using embedded links to the integrated Spotify and YouTube platforms. Although not all events are available on both platforms, the dashboard is regularly updated to ensure that the provided information is current.

4.2.5 Event-List Visualization Panel

Shown in Figure 5, the event-list visualization panel allows users to explore the searched or selected event list using interactive bar, scatter, and histogram plots. For each plot, users may select the appropriate metadata variable(s) from a dropdown list, edit the plot using Plotly Chart Studio,⁵ and finally export the plot in SVG format.

5. EXAMPLE USE CASE

The metadata and visualizations produced by the MIRAGE online dashboard have numerous applications for users. Figure 5, for example, examines Anggun’s “Snow on the Sahara” within the context of events produced by Indonesian artists across the globe (left), or streaming on Indonesian radio stations (right). The MIRAGE-MetaCorpus features 37 events (and 12 tracks) produced by Anggun, of which 21 were “Snow on the Sahara” (or its French language version, “La neige au Sahara”).

Among Indonesian artists, music genres familiar to western listeners like pop, pop-rock, and alternative rock rank in the top 10, along with characteristic southeast Asian genres like dangdut and koplo. Among events streaming on Indonesian stations, music by Indonesian artists also ranks first, though several Anglophone countries also rank in the top ten (USA, UK, etc.). Scatter plots of a two-dimensional arousal-valence emotion space and a two-component solution from a principal components analysis of the track’s danceability, speechiness, acousticness, liveness, and instrumentality further reveal the track’s unconventional expressive and musical characteristics relative to the other tracks produced by Indonesian artists or streaming on Indonesian stations. Finally, histograms of the track’s popularity and year of release reflect the song’s enduring popularity more than two decades after its initial release.

6. CONCLUSION & ETHICAL CONSIDERATIONS

This development release (v0.2) of the MIRAGE online dashboard provides a snapshot of the contemporary global listening landscape for scholars across the (digital) humanities. Our purpose in doing so is to facilitate cross-cultural, comparative research, which has become a pressing concern in several music disciplines [1, 19–21]. To that end, the MIRAGE-MetaCorpus features metadata for 1 million events that streamed on 10,000 radio stations across the globe, and the dashboard is interoperable with several platforms and operating systems [22, 23].

⁵ chart-studio.plotly.com.

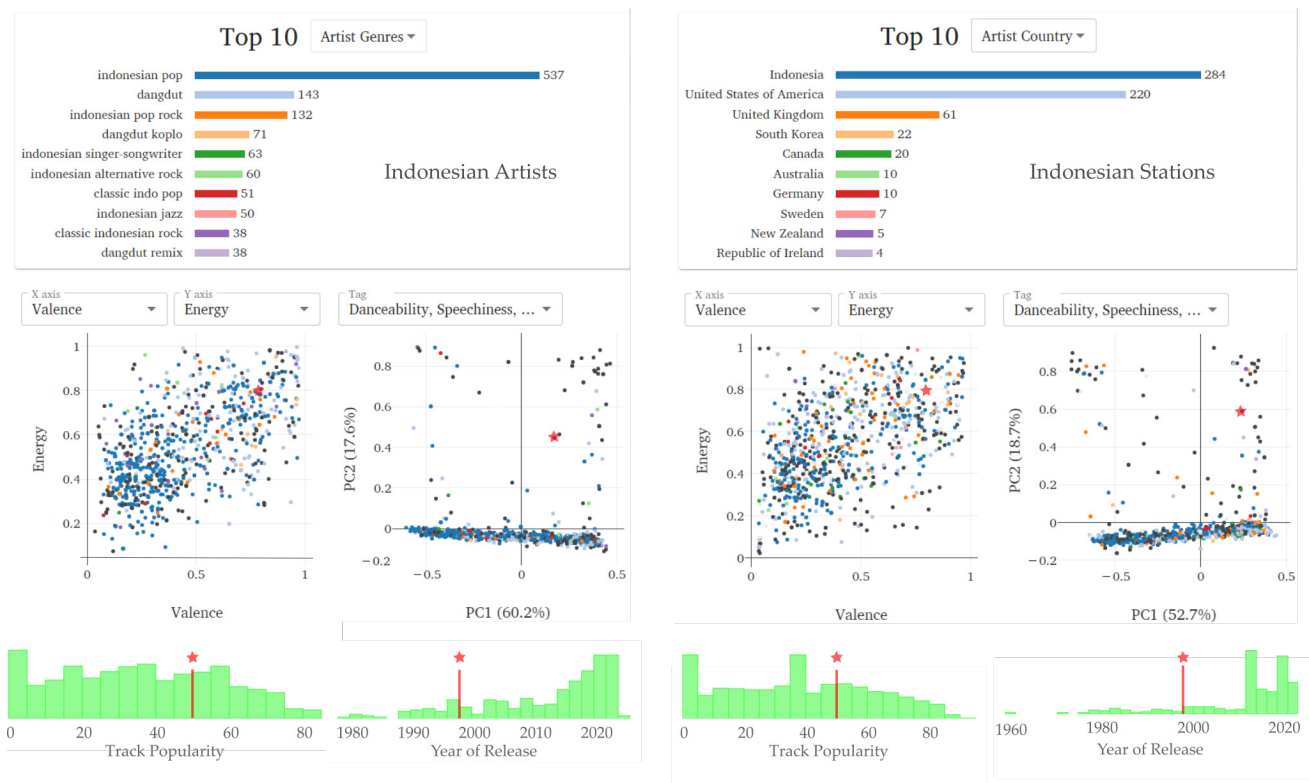


Figure 5. Event-List Visualization Pane for events by Indonesian artists (left) or Indonesian stations (right) in the MIRAGE online dashboard. The red star indicates the position of Anggun’s “Snow on the Sahara.”

As a metadata repository, the MIRAGE-MetaCorpus contains links to online resources that we do not control. To mitigate the potential for dataset degradation over time, the research team plans to update (and collect additional) metadata annually. Nevertheless, the attribution metadata provided by the radio station’s stream encoder does not always reliably match metadata provided by online music libraries. In our view, nonmatching (or ‘unreliable’) metadata allow the research community to evaluate the coverage (i.e., bias) of online music libraries for the music found on internet radio. Nevertheless, MIRAGE users should be aware of the potential for matching errors. For tasks where higher match quality is important, users may search for reliable metadata in the online dashboard, or export reliable subsets of the MIRAGE-MetaCorpus.

Similarly, this project provides access to metadata and musicological features produced by proprietary (or otherwise undisclosed) algorithms, often trained on western musical traditions and their associated organizational principles. As a result, we encourage the research community to treat the attribution metadata in MIRAGE as a starting point for developing corpora and methodologies involving other musical traditions [24, 25].

In developing the MIRAGE online dashboard, the research team has attempted to protect the interests of copyright holders by only including publicly available metadata protected under fair use while enabling users to stream recordings using embedded links to commercial services like Spotify or YouTube. The dashboard also adheres to the user agreements from the libraries and streaming ser-

vices mentioned above (e.g., Radio Garden, Spotify, WikiData), according to which users may access and interact with all data on the online dashboard, but they may only export public-domain data for further analysis and study (i.e., from the Radio Garden API, the Natural Earth data set, station stream encoder, and WikiData). Perhaps most importantly, this project did not directly record/store audio from station streams at any point in the data-collection pipeline.

Nevertheless, we acknowledge the concerns of copyright holders (artists, radio stations, online music libraries, and streaming services) who do not wish to share attribution metadata about their work (e.g., artist demographics, track details, etc.). We only provide links to publicly available sources and do not own the copyright for any music referenced in the MIRAGE-MetaCorpus. For that reason, copyright holders may request the removal of metadata from the MIRAGE project.⁶

In addition to completing station-list review for the remaining stations in MIRAGE-MetaCorpus, future versions of the dashboard will transition from React+Nodejs to Remix in order to enhance the speed of queries and allow users to access and review more than 1,000 events simultaneously in the event-list panel. The team also plans to conduct a usability study to examine the dashboard’s practical utility, as well as incorporate additional customizable sampling and visualization tools like statistical surface maps to enhance the user’s exploration of metadata variables in MIRAGE [26]. In doing so, we hope future versions of

⁶ Please contact miragedashboard@gmail.com.

this dashboard will facilitate cross-cultural, comparative research for a medium that places diversity center stage.

7. REFERENCES

- [1] K. Lacey, "Up in the air? the matter of radio studies," *Radio Journal: International Studies in Broadcast Audio Media*, vol. 16, no. 2, pp. 109–126, 2018.
- [2] A. J. Bottomley, *Sound streams: A cultural history of radio-internet convergence*. Ann Arbor, MI: University of Michigan Press, 2020.
- [3] M. Glantz, "Internet radio adopts a human touch: A study of 12 streaming music services," *Journal of Radio Audio Media*, vol. 23, no. 1, p. 36–49, 2016.
- [4] T. Wall, "The political economy of internet music radio," *The Radio Journal*, vol. 2, no. 1, p. 27–44, 2004.
- [5] T. Chambers, "Radio programming diversity in the era of consolidation," *Journal of Radio Studies*, vol. 10, no. 1, p. 33–45, 2003.
- [6] D. Hendy, *Radio in the global age*. Cambridge, UK: Polity Press, 2000.
- [7] H. Uimonen, "Beyond the playlist: Commercial radio as music culture," *Popular Music*, vol. 36, no. 2, pp. 178–195, 2017.
- [8] J. A. Burgoyne, J. Wild, and I. Fujinaga, "An expert ground-truth set for audio chord recognition and music analysis," in *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR)*, A. Klapuri and C. Leider, Eds., Miami, FL, 2011, pp. 423–428.
- [9] T. de Clercq and D. Temperley, "A corpus analysis of rock harmony," *Popular Music*, vol. 30, pp. 47–70, 2011.
- [10] D. R. W. Sears and D. Forrest, "Triadic patterns across classical and popular music corpora: Stylistic conventions, or characteristic idioms?" *Journal of Mathematics and Music*, vol. 15, no. 2, p. 140–153, 2021.
- [11] T. Bertin-Mahieux, D. P. W. Ellis, B. Whitman, and P. Lamere, *The million song dataset*, Miami, FL, 2011.
- [12] N. Aizenberg, Y. Koren, and O. Somekh, *Build your own music recommender by modeling internet radio streams*, Lyon, France, 2012.
- [13] T. Dang, A. Anand, and L. Wilkinson, *FmFinder: Search and filter your favorite songs*. Heidelberg, Germany: Springer-Verlag, 2012, vol. 7431.
- [14] G. R. L. Silva, L. M. de Oliveira, R. R. de Medeiros, O. Goussevskaia, and F. Benevenuto, "Characterizing internet radio stations at scale," in *Proceedings of the International Conference on Web Intelligence (WI 2017)*. Association for Computing Machinery, 2017, Conference Proceedings, pp. 670–677.
- [15] M. Ala-Fossi, S. Lax, B. O'Neill, and H. Shaw, "The future of radio is still digital—but which one? expert perspectives and future scenarios for radio media in 2015," *Journal of Radio Audio Media*, vol. 15, no. 1, p. 4–25, 2008.
- [16] R. A. Lind and N. J. Medoff, "Radio stations and the world wide web," *Journal of Radio Audio Media*, vol. 6, no. 2, pp. 203–221, 1999.
- [17] F. Kuhn, "Internet radio flows: Between the local and the global," *Radio Journal: International Studies in Broadcast Audio Media*, vol. 9, no. 1, pp. 35–49, 2011.
- [18] D. R. W. Sears, "Music Informatics for Radio Across the GlobE (MIRAGE) MetaCorpus (v0.2)," Jul. 2024. [Online]. Available: <https://doi.org/10.5281/zenodo.12786202>
- [19] P. A. Ewell, "Music theory and the white racial frame," *Music Theory Online*, vol. 26, no. 2, 2020. [Online]. Available: <https://www.mtosmt.org/issues/mto.20.26.2.ewell.php>
- [20] N. Jacoby, E. H. Margulis, M. Clayton, E. Hannon, H. Honing, J. Iversen, T. R. Klein, S. A. Mehr, L. Pearson, I. Peretz, M. Perlman, R. Polak, A. Ravnigani, P. E. Savage, G. Steingo, C. J. Stevens, L. J. Trainor, S. E. Trehub, and M. Veal, "Cross-cultural work in music cognition: Challenges, insights, and recommendations," *Music Perception*, vol. 37, no. 3, pp. 185–195, 2020.
- [21] P. E. Savage and S. Brown, "Toward a new comparative musicology," *Analytical Approaches to World Music*, vol. 2, no. 2, pp. 149 – 197, 2013.
- [22] F. C. Moss and M. Neuwirth, "Fair, open, linked: Introducing the special issue on open science in musicology," *Empirical Musicology Review*, vol. 16, no. 1, pp. 1–4, 2021.
- [23] M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, ..., and B. Mons, "The fair guiding principles for scientific data management and stewardship," *Scientific Data*, vol. 3, no. 1, p. 160018, 2016.
- [24] G. Born, "Diversifying MIR: Knowledge and real-world challenges, and new interdisciplinary futures," *Transactions of the International Society for Music Information Retrieval*, vol. 3, no. 1, pp. 193–204, 2020.
- [25] R. S. Huang, A. Holzapfel, B. L. Sturm, and A. K. Kaila, "Beyond diverse datasets: Responsible MIR, interdisciplinarity, and the fractured worlds of music," *Transactions of the International Society for Music Information Retrieval*, vol. 6, no. 1, pp. 43–59, 2023.
- [26] D. O'Sullivan and D. J. Unwin, *Geographic Information Analysis*, 2nd ed. Hoboken, NJ: Wiley, 2010.

MIDI-TO-TAB: GUITAR TABLATURE INFERENCE VIA MASKED LANGUAGE MODELING

Drew Edwards Xavier Riley Pedro Sarmento Simon Dixon

Centre for Digital Music, Queen Mary University of London, UK

{a.c.edwards, j.x.riley, p.p.sarmento, s.e.dixon}@qmul.ac.uk

ABSTRACT

Guitar tablatures enrich the structure of traditional music notation by assigning each note to a string and fret of a guitar in a particular tuning, indicating precisely where to play the note on the instrument. The problem of generating tablature from a symbolic music representation involves inferring this string and fret assignment per note across an entire composition or performance. On the guitar, multiple string-fret assignments are possible for most pitches, which leads to a large combinatorial space that prevents exhaustive search approaches. Most modern methods use constraint-based dynamic programming to minimize some cost function (e.g. hand position movement). In this work, we introduce a novel deep learning solution to symbolic guitar tablature estimation. We train an encoder-decoder Transformer model in a masked language modeling paradigm to assign notes to strings. The model is first pre-trained on DadaGP, a dataset of over 25K tablatures, and then fine-tuned on a curated set of professionally transcribed guitar performances. Given the subjective nature of assessing tablature quality, we conduct a user study amongst guitarists, wherein we ask participants to rate the playability of multiple versions of tablature for the same four-bar excerpt. The results indicate our system significantly outperforms competing algorithms.

1. INTRODUCTION

Tablatures (tabs) are a type of music notation where each played note is indicated by its physical position on the instrument, as opposed to merely its pitch. Whereas on (e.g.) the piano, each pitch can be played in exactly one location on the instrument, most playable pitches on stringed instruments like the guitar or violin can be played in multiple positions [1]. This redundancy introduces an additional layer of analysis to derive mechanics of a performance from raw pitches. In traditional music scores, e.g. for classical guitar, it is the burden of the performer to select appropriate fingerings and positions for the notes in the sheet music. Similarly, a MIDI transcription of a guitar recording lacks

this crucial information for a guitarist to replicate the performance.

In this research we examine the problem of mapping a symbolic representation of a musical performance to guitar tablature. There are few recent publications on this topic (see Section 2), although there are commercial solutions available. Most existing methods propose a manually defined objective function, often related to the difficulty of hand stretches to play chords and distances between hand positions, and seek a solution that minimizes the cost. We take a different approach and provide a modern machine learning treatment of the problem.

We cover the following aspects of our research in this paper: first, we provide a background on the research related to guitar transcription and tablature estimation. Then we formally define the problem of tablature inference from symbolic music notation. Next, we describe the methods of our research, which include: a simple tokenization, a masked language model learning task, a Transformer model solution, pre-training and fine-tuning phases, and a custom beam search inference. Finally, we characterize the performance of our system with quantitative and qualitative metrics, including a detailed user study with 15 guitarists rating various tablatures for short solo guitar excerpts. Our results indicate that guitarists significantly prefer our automatic tablatures versus the commercial alternatives we benchmark.

2. RELATED WORK

The earliest algorithmic approaches to automatic guitar tablature systems date back to Sayegh [2]. His approaches include an expert system approach assigning rules of permissible transitions between hand positions. These are encoded and enforced via Prolog and its native constraint solver. The second approach described assigns costs to transitions between fingerings and uses dynamic programming (Viterbi [3]) to find an optimal path through the constructed weighted graph. This latter approach represents the standard classical benchmark for tablature inference. Alternate approaches include genetic algorithms [4], hybrid expert systems [5], and hidden Markov models [6]. Hori and Sagayama [7] extend the dynamic programming approach by finding a path that minimizes the maximum cost of a local transition across a phrase, as opposed to minimizing the global cost as in Sayegh. Radicioni [8] estimates the fingers employed as well as the fret-string



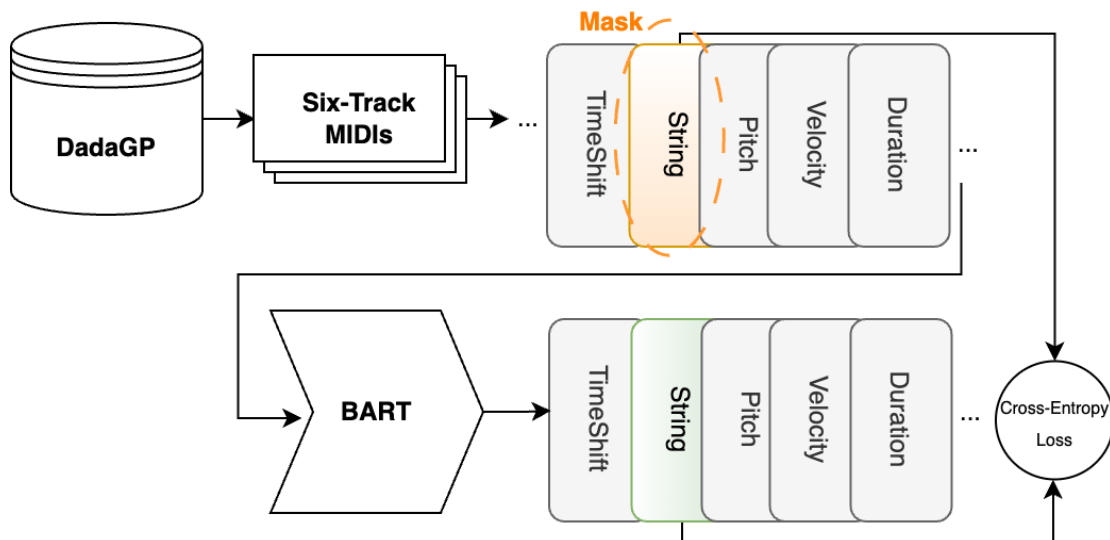


Figure 1. Overview of the training procedure. Guitar Pro files from DadaGP are converted to six-track MIDI files, one file per distinct guitar part and one track per string. These are tokenized into the Structured tokenization of MidiTok. We train a BART model in a simple masked language modeling task where the string tokens are masked out. Only the predictions for the string tokens are used for loss signal propagation.

combinations, using a graph search paradigm to optimize the bio-mechanical comfort of rendering a piece.

In addition to purely symbolic approaches, there is considerable research on the topic of automatic guitar transcription from audio input. Yazawa et al. [9] follow a two-stage approach which uses latent harmonic allocation for multi-pitch estimation (MPE) and then removes unplayable pitches as determined by a fingering cost algorithm similar to Sayegh. Wiggins and Kim [10] apply a convolutional neural network to jointly perform MPE and tablature fingering. The strongest performing MPE methods for guitar [11, 12] leverage the vast amount of transcription material from other instruments (particularly piano) to enlarge the training dataset, but they fall short in offering no tablature estimations.

The most similar approach to our own is described in the master’s thesis of Mistler [13], where recurrent neural networks are trained to predict guitar tablatures. However, the training dataset used only contained 74 songs and uses hand-crafted features extracted from the input MusicXML. In contrast, we train on tens of thousands of tabs and process a raw, MIDI-derived tokenization of the input score. The data used for training our network comes from the DadaGP dataset [14], comprising 26,181 song scores in the Guitar Pro format.

3. PROBLEM FORMULATION

We simplify the task of guitar tablature estimation to the task of assigning notes to strings. For a specific guitar tuning, the combination of pitch, string, and fret has only two degrees of freedom. Thus, since the pitch is known a priori, we may predict the string and compute the resulting fret for the assignment. This essentially reduces the problem to *se-*

quence labeling. In order to increase the flexibility of our system to process a variety of data sources, we begin with MIDI data. Any digital score can be converted to MIDI, and most automatic transcription systems produce MIDI data as well, enabling our tablature system to be composed with any MPE algorithm.

The problem is formally structured as follows: given a one-track MIDI file M , the system produces a six-track MIDI file M_S , where each track contains the notes assigned to a particular string. Let

$$\mathcal{O} = \{64, 59, 55, 50, 45, 40\}$$

denote the list of MIDI note numbers for the open strings of the guitar in standard tuning, corresponding to $E_4, B_3, G_3, D_3, A_2, E_2$, respectively. Thus, to derive the fret of a note with MIDI note number n assigned to a string s , where $s \in \{1, 2, 3, 4, 5, 6\}$, the fret f is calculated by:

$$f = n - \mathcal{O}[s]$$

Although our derivation of the fret value assumes a standard tuning, this approach could be easily modified to permit alternate tunings by changing the values of \mathcal{O} .

4. METHODS

4.1 Architecture

Our solution to the problem uses a Transformer with a bi-directional auto-encoder and a left-to-right decoder, based on the BART model architecture [15]. Using such an approach requires a tokenization of the input data. For this, we use the *Structured* tokenization scheme of Huang and Yang [16]. For each MIDI note, we produce five tokens: time shift, string (i.e. track), pitch, velocity, and duration.

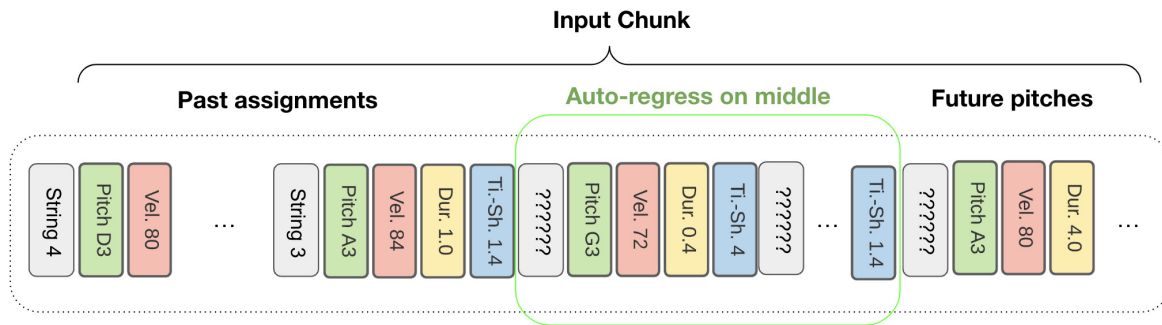


Figure 2. A diagram of our quintile inference algorithm. The middle fifth of the attention window is predicted in an auto-regressive fashion. String assignments from earlier quintiles are fixed. Future notes are available in the context window but will not be assigned until the processing window places them in the center. The beam search is not depicted.

During training, we use a simple masked language modeling supervision scheme, masking the string tokens. This permits both past and future note values to be available for the network during inference, but only past string assignments can be seen. During training, we only compute the loss for string token predictions. We use the Hugging Face Transformers package to define the network, using hyperparameter settings that simply halve the `bert-base` configuration: 384-dimensional hidden size, 6 hidden layers, 6 attention heads, 1536-dimensional intermediate size, dropout probability of 10%. These hyperparameter values were not finetuned.

4.2 Training

Training takes a two-phase approach (see Figure 1): first we train from scratch on 27,619 guitar tablatures derived from DadaGP¹. We use the pre-processing code from the SynthTab project [17] to produce a six-track MIDI file for each guitar part in the Guitar Pro files. Training is performed with the AdamW optimizer of PyTorch, employing a linear decay schedule with warm-up and an initial learning rate of 1×10^{-4} and runs for 100 epochs. The second phase of training is a finetuning on precisely annotated guitar performances from the training splits of Riley et al. [12] and GuitarSet [18]. The fine-tuning stage is motivated by the concern of data quality in the DadaGP annotations, which were scraped from the online, crowd-sourced tab library Ultimate Guitar². Here we fine-tune with a learning rate of 1×10^{-5} , again for 100 epochs, on the much smaller data of 281 tabs. Examples are fed into the network in note sequences of length 50, corresponding to 250 tokens per example.

4.3 Inference

A common problem when training Transformer models for auto-regressive tasks is an asymmetry between training and inference regarding previously predicted sequence values. To leverage the parallelism of the Transformer architecture, ground truth labels must be used as decoder input for

masked preceding values. However, during inference on unseen data, these labels are unavailable.

We implement a novel inference mechanism for our algorithm. We break up the input segment into quintiles (10 notes or 50 tokens per quintile). Excluding boundary cases, we only make predictions for the center quintile (see Figure 2). This allows our network to have the ability to see the 20 previous note-string assignments and the next 20 future note values. The attention window is advanced by 10 notes per inference step. Additionally, we implement a custom beam search inference. For each string prediction in a quintile, we retain the top two string values for the note. We limit the number of potential paths to 32. The paths are batched to keep inference times nearly equivalent to naive autoregression. Paths are pruned by taking the maximum probability computed by summing the logits of the string predictions. While this does not fully resolve the asymmetry between inference and training, the beam search and additional context provide more probable decoder input values than naive autoregression.

4.4 Post-processing

Thus far we have not imposed any constraints on the output of the network. Ideally, we would take the string predictions and directly augment the score information with the resulting tablature. However, in our qualitative assessment of the system, there are occasions where a string-fret prediction can lead to invalid or unplayable notes. To address these outliers, we attempt to relocate the note to a more suitable string.

The heuristic algorithm is as follows:

1. Merge and sort notes from all strings by start time.
2. Set maximum allowable deviation from the average fret position ($MAX_DEVIATION = 5$) and the highest playable fret ($MAX_FRET = 21$).
3. For each run of 11 notes (5 past, 1 middle, 5 future):
 - (a) Find the average fret value of the run, excluding open strings from the computation.
 - (b) If the middle note has a fret value exceeding MAX_FRET or $MAX_DEVIATION$:

¹ Some pieces in DadaGP have multiple guitar parts, and some tracks are filtered out in the conversion process.

² <https://www.ultimate-guitar.com/>

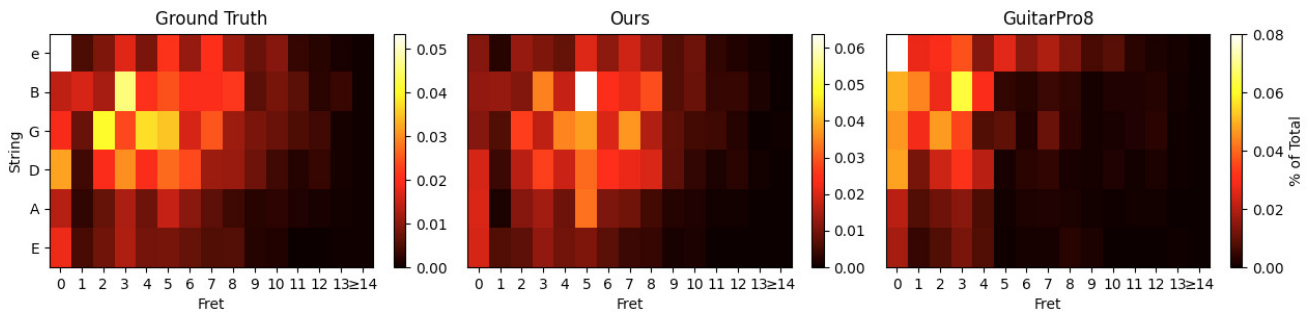


Figure 3. Heatmaps of the fret-string distributions for three of the five tablature systems (ground truth, ours, and Guitar Pro 8). Overall, our system has a similar distribution to the ground truth, but the output appears to be biased away from open strings. Guitar Pro 8 shows a heavy skew to low frets, which perhaps suggest a bias towards playing in “first position” (playing primarily on frets 1 to 4).

- i. Define the available strings to be those with no notes intersecting with the note under consideration.
- ii. If an open string is available, select it.
- iii. Else, select the string yielding a fret value closest to the neighborhood mean.

Ultimately, in our test set from Riley et al. [12], our post-processing algorithm only modifies 0.53% of all notes. However rare, addressing these failures is important to ensure the resulting tab is playable.

5. RESULTS

5.1 Quantitative Results

We use the training splits from Riley et al. [12], with 61 pieces in the training set, 8 in the validation set, and 9 in the test set, corresponding to 58,080, 7,031, and 8,451 notes respectively. At the end of finetuning, our next note accuracy on the validation set is 94.35%. This is the probability of correctly inferring the next note-string assignment given ground truth labels up to the point of prediction. When evaluating autoregressively on the held-out test set across 50-note³ examples, our network agrees with ground truth on 82.52% of predictions. This discrepancy highlights the difference between teacher-forcing and errors accumulated in auto-regressive inference. We measure the impact of the finetuning step by evaluating the pre-trained model without finetuning, which gives 78.48% agreement, corresponding to a 4.04 percentage point difference due to finetuning.

To compare our algorithm to existing technologies, we use one commercially available and two open-source implementations of automatic tablature systems. Guitar Pro 8⁴ is a music software program designed for editing, visualizing, and sharing guitar, bass, and other stringed instruments’ tablatures, and includes an algorithm to automatically produce tablature from score or MIDI. MuseScore⁵ is an open-source score editor with a similar functionality for generating tablature. TuxGuitar⁶ is free, open-source

software for creating and playing guitar tablature and standard musical notation. We use each of these systems to generate MusicXML files with tablature for our 9 held-out test scores from our finetuning dataset, which are then used for evaluation.

Objective evaluation of guitar tablature is difficult, as we will discuss further in Section 5.2. We provide three metrics that illustrate the strength of our system. The first metric is a measure of agreement between the ground truth note-string assignment and each algorithm’s assignment for the corresponding note. The metric is computed by matching⁷ notes from each measure of the ground truth with the notes from the inferred tablature’s corresponding measure. An agreement occurs when the ground truth and the inferred tab assign the same string. The total number of agreements is counted across all examples and then divided by the total number of notes compared. Our system shows the highest agreement of 73.18% (see Table 1). This falls below our 84.42% agreement from the 50-note examples, because early disagreements on fretboard location for a group of notes will likely cause subsequent note assignments to continue to disagree.

The other two metrics relate to the “stretch” values across chords in the MusicXML. Chords are extracted as note onsets occurring at the exact same time. For all such groups of notes, we define the stretch as the maximum fret-wise distance between any two notes in the chord. For example, a chord with notes F3, C4, E4, and A4 played on the string-fret locations⁸ (4, 3), (3, 5), (2, 5), (1, 5) will have a stretch value of 2. Open strings do not restrict hand positions so they do not contribute to the stretch. We report the maximum and average stretch across the chords in the test set. The ground truth has the lowest maximum stretch of 6. Our system demonstrates occasional erratic behavior of assigning high notes to lower strings, resulting in a shifted mean and larger maximum stretch value of 12. In Figure 4, we compare frequencies of maximum fret distances between our system and the ground truth. An example failure is shown in Figure 5. The mean and median values indicate that, in general, all algorithms attempt to place chords

³ Recall the model has a context window of 50 notes.

⁴ <https://www.guitar-pro.com/>

⁵ <https://musescore.com/>

⁶ <https://www.tuxguitar.app/>

⁷ Matching is required due to reordering of simultaneous notes.

⁸ String 1 is the High E string, and fret values start at 0 for open strings.

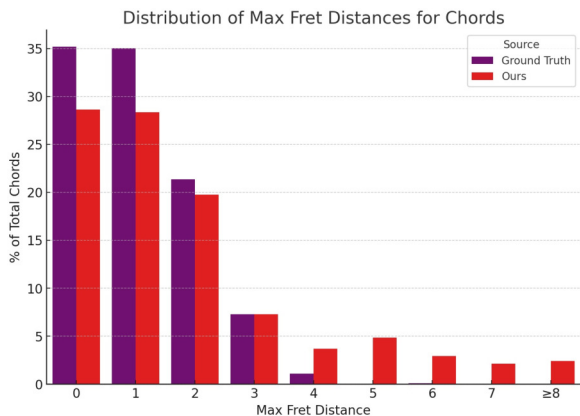


Figure 4. Comparison of the distributions of stretch distances between chords in the test set.

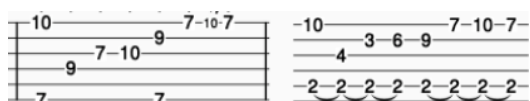


Figure 5. An example failure of our system. Ground truth is left, ours is right. The assignment of B2 to the fifth string creates an 8-fret stretch, which is essentially unplayable.

within a narrow band of frets. The presence of these large stretches motivates future work to better inform our algorithm about the importance of physical playability.

As a final quantitative comparison, we compute fret-string distributions for all five sets of tablatures. For each distribution, we compute the Kullback–Leibler divergence against the ground truth distribution. Our system has the lowest value of 0.099; the other values are: 0.462 for Guitar Pro 8, 0.635 for MuseScore, and 1.286 for TuxGuitar. Three of these distributions are shown in Figure 3.

5.2 User Study

A purely quantitative evaluation of automatic guitar tablature systems is problematic because there may be multiple ways to play the same phrase or excerpt of music. For example, in Figure 6, we show two distinct tablatures for the same one-bar phrase. The top is the ground truth transcription

Table 1. Summary of quantitative analysis, showing maximum, mean and median “stretch” of chords in the tablature, defined as the maximum fret distance between any two notes in the chord. We also report the percent agreement with the ground truth note-string assignment. All metrics are averaged over the entire withheld test set.

Source	Max Stretch	Mean Stretch	Median Stretch	% Agree
Ground Truth	6	1.04	0	–
Ours	12	1.84	1	73.58
MuseScore	10	1.19	1	62.51
Guitar Pro 8	12	0.78	0	62.27
TuxGuitar	18	2.03	1	55.42

tion and the bottom is the layout from our system. At a glance, both provide reasonable fretboard fingerings. The ground truth shows a preference for open strings, but our system better minimizes the maximum span between successive notes (2 frets versus 4 frets). However, in this example, our system only agrees with the ground truth on two notes, which corresponds to an accuracy of 16.67%. On the other hand, hand-crafted metrics (such as maximum or average span between notes) fail to capture complex preferences of guitar tablature – otherwise existing systems that minimize these values as cost functions would suffice.

To complement the quantitative analysis and circumvent some of the potential limitations of the approach, we conducted a study to assess guitarists’ opinions on the playability and overall preferences for tablatures. Participants were exposed to 30 audio excerpts consisting of 4-bars of solo jazz guitar audio, and for each were shown 5 distinct tablature transcriptions from the following groups: TuxGuitar (*TG*), MuseScore (*MS*), Guitar Pro 8 (*GP*), our system (*Ours*) and ground truth (*GT*), which was created by a professional transcriber. The stimuli were selected by randomly sampling the test split of Riley et al. [12]. Via an online listening study, we probed how guitar players deem the tablatures generated by our system, and how they rank them against the ground truth and the outputs from other tablature generation software (i.e. *TG*, *MS* and *GP*).

The online listening study took approximately 1.5 hours to complete and both the order of audio excerpts and the order of tablature transcriptions were randomized. As conditions to take part in the study we proposed that participants should be guitar players and have a familiarity with reading tablatures, access to headphones or speakers and normal hearing. Subjects were instructed to ignore the difficulty of the music excerpts as they rated the playability of the tablatures. Overall, we recruited 15 guitarists and invited them to attempt to play each of the tablature examples on the guitar during the study, while rating each of the tablature transcription groups on a scale from 1 to 10. Participants, with an age distribution of 39 ± 14 years, reported a median value of 10 years of daily regular engagement with practice of the guitar, and a median value of 3 hours of guitar practice per day at the peak of their interest. The study received ethical approval from the Queen Mary University of London Ethics of Research Committee (QMERC20.565.DSEEC24.012), and participants were compensated with an Amazon gift voucher. Results for the listening test can be observed in Figure 7.

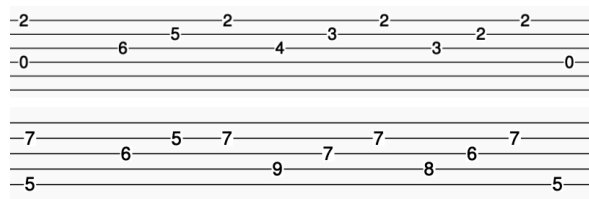


Figure 6. Two tablatures for the same musical excerpt. The top is ground truth, the bottom is from our system.

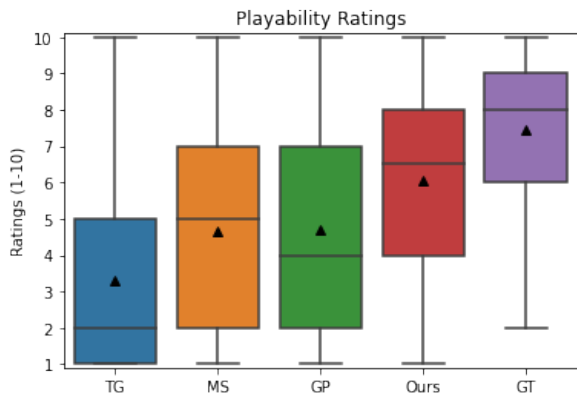


Figure 7. Box plots of the results of the listening study on the playability of tablatures. Bars indicate median values and triangles indicate mean values, for each group.

As expected, the ground truth group ranks highest (7.45 ± 2.62). We hypothesise that the reason why the ground truth falls well short of a “perfect” score is linked to the subjective preferences of participants in terms of fingerings and note position choices, which are inherently linked to their guitar playing techniques and overall style. Furthermore, the ground truth represents professional transcriptions of jazz recordings, where perfect information of the original tablature is not available. As discussed above and illustrated in Figure 6, there can be multiple reasonable ways to play a phrase, and we observed that the ratings for our system were higher than those of the ground truth group for 101 ratings out of 450 in total. The results show that participants tend to rate the playability and overall preference of the tablatures from our system (6.04 ± 2.67) higher than the ones from the competing software (*TG*: 3.32 ± 2.87 ; *MS*: 4.67 ± 2.88 ; *GP*: 4.69 ± 2.64).

Of the 2,250 data points collected (30 excerpts \times 5 tabs \times 15 participants), a Shapiro-Wilk test showed that data was not normally distributed within groups. Due to the repeated measurements characteristic of the test (every participant rates all the stimuli), we use a Friedman test to investigate the effects of the type of tablature transcription system on the perceived playability of tablatures, with a Type I error α of 0.05. The statistical results showed a highly significant effect of the tablature transcription system in the participants’ responses amongst groups ($\chi^2(4) = 532.09$, $p < .001$). Finally, in order to determine if there were statistically significant differences between groups, we conducted a post-hoc pairwise Wilcoxon test, Bonferroni-adjusted α level of 0.005 ($.05/10$). This yielded highly significant differences in ratings between groups, except for (*MS*, *GP*).

6. DISCUSSION

Our results suggest a data-driven approach to guitar tablature inference can yield predictions that are significantly more aligned with guitarists’ preferences than existing methods. These results are very encouraging given the simplicity of our approach. Our system imposes no con-

straints on the predicted tablatures until the final post-processing, during which less than 1% of note-string assignments are modified. Future research in this direction may benefit from more directly encoding positional fretboard locations and physical limitations as input to the network.

Despite the strong results, our system has several limitations to be addressed. Guitar tuning is never explicitly encoded as input to the model. Since we only predict string values, our fret predictions are always derived from the note-string assignment and an assumption of standard tuning. Similarly, our system does not handle the use of capos. The system is unaware of many guitar specific articulations, such as harmonics, hammer-ons, pull-offs, and pitch bends. Finally, we make no attempt to assign individual notes to the fingers of a guitarist, which is occasionally done in professional scores or transcriptions, and would be a necessary step in order to estimate playability explicitly.

Another criticism of our approach is that it does not use visual and audio cues for fretboard prediction. As shown by Bastas et al. [19], inharmonicity analysis of a particular instrument can improve string predictions. Likewise, Duke and Salgian [20] demonstrate how computer vision models can be used for accurate and real-time tablature transcription. Both of these directions of research offer a more faithful reproduction of a particular performance, since a symbolic approach simply has no access to disambiguating signals such as hand position or string inharmonicity. However, this shortcoming can also be viewed as a strength: our system does not need access to video nor audio. From this perspective, our approach can be viewed as an automatic arranging system for guitar tablature performance.

The main failure mode of our system is the assignment of unplayable chords at a small but significant frequency (2.4% of chords have a maximum fret distance exceeding 7). Future research may explore different tokenization schemes: encoding fret values as input, physically inspired loss functions, or more carefully designed post-processing to handle these cases. However, the vast majority of the mass of the distribution of chord stretch distances falls within playable limits, which indicates that the algorithm is implicitly modeling some of the physical constraints that classical systems use to derive tablatures.

7. CONCLUSION

We present a deep learning algorithm to predict guitar tablature from symbolic music notation. Our methodology trains an encoder-decoder Transformer to learn tablature assignment from raw note events. Drawing inspiration from natural language processing, we begin by pre-training on a dataset of tens of thousands of tablatures and then fine-tune on a curated dataset of professional guitar scores. We evaluate our system against commercially available software and demonstrate a significant preference for our system through a user study among guitarists. Our MIDI-to-Tab system represents a first step towards achieving human-level tablature inference via machine learning.

8. ACKNOWLEDGMENTS

Authors DE, XR, and PS are research students at the UKRI Centre for Doctoral Training in Artificial Intelligence and Music, supported by UK Research and Innovation [grant number EP/S022694/1] and Yamaha Corporation (DE).

9. REFERENCES

- [1] P. Sarmiento, A. Kumar, D. Xie, C. Carr, Z. Zukowski, and M. Barthelet, "ShredGP: Guitarist style-conditioned tablature generation," in *The 16th International Symposium on Computer Music Multidisciplinary Research*, Tokyo, Japan, 2023.
- [2] S. I. Sayegh, "Fingering for string instruments with the optimum path paradigm," *Computer Music Journal*, vol. 13, no. 3, pp. 76–84, 1989. [Online]. Available: <http://www.jstor.org/stable/3680014>
- [3] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, vol. 13, no. 2, pp. 260–269, 1967.
- [4] D. R. Tuohy and W. D. Potter., "A genetic algorithm for the automatic generation of playable guitar tablature," in *Proceedings of International Computer Music Conference*, 2005, p. 499–502.
- [5] M. Miura, I. Hirota, N. Hama, and M. Yanagida, "Constructing a system for finger-position determination and tablature generation for playing melodies on guitars," *Systems and Computers in Japan*, vol. 35, no. 6, pp. 10–19, 2004. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/scj.10609>
- [6] G. Hori, H. Kameoka, and S. Sagayama, "Input-output HMM applied to automatic arrangement for guitars," *Journal of Information Processing*, vol. 21, pp. 264–271, 04 2013.
- [7] G. Hori and S. Sagayama, "Minimax Viterbi algorithm for HMM-based guitar fingering decision," in *17th International Society for Music Information Retrieval Conference*, 2016. [Online]. Available: <https://api.semanticscholar.org/CorpusID:10478009>
- [8] D. Radicioni, "Computational modeling of fingering in music performance." Ph.D. dissertation, Università di Torino, Centro di Scienza Cognitiva, 2005.
- [9] K. Yazawa, K. Itoyama, and H. G. Okuno, "Automatic transcription of guitar tablature from audio signals in accordance with player's proficiency," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 3122–3126.
- [10] A. Wiggins and Y. Kim, "Guitar tablature estimation with a convolutional neural network," in *Proceedings of the 20th International Society for Music Information Retrieval Conference*. ISMIR, Nov. 2019, pp. 284–291. [Online]. Available: <https://doi.org/10.5281/zenodo.3527800>
- [11] W. T. Lu, J. Wang, and Y. Hung, "Multitrack music transcription with a time-frequency perceiver," *CoRR*, 2023, abs/2306.10785.
- [12] X. Riley, D. Edwards, and S. Dixon, "High resolution guitar transcription via domain adaptation," in *2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 1051–1055.
- [13] E. Mistler, "Generating guitar tablatures with neural networks," Master's thesis, The University of Edinburgh, School of Informatics, 2017.
- [14] P. Sarmiento, A. Kumar, C. Carr, Z. Zukowski, M. Barthelet, and Y.-H. Yang, "DadaGP: A dataset of tokenized GuitarPro songs for sequence models," in *Proc. of the 22nd Int. Soc. for Music Information Retrieval Conf.*, 2021, pp. 610–618.
- [15] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A.-R. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Annual Meeting of the Association for Computational Linguistics*, 2019. [Online]. Available: <https://api.semanticscholar.org/CorpusID:204960716>
- [16] Y.-S. Huang and Y.-H. Yang, "Pop Music Transformer: Beat-based modeling and generation of expressive pop piano compositions," in *Proceedings of the 28th ACM International Conference on Multimedia*. New York, NY, USA: Association for Computing Machinery, 2020, p. 1180–1188. [Online]. Available: <https://doi.org/10.1145/3394171.3413671>
- [17] Y. Zang, Y. Zhong, F. Cwitkowitz, and Z. Duan, "SynthTab: Leveraging synthesized data for guitar tablature transcription," in *2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024.
- [18] Q. Xi, R. M. Bittner, J. Pauwels, X. Ye, and J. P. Bello, "GuitarSet: A dataset for guitar transcription," in *Proceedings of the 19th International Society for Music Information Retrieval Conference*, Paris, France, 2018, pp. 453–460.
- [19] G. Bastas, S. Koutoupis, M. Kaliakatsos-Papakostas, V. Katsouros, and P. Maragos, "A few-sample strategy for guitar tablature transcription based on inharmonicity analysis and playability constraints," in *2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 771–775.
- [20] B. Duke and A. Salgian, "Guitar tablature generation using computer vision," in *Advances in Visual Computing: 14th International Symposium on Visual Computing (ISVC 2019), Proceedings, Part II*. Berlin, Heidelberg: Springer-Verlag, 2019, p. 247–257. [Online]. Available: https://doi.org/10.1007/978-3-030-33723-0_20

TRANSCRIPTION-BASED LYRICS EMBEDDINGS: SIMPLE EXTRACTION OF EFFECTIVE LYRICS EMBEDDINGS FROM AUDIO

Jaehun Kim Florian Henkel Camilo Landau Samuel E. Sandberg Andreas F. Ehmann
SiriusXM+Pandora, USA

firstname.lastname@siriusxm.com

ABSTRACT

The majority of Western popular music contains lyrics. Previous studies have shown that lyrics are a rich source of information and are complementary to other information sources, such as audio. One factor that hinders the research and application of lyrics on a large scale is their availability. To mitigate this, we propose the use of *transcription-based lyrics embeddings* (TLE). These estimate ‘ground-truth’ lyrics embeddings given only audio as input. Central to this approach is the use of transcripts derived from an automatic lyrics transcription (ALT) system instead of human-transcribed, ‘ground-truth’ lyrics, making them substantially more accessible. We conduct an experiment to assess the effectiveness of TLEs across various music information retrieval (MIR) tasks. Our results indicate that TLEs can improve the performance of audio embeddings alone, especially when combined, closing the gap with cases where ground-truth lyrics information is available.

1. INTRODUCTION

Lyrics play an important role in music consumption [1–3], often providing additional context to the perceived audio, such as lyrical themes and semantic meaning. As such, lyrics also have a wide range of applications in MIR, including mood/sentiment prediction [4–8], recommendation [2, 9], genre [2, 10–12] and music tag prediction [11].

However, the absence of lyrics on a large scale poses a significant challenge. While they are often available for popular music, this might not be the case for the majority of songs in a music catalog, either because they are non-existent, i.e., not yet transcribed by a human, or due to missing copyrights. Automatic lyrics transcription (ALT) systems are an important step towards alleviating this problem by directly transcribing the lyrical content from a piece of audio [13–17]. Still, these systems are not infallible and some efforts have been made to further refine the resulting (potentially faulty) transcriptions, e.g., by using large language models (LLMs) [15].

In this work, we investigate the use of lyrics embeddings on a variety of MIR downstream tasks, ranging from music tagging to recommendation. We focus on a comparison between embeddings stemming from human-transcribed or ‘ground-truth’ lyrics and their machine-transcribed counterparts, which we refer to as transcription-based lyrics embeddings (TLE) throughout this work. In particular, we are interested in the effectiveness of two TLE variants compared to audio embeddings and ‘ground-truth’ lyrics embeddings, where we assume the performance of the latter as an upper bound to TLE. To that end, we answer the following research questions:

- **RQ1** Do TLE provide useful additional information compared to audio embeddings alone?
- **RQ2** Can TLE be efficiently refined to close the gap to ‘ground-truth’ lyrics embeddings?

The remainder of the paper is structured as follows. Section 2 discusses related work on lyrics embeddings and automatic lyrics transcription. In Section 3 we introduce the concept and types of TLEs we evaluate in this work. Section 4 covers our experimental setup including choices of audio/lyrics embeddings as well as datasets and tasks. In Section 5 we investigate and discuss the aforementioned research questions. Finally, we conclude this work in Section 6 and highlight potential future work directions.

2. RELATED WORK

Extracting information from lyrics has long been studied in the MIR community. In particular, representing such information quantitatively, e.g., with feature or latent vectors, has been a strong focus. For instance, linguistic features (e.g., rhyme and stylistic features) are shown to be useful in various tasks [6, 8, 18], as well as approaches using psychologically validated dictionaries [2, 19].

For representation modelling, bag-of-words (BoW) [20] and term frequency inverse document frequency (TF-IDF) have been common and effective choices for lyrics [6, 12], which is further extended to latent document or topic modeling that has been successful in lyrics similarity estimation and exploration [21, 22], as well as genre and mood classification [6, 11, 12]. Another successful method is to employ word2vec [4, 18, 23], where lyrics documents are typically represented as the average of word vectors.

Lately, deep learning (DL) has been a popular choice for lyrics representation learning. In supervised learning,



it typically is accomplished implicitly within hidden layers via end-to-end learning, which proves to be effective on a range of downstream tasks [10, 11, 18]. More recently, LLMs have introduced self-supervised learning based latent text representations, which are shown to be effective on several MIR tasks [24, 25].

Regardless, ALT remains a challenging problem today [15, 16, 26, 27]. Along with efforts in building lyrics-specific transcription systems [16, 17], Automatic Speech Recognition (ASR) applied to the ALT task has also been shown to be effective [27–29].

3. TRANSCRIPTION-BASED LYRICS EMBEDDINGS

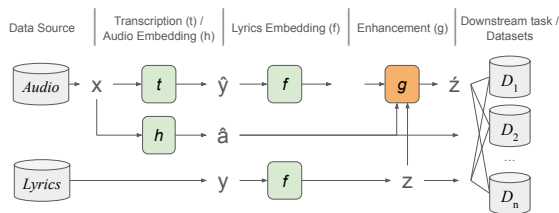


Figure 1. The diagram of proposed lyrics embedding estimation. The models in the green-colored boxes ($\{t, h, f\}$) are assumed to be pre-trained, whereas lyrics enhancement model g is trained employing embeddings obtained from those pre-trained models.

In this work, we propose a system that estimates lyrics embeddings (LE) independent of ‘ground-truth’ lyrics data $y \in \mathcal{Y}$ by only relying on audio data $x \in \mathcal{X}$, which we generally refer to as transcription-based lyrics embeddings (TLE) in the following. To achieve this, we consider several off-the-shelf pre-trained models, including an ALT model $t : \mathcal{X} \rightarrow \hat{\mathcal{Y}}$, an audio embedding model $h : \mathcal{X} \rightarrow \mathcal{A}$, and finally a lyrics embedding model $f : \mathcal{Y} \rightarrow \mathcal{Z}$.

Given the availability of a pre-trained ALT as well as word or sentence embedding models, it is straight-forward to devise a sequential system that allows one to directly input audio data and obtain high-quality lyrics embeddings that are ready to be used for a variety of downstream music tasks. We propose such an ALT based embedding as the first type of TLE, which is further referred to as \mathbf{TLE}_T and denoted as \hat{z} in Figure 1.

Despite remarkable recent improvements, ALT models are not yet completely error-free, due to the challenging nature of this task [15]. As a result the transcription $\hat{y} \in \hat{\mathcal{Y}}$, and hence an embedding computed from it may contain a certain degree of error when compared to ‘ground-truth’ lyrics embeddings. We aim to improve the fidelity of \mathbf{TLE}_T by introducing an ‘enhancement’ model which regresses to the ground-truth lyrics embeddings from noisy transcription-based embeddings by using audio embeddings as an additional input. In the following we refer to this approach as \mathbf{TLE}_R (denoted as \hat{z} in Figure 1).

Given pre-configured audio $a \in \mathcal{A} \subset \mathbb{R}^{d_a}$ and lyrics embedding $z \in \mathcal{Z} \subset \mathbb{R}^{d_z}$ spaces, the main goal of ‘enhancement’ is to find a function $g : \Phi \rightarrow \mathcal{Z}$ which maps

the concatenated audio-lyrics embedding $\phi = [a; \hat{z}] \in \Phi \subset \mathbb{R}^{(d_a+d_z)}$ to the lyrics embedding space $z' \in \mathbb{R}^{d_z}$. Specifically, we minimize the sum of squared error between the estimated and ground-truth lyrics embedding as the main learning objective:

$$\min_{\Theta} \sum_{(z,x) \sim \mathcal{D}_{\text{train}}} \|z - g(\phi; \Theta)\|^2 + \alpha \mathcal{R}(\Theta) \quad (1)$$

where Θ are the parameters of the regressor g and $\mathcal{D}_{\text{train}}$ denotes the training dataset where we have access to both the audio x and lyrics y as well as their corresponding embeddings a and z . Finally, \mathcal{R} is the regularizer for the parameters Θ which is controlled by coefficient α .

4. EXPERIMENTAL SETUP

The main hypotheses correspond to each RQ: 1) \mathbf{TLE}_T effectively provides lyrics information that is complementary to audio 2) \mathbf{TLE}_R improves the effect of \mathbf{TLE}_T . Concretely, we design an experiment comparing the performance of three treatments, \mathbf{LE} , \mathbf{TLE}_T , and \mathbf{TLE}_R , on relevant downstream tasks, with respect to a range of lyrics and audio embeddings. In the experiment, we define a treatment as a scenario where a single type of (transcription-based) lyrics embeddings is employed to represent text information, both in the training and testing phase of the machine learning (ML) experiment.¹ The rest of this section describes each component of the experimental design.

4.1 Machine Transcription

Similar to [15], we rely on a Whisper-based model [28] to transcribe the lyrics of a song from its audio recording. In contrast to [15], we do not perform a correction step in the form of ChatGPT², as this would be too costly on a large scale. Instead we directly create embeddings from the potentially faulty transcriptions (\mathbf{TLE}_T) and subsequently try to improve the embeddings using a learned correction function (\mathbf{TLE}_R).

Considering that we aim to transcribe a large set of audio recordings (see Table 1), we employ Distil-Whisper for an efficient transcription process without significant performance losses [32, 33]. As suggested in [15] we use “lyrics:” as a prefix prompt.

4.2 Embeddings

In the following, we introduce the different embeddings for each modality used in our experiments. While each embedding is tested separately, we also test *combined cases*, where both audio and lyrics embeddings are provided in the downstream task as a concatenated embedding vector.

¹ We do not consider the scenarios where different treatments are used in training and testing phase to control for a possible data drift [30, 31] and to simplify the experimental design.

² <https://openai.com/blog/chatgpt>

4.2.1 Lyrics Embeddings

We consider three text embeddings, ranging from conventional to more modern transformer-based embeddings to ensure the generality of the study.

Bag-of-Words embeddings (BE): BE embeddings serve as the baseline lyrics embedding approach within the experimental design. Unless the lyrics data is pre-tokenized by words such as the MSD-MusiXmatch (MSD-MXM) dataset [34], we employ the Byte-Pair Encoding (BPE) tokenization [35] instead of actual words. The resulting representation is a sparse song-token count matrix on which we apply TF-IDF [36] and randomized singular value decomposition (rSVD) [37] with a dimensionality of $d = 300$ to subsequently obtain a dense, low-rank vector representation of each lyrics.

Wasserstein embeddings (WE): WE embeddings are learned by applying linear optimal transport which minimizes the Wasserstein distance between distributions of the learned embeddings and given reference vectors [38]. For the reference vectors, we train token embeddings using their co-occurrence matrix. This provides token-to-token transition frequency information on top of the document-token frequency which is the only information source to BE. We choose an embedding dimensionality of $d = 300$.

Sentence BERT embeddings (sBERT): We use a pre-trained sentence BERT model [39], which is fine-tuned using a general language model called MPNet [40, 41]. In particular, the fine-tuning training involved a large scale text corpora to effectively estimate the semantic similarity between paraphrased sentences. Such a property can be crucial for lyrics data, which often is highly abstract and irregular compared to conversational language. The embedding has a dimensionality of $d = 768$.

4.2.2 Audio Embeddings

We employ two open-source and one proprietary music audio embedding models.

OpenL3: is a video-audio multimodal representation trained using self-supervised learning. Specifically, the model encodes video and audio features in respective embeddings, and minimizes the matching error between them, assuming the best matches happen when they are extracted from the same video clip [42,43]. We employ the audio encoding sub-network from the ‘music’ variant of OpenL3 as the embedding encoder with a dimensionality of $d = 6144$ and 128-band mel spectrograms as input.

MULE: is an open-source music audio embedding model trained in a self-supervised way by using contrastive learning on MusicSet, a large-scale proprietary music audio dataset [44]. We choose this for representing a modern, generic music audio embedding which is effective on wide range of downstream tasks.

MSLE: is the supervised counterpart to MULE where the music labels of MusicSet are used for its supervised learning [44]. We employ it for the proprietary datasets (i.e., InternalLT, InternalRec, see following section). Both embeddings have the same dimensionality $d = 1728$.

4.3 Tasks and Datasets

4.3.1 Automatic Music Tagging (AMT)

AMT has been a popular downstream task in MIR [45]. While there are several datasets [34, 46–48], few of them focus on lyrics specifically. To measure the effect of lyrics more clearly, we devise a subset of the Million Song Dataset (MSD) for tagging [34] that is more relevant for lyrics data, which we refer to as **MSDSnippetLT**.

It is composed as the subset of social tags that MSD provides and that are specifically relevant to the lyrics’ subject matter and language. It involves a machine-assisted tag selection process, where we first identify lyric-relevant tags by ranking MSD tags using the correlation with approximately a dozen privately-curated lyric-related tags and language metadata, with songs matched to an annotated proprietary music catalog. Among the top 200 MSD tags per each proprietary lyrics tag, three researchers voted³ for a final subset based on the following selection rules: 1) the MSD tag has to be clearly related to the targeted proprietary lyrics tag, 2) the MSD tag is not a music genre, 3) the MSD tag is not an artist. After filtering songs that map to the MSD-MXM subset which provides lyrics data for a subset of MSD songs, the resulting dataset contains 74,545 MSD songs and 87 unique tags in total, where approximately half of them center around the lyrical subject (i.e., “melancholy”, “political”), while the other half are related to the lyrics’ language (i.e., “british”, “Español”). We also experiment with a proprietary subset that we refer to as **MSDFullLT** where we have access to the full lyrics. The dataset consists of 35,264 songs and is a complete subset of MSDSnippetLT. We hypothesize that the dataset can provide useful insights on the effect of incompleteness of the snippet/preview lyrics.

Additionally, we experiment with two popular tagging datasets and one proprietary lyrics subject tagging dataset: **MSDSnippetMT** is a subset of the popular MSD tagging dataset, [34] where we select the commonly used 50 tags [45] to compare to the MSDSnippetLT dataset. As we have to consider the availability of lyrics within MSD, the resulting subset includes a total of 68,363 songs. We further test with **JamendoMood** dataset which is a subset of the MTG-Jamendo dataset [47] specifically focused on music mood. The main purpose of this dataset is to show how effective TLEs are for general music mood tagging when ‘ground-truth’ lyrics are not available. It contains 17,982 songs annotated with 56 music mood tags. Finally, **InternalLT** is the subset of a proprietary lyrics-subject dataset providing a set of high-quality lyrics-subject tags as well as full ‘ground-truth’ lyrics.

We apply 5-fold cross validation for all datasets except JamendoMood, where we use the provided pre-defined split. The model performance is evaluated by the sample-weighted mean average precision (wmAP) averaged across all tags. The main motivation of applying sample weights

³ A weighted majority voting is conducted where one of three researchers has three times larger weight than the other two, considering the substantial musical experience and training. For further details on the dataset creation, we kindly refer readers to the supplementary material.

Dataset	task	#songs	#tags	text	audio
MSDSnippetMT	music tagging	68,363	50	BoW5k	preview
MSDSnippetLT	lyrics tagging	75,545	87	BoW5k	preview
MSDFullLT	lyrics tagging	35,264	87	full-text	preview
JamendoMood	mood tagging	17,982	56	N/A	full-audio
InternalLT	lyrics tagging	51,240	15	full-text	full-audio
MSDSnippetRec	RecSys	112,769	N/A	BoW5k	preview
InternalRec	RecSys	138,984	N/A	full-text	full-audio

Table 1. Details on the datasets.

is that the majority of the datasets, except JamendoMood, provide the tagging confidence values, which is useful to both training and evaluating the task. The sample weight $w_{i,j} \in [0, 1]$ is defined as the normalized confidence value for the observed annotation of tag j on song i . For all other pairs of i and j (unannotated tags on songs) we set it to 1.

4.3.2 Music Recommendation

We further explore the effectiveness of lyrics embeddings within a music recommendation system (RecSys) problem. To maximize the effect of music content in a RecSys task, we experiment with the ‘item cold-start’ scenario; a subset of songs lack user interactions (e.g., new releases) hence a content-based recommendation is more effective than collaborative filtering [49, 50].

The dataset consists of triplets of {user, song, listening count} which is translated into a user-song matrix. The interaction data is split into five sets of *train*, *validation* and *test* set by songs in approximately 3:1:1 ratio via 5-fold cross-validation. The user-song interactions within the training (song) set is assumed as ‘observed’ and thus can be used as the training data, while those within the test (song) set are treated as ‘future’ interactions which the recommendation system is expected to rank higher. An effective measure commonly used is the binary normalized discounted cumulative gain (nDCG) [51] applied on a truncated list of the top 500 recommended songs.⁴

We employ two datasets for the RecSys task: MSD-Echonest subset⁵ is a popular recommendation dataset which contains {user, song, listening count} triplets. We derived a subset by including songs overlapping with the MSD-MXM subset only, which we refer to as **MSDSnippetRec**. We apply 5-core filtering, i.e., we filter out users who interacted with less than or equal to five unique songs, and vice versa. Similarly, we derived a subset of proprietary streaming listening data in the aforementioned format and apply the same pre-processing steps. We refer to this dataset as **InternalRec**.

4.3.3 Pre-processing on Text Representation

The subset of datasets involving MSD-MXM data only provide a pre-tokenized BoW representation, while for the rest we have access to a natural text representation of lyrics, except for JamendoMood where we do not have access to any ‘ground-truth’ lyrics. We refer to the pre-tokenized BoW representation as *BoW5K* as it specifically

is limited to the 5000 most frequent words. This applies to MSDSnippetLT, MSDSnippetMT, MSDSnippetRec.

Furthermore, as the transcription of music previews tend to be substantially shorter, the BoW5K representation of those has less counts compared to the one provided by MSD-MXM, which is extracted from the full-text lyrics. To correct this bias, we apply the following adjustment to the transcription based word count a :

$$\tilde{a}_{i,b} = \tilde{N}_i(\gamma p_{i,b}^a + (1 - \gamma)p_{i,b}^{\text{prior}}) \quad (2)$$

where $p_{i,b}^a$ denotes the normalized frequency of a word b on the i th lyrics based on the transcription, while $p_{i,b}^{\text{prior}}$ represents the global probability of a word b based on the MSD-MXM corpus. $\tilde{N}_i = r_i N_i$ denotes the estimated word count of the full text based on the length ratio r_i between full audio and snippet audio, and the observed word count from the transcription N_i . Based on a preliminary study, we choose the mixing coefficient $\gamma = 0.8$ which yielded the best adjustment quality.⁶

Given that MSDFullLT, InternalLT, InternalRec, and JamendoMood (via transcription) directly provide full text lyrics for the embedding encoding, they do not require any of the aforementioned pre-processing steps. An overview of the datasets can be found in Table 1.

4.4 Experimental Setup Details

4.4.1 Lyrics Embedding Models

Unlike sBERT, for which we use a pre-trained model, we train BE and WE models either with the MSD-MXM dataset or a proprietary lyrics corpus.⁷ As discussed in section 4.3.3, downstream task datasets based on MSD-MXM are pre-processed with the BoW5K representation, which lacks the token sequential dependency information. As WE requires the reference embeddings where typically pre-trained token/word embeddings are used, we employ *glove-840B* [52] word embeddings. For training BE and WE embeddings, we only use half of the songs uniformly sampled from MSD-MXM (118, 831/237, 662) to consider the scenario where lyrics are only available for a subset of songs. For datasets with the full texts available, we employ BE and WE pre-trained on a subset of a proprietary lyrics corpus containing 3 million unique lyrics.

⁴ For efficient evaluation, we compute estimates per fold by averaging nDCG over 5 randomly sampled subsets of 3000 users. It is shown that the estimation error is marginal, not impacting the overall conclusion.

⁵ <http://millionsongdataset.com/tasteprofile/>

⁶ The BoW5K matrix becomes dense after this adjustment, which still is tractable for computing BE and WE, due to the word truncation at 5000. However, for a large scale dataset, we suggest to set $\gamma = 1$, which disregards the prior but significantly improves computational efficiency.

⁷ We use implementations from the vectorizers package.

4.4.2 Regression Model for TLE_R

For the enhancement model for TLE_R , we apply multivariate linear ridge regression where the regularizer $\mathcal{R}(\Theta) = \|\Theta\|$ and the optimal α is selected from the range $\{10^p : p = [-6, -5, \dots, 5, 6]\}$ via cross-validation.

4.4.3 Downstream Task Pre-processing & Models

We apply standardization followed by Principal Component Analysis (PCA) to embeddings at 99.9% explained variance ratio with whitening. This is especially useful for combined audio-lyrics embeddings in order to balance the contribution of each modality.

For *Tagging* tasks, we apply ridge logistic regression, populated per tag to handle the multi-label classification problem. The regularization coefficient is found by cross-validation, from the same range used for the regressor described in Section 4.4.2.

For the *RecSys* task, we employ item K-Nearest Neighbor (itemKNN) [53]. For each song, it computes and caches the K most similar songs by measuring cosine distances between song embedding vectors, which results in a sparse song-song similarity matrix. Later, it serves songs that are most similar to the users previously listened songs by employing this similarity matrix. Finally, the optimal K is found by cross validation per fold in each feature/dataset combination from the range of [20, 50, 100, 200, 500, 1000, 2000].

5. RESULTS & DISCUSSION

5.1 Are LE in general useful for MIR tasks?

Although our main focus is the effectiveness of TLEs on MIR tasks, we briefly discuss its ideal counterpart, LE. Our main interest is whether LE outperforms the baseline scenario where *only the audio embedding is used*, compared to scenarios where LE is either used alone or in combination with audio embeddings for downstream tasks. As Figure 2 suggests, LE (round points in pink) outperforms the baseline (dashed horizontal line) particularly when the task is lyrics focused (i.e., MSDSnippetLT, MSDFullLT, InternalLT), or when the combined lyrics and audio embedding is given to downstream task models (i.e., the three “+Audio” columns to the right of each grouping). It is notable that a performance improvement is observed on most of the tagging datasets and one RecSys scenario (i.e., InternalRec using MSLE) when LE is combined with audio embeddings. However, overall we observe a smaller effect for RecSys tasks. We assume that this is due to the smaller relative effect of LE against the baseline in those cases.

Comparing audio baselines, OpenL3 performs worse than MULE and MSLE on most tagging tasks (except MSDFullLT), while performing better in RecSys tasks. Despite these differences, we observe similar trends regarding the performance of LEs compared to those baselines.

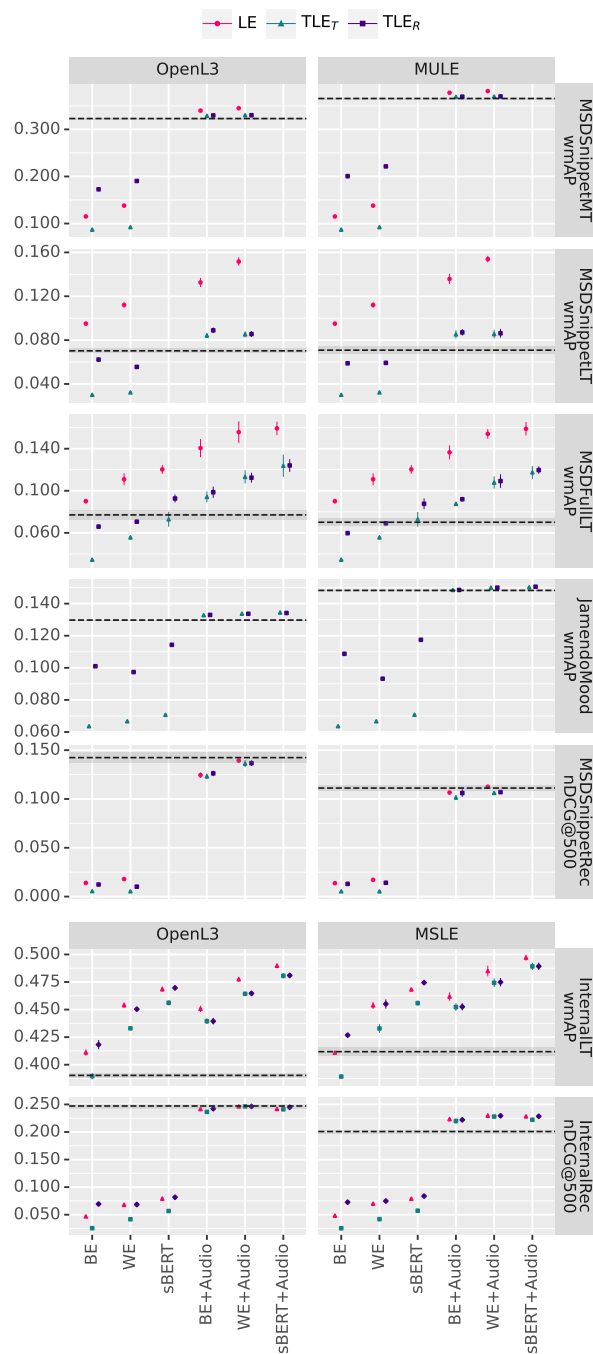


Figure 2. Each sub-figure corresponds to one dataset (row) and audio embedding (column). x and y axes represent embedding combinations and performance measures per task, respectively. Dashed horizontal lines and the shaded gray area in each figure represents the average performance and confidence interval when only the audio embedding is used. Each other point and vertical bar indicates the average performance and confidence interval of an embedding combination. We set confidence intervals at 95%.

5.2 Does TLE_T provide complementary information to audio embeddings?

In practice, the confirmed effectiveness of LE is unlikely to be helpful due to limited access to ‘ground-truth’ lyrics. Regardless, our results indicate that TLE_T can also achieve better testing performance compared to audio-only base-

lines, similar to LE.

We observe degradation of performance compared to LE in most of the cases, which is expected due to the transcription error of the ALT process. The error can be seen in Figure 3, where we measure the cosine similarity between corresponding pairs of LE and TLEs. This suggests that the transcription error can be severe such that the cosine similarity of a large number of pairs approaches 0 (i.e., MSDSnippetLT, MSDFullLT).

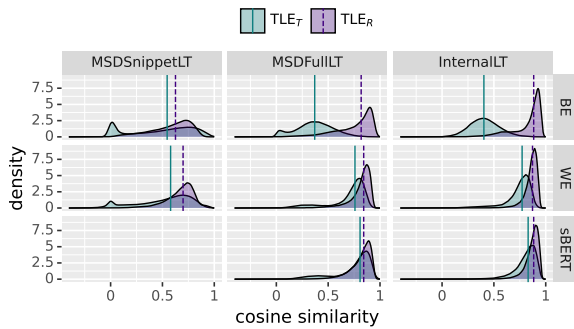


Figure 3. Distribution of cosine similarities measured between pairs of TLE_T – LE and TLE_R – LE, respectively. Each sub-figure represents the result per dataset and lyrics embeddings tested. Vertical bars denote the median.

However, TLE_T , even with such transcription errors and hence a loss of fidelity in the resulting embeddings, still provides meaningful performance gains when compared to an audio-embedding-only scenario. This is especially true on lyrics focused tasks or when combining the audio and (noisy) lyrics embeddings. In JamendoMood and MSDSnippetMT, where the lyrics data is not available upfront or the task is not focused on lyrics, we still observe that TLE_T combined with audio features outperforms audio embeddings alone.

This implies that one can build lyrics-based ML systems that can be applied to all the songs in the catalog of interest, with an expectation of a performance gain compared to models solely dependent on audio embeddings.

Comparing the performance of TLE_T between MSDSnippetLT and MSDFullLT, it is notable that the truncation of transcribed lyrics influences the effectiveness of the resulting embeddings. This suggests that providing full-length lyrics transcripts where possible is important.

5.3 Does TLE_R further improve TLE_T ?

Next, we focus on TLE_R which applies regression on top of TLE_T . Ideally, we would expect that the regression improves the fidelity of TLE_T , which likely results in an improved downstream task performance. First, we can confirm that the regression indeed improves the fidelity, as suggested by Figure 3. Measured on the testing samples of matching pairs of LE and TLE_R , the average cosine similarity is improved in all the cases. The effect is more obvious when the initial TLE_T has lower fidelity (i.e., BE, MSDSnippetLT). This indicates that the regression does increase the fidelity to some degree.

However, on the downstream tasks, the effect is not as consistent as in the cosine similarity (fidelity) result. In the case where the audio embedding is not included as input, the result indicates that TLE_R improves the downstream performance over TLE_T , sometimes even outperforming corresponding LEs (i.e., MSDSnippetMT, InternalLT). However, once combined with audio embeddings, the effect is not as distinct. While overall a small positive effect is observed in the RecSys datasets, the effect in the tagging datasets seems to be less clear, with TLE_R generally performing on par with TLE_T .

One explanation could be that the concatenation of audio embeddings for downstream tasks would eventually provide the same degree of audio information for TLE_T as already provided for TLE_R . The regression of TLE_R is conditioned both by TLE_T and the audio embedding, and thus would likely inherit the audio information. This is a possible explanation for the cases where TLE_R outperforms both LE and TLE_T . Similarly, TLE_T combined with the audio embedding explicitly fusing the two modalities via concatenation, shows performance that is on par with TLE_R in most of the cases.

6. CONCLUSION & FUTURE WORK

In this work, we introduce and assess transcription-based lyrics embeddings which tackles the problem of lyrics availability. An experiment is conducted to evaluate the effectiveness of TLEs in popular MIR downstream tasks, assessed against two comparisons, namely ‘ground-truth’ lyrics embeddings and audio embeddings. The result indicates that TLEs perform generally in between these two contenders, especially when combined with audio embedding and on the lyrics-focused tasks. This implies that TLEs can be an effective approach to be applied in lyrics-relevant MIR tasks where lyrics are often unavailable.

In particular, our results suggest that TLE_T is a simple, yet effective method for various downstream tasks when combined with audio embeddings. It is shown to complement audio embeddings by improving performance when combined with them. These gains can be achieved by using only off-the-shelf pre-trained models, while not requiring any access to ‘ground-truth’ lyrics whatsoever. Additionally, this approach does not require any subsequent refinement processes as is the case with TLE_R .

Furthermore, we identify some areas of exploration by which TLE could be improved: 1) end-to-end learning that directly associates the audio and LE to potentially improve the quality of TLE_R and avoids the transcription process, 2) instead of a simple linear regression model, more advanced methods such as semi-supervised learning [54] could further improve the fidelity of TLE_R . Additionally, 3) using context vectors from DL-based ALT models could be a viable alternative TLE, which bypasses the lyrics text embedding models. Finally, 4) while not the main focus due to the prevalence of English language in the data, multi-lingual transcription and embedding models could further improve the results in a more general setup.

7. REFERENCES

- [1] A. M. Demetriou, A. Jansson, A. Kumar, and R. M. Bittner, "Vocals in Music Matter: the Relevance of Vocals in the Minds of Listeners," in *Proceedings of the 19th International Society for Music Information Retrieval Conference*, 2018, pp. 514–520.
- [2] J. Kim, A. M. Demetriou, S. Manolios, M. S. Tavella, and C. C. Liem, "Butter Lyrics Over Hominy Grit: Comparing Audio and Psychology-Based Text Features in MIR Tasks." in *Proceedings of the 21st International Society for Music Information Retrieval Conference*, 2020, pp. 861–868.
- [3] K. Choi, J. H. Lee, X. Hu, and J. S. Downie, "Music subject classification based on lyrics and user interpretations," in *Proceedings of the 2016 Annual Meeting of the Association for Information Science and Technology*, vol. 53, no. 1, 2016, pp. 1–10.
- [4] X. Hu and J. S. Downie, "When Lyrics Outperform Audio for Music Mood Classification: A Feature Analysis," in *Proceedings of the 11th International Society for Music Information Retrieval Conference*, 2010, pp. 619–624.
- [5] S. Naseri, S. Reddy, J. Correia, J. Karlgren, and R. Jones, "The Contribution of Lyrics and Acoustics to Collaborative Understanding of Mood," in *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 16, 2022, pp. 687–698.
- [6] R. Mayer, R. Neumayer, and A. Rauber, "Rhyme and Style Features for Musical Genre Classification by Song Lyrics," in *Proceedings of the 9th International Conference on Music Information Retrieval*, 2008, pp. 337–342.
- [7] K. Watanabe and M. Goto, "Query-by-blending: A music exploration system blending latent vector representations of lyric word, song audio, and artist," in *Proceedings of the 20th International Society for Music Information Retrieval Conference*, 2019, pp. 144–151.
- [8] A. G. Smith, C. X. S. Zee, and A. L. Uitdenbogerd, "In your eyes: Identifying clichés in song lyrics," in *Proceedings of the Australasian Language Technology Association Workshop*, 2012, pp. 88–96.
- [9] P. Knees and M. Schedl, "A Survey of Music Similarity and Recommendation from Music Context Data," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 10, no. 1, pp. 1–21, 2013.
- [10] A. Tsaptsinos, "Lyrics-based music genre classification using a hierarchical attention network," in *Proceedings of the 18th International Society for Music Information Retrieval Conference*, 2017, pp. 694–701.
- [11] M. McVicar, B. D. Giorgi, B. Dundar, and M. Mauch, "Lyric document embeddings for music tagging," *arXiv preprint arXiv:2112.11436*, 2022.
- [12] C. Laurier, J. Grivolla, and P. Herrera, "Multimodal Music Mood Classification Using Audio and Lyrics," in *Proceedings of the 7th International Conference on Machine Learning and Applications*, 2008, pp. 688–693.
- [13] X. Gao, C. Gupta, and H. Li, "Genre-conditioned acoustic models for automatic lyrics transcription of polyphonic music," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2022, pp. 791–795.
- [14] L. Ou, X. Gu, and Y. Wang, "Transfer learning of wav2vec 2.0 for automatic lyric transcription," in *Proceedings of the 23rd International Society for Music Information Retrieval Conference*, 2022, pp. 891–899.
- [15] L. Zhuo, R. Yuan, J. Pan, Y. Ma, Y. Li, G. Zhang, S. Liu, R. Dannenberg, J. Fu, C. Lin *et al.*, "LyricWhiz: Robust Multilingual Lyrics Transcription by Whispering to ChatGPT," in *Proceedings of the 24th International Society for Music Information Retrieval Conference*, 2023, pp. 343–351.
- [16] T. Deng, E. Nakamura, and K. Yoshii, "End-to-end lyrics transcription informed by pitch and onset estimation," in *Proceedings of the 23rd International Society for Music Information Retrieval Conference*, 2022, pp. 633–639.
- [17] E. Demirel, S. Ahlbäck, and S. Dixon, "Mstre-net: Multistreaming acoustic modeling for automatic lyrics transcription," in *Proceedings of the 22nd International Society for Music Information Retrieval Conference, ISMIR 2021, Online, November 7-12, 2021*, 2021, pp. 151–158.
- [18] K. Watanabe and M. Goto, "A chorus-section detection method for lyrics text," in *Proceedings of the 21th International Society for Music Information Retrieval Conference*, 2020, pp. 351–359.
- [19] D. Yang and W. Lee, "Music emotion identification from lyrics," in *Proceedings of the 11th IEEE International Symposium on Multimedia*, 2009, pp. 624–629.
- [20] Z. S. Harris, "Distributional structure," *WORD*, vol. 10, no. 2-3, pp. 146–162, 1954.
- [21] S. Sasaki, K. Yoshii, T. Nakano, M. Goto, and S. Morishima, "Lyricsradar: A lyrics retrieval system based on latent topics of lyrics," in *Proceedings of the 15th International Society for Music Information Retrieval Conference*, 2014, pp. 585–590.
- [22] B. Logan, A. Kositsky, and P. J. Moreno, "Semantic analysis of song lyrics," in *Proceedings of the 2004 IEEE International Conference on Multimedia and Expo*, 2004, pp. 827–830.

- [23] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” in *Workshop Track Proceedings of the 1st International Conference on Learning Representations*, 2013.
- [24] P. Donnelly and A. Beery, “Evaluating Large-Language Models for Dimensional Music Emotion Prediction from Social Media Discourse,” in *Proceedings of the 5th International Conference on Natural Language and Speech Processing*, 2022, pp. 242–250.
- [25] Q. Huang, A. Jansen, J. Lee, R. Ganti, J. Y. Li, and D. P. W. Ellis, “MuLan: A Joint Embedding of Music Audio and Natural Language,” in *Proceedings of the 23rd International Society for Music Information Retrieval Conference*, 2022, pp. 559–566.
- [26] H. Fujihara, M. Goto, J. Ogata, and H. G. Okuno, “LyricSynchronizer: Automatic Synchronization System Between Musical Audio Signals and Lyrics,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 6, pp. 1252–1261, 2011.
- [27] C. Wang, R. Lyu, and Y. Chiang, “An automatic singing transcription system with multilingual singing lyric recognizer and robust melody tracker,” in *Proceedings of the 8th European Conference on Speech Communication and Technology*, 2003, pp. 1197–1200.
- [28] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. Mcleavey, and I. Sutskever, “Robust speech recognition via large-scale weak supervision,” in *Proceedings of the 40th International Conference on Machine Learning*, 2023, pp. 28 492–28 518.
- [29] O. Cífka, C. Dimitriou, C. Wang, H. Schreiber, L. Miner, and F. Stöter, “Jam-alt: A formatting-aware lyrics transcription benchmark,” *arXiv preprint arXiv:2311.13987*, 2023.
- [30] A. Mallick, K. Hsieh, B. Arzani, and G. Joshi, “Matchmaker: Data drift mitigation in machine learning for large-scale systems,” in *Proceedings of Machine Learning and Systems*, 2022, pp. 77–94.
- [31] S. Ackerman, E. Farchi, O. Raz, M. Zalmanovici, and P. Dube, “Detection of data drift and outliers affecting machine learning model performance over time,” in *Proceedings of the Joint Statistical Meetings Conference*, 2020, pp. 144–160.
- [32] S. Gandhi, P. von Platen, and A. M. Rush, “Distil-Whisper: Robust Knowledge Distillation via Large-Scale Pseudo Labelling,” *arXiv preprint arXiv:2311.00430*, 2023.
- [33] distill-whisper/distill-large-v2. <https://huggingface.co/distil-whisper/distil-large-v2>. Accessed: 2024-07-30.
- [34] T. Bertin-Mahieux, D. P. W. Ellis, B. Whitman, and P. Lamere, “The million song dataset,” in *Proceedings of the 12th International Society for Music Information Retrieval Conference*, 2011, pp. 591–596.
- [35] P. Gage, “A new algorithm for data compression,” *The C Users Journal*, vol. 12, no. 2, pp. 23–38, 1994.
- [36] A. Rajaraman and J. D. Ullman, *Data Mining*. Cambridge University Press, 2011, p. 1–17.
- [37] N. Halko, P. Martinsson, and J. A. Tropp, “Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions,” *SIAM Review*, vol. 53, no. 2, pp. 217–288, 2011.
- [38] C. Moosmüller and A. Cloninger, “Linear optimal transport embedding: provable Wasserstein classification for certain rigid transformations and perturbations,” *Information and Inference: A Journal of the IMA*, vol. 12, no. 1, pp. 363–389, 09 2022.
- [39] N. Reimers and I. Gurevych, “Sentence-bert: Sentence embeddings using siamese bert-networks,” in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, 2019, pp. 3982–3992.
- [40] K. Song, X. Tan, T. Qin, J. Lu, and T. Liu, “MPNet: Masked and Permuted Pre-training for Language Understanding,” in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 16 857–16 867.
- [41] sentence-transformers/all-mpnet-base-v2. <https://huggingface.co/sentence-transformers/all-mpnet-base-v2>. Accessed: 2024-07-30.
- [42] R. Arandjelovic and A. Zisserman, “Look, Listen and Learn,” in *IEEE International Conference on Computer Vision*, 2017, pp. 609–617.
- [43] J. Cramer, H. Wu, J. Salamon, and J. P. Bello, “Look, Listen, and Learn More: Design Choices for Deep Audio Embeddings,” in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2019, pp. 3852–3856.
- [44] M. C. McCallum, F. Korzeniowski, S. Oramas, F. Gouyon, and A. F. Ehmann, “Supervised and Unsupervised Learning of Audio Representations for Music Understanding,” in *Proceedings of the 23rd International Society for Music Information Retrieval Conference*, 2022, pp. 256–263.
- [45] K. Choi, G. Fazekas, and M. B. Sandler, “Automatic Tagging Using Deep Convolutional Neural Networks,” in *Proceedings of the 17th International Society for Music Information Retrieval Conference*, 2016, pp. 805–811.
- [46] E. Law, K. West, M. I. Mandel, M. Bay, and J. S. Downie, “Evaluation of algorithms using games: The case of music tagging,” in *Proceedings of the 10th International Society for Music Information Retrieval Conference*, 2009, pp. 387–392.

- [47] D. Bogdanov, M. Won, P. Tovstogan, A. Porter, and X. Serra, “The MTG-Jamendo Dataset for Automatic Music Tagging,” in *Machine Learning for Music Discovery Workshop, International Conference on Machine Learning*, 2019.
- [48] M. Defferrard, K. Benzi, P. Vandergheynst, and X. Bresson, “FMA: A dataset for music analysis,” in *Proceedings of the 18th International Society for Music Information Retrieval Conference*, 2017, pp. 316–323.
- [49] A. van den Oord, S. Dieleman, and B. Schrauwen, “Deep content-based music recommendation,” in *Advances in Neural Information Processing Systems*, vol. 26, 2013, pp. 2643–2651.
- [50] S. Oramas, O. Nieto, M. Sordo, and X. Serra, “A Deep Multimodal Approach for Cold-start Music Recommendation,” in *Proceedings of the 2nd Workshop on Deep Learning for Recommender Systems*, 2017, pp. 32–37.
- [51] K. Järvelin and J. Kekäläinen, “Cumulated gain-based evaluation of IR techniques,” *ACM Transactions on Information Systems*, vol. 20, no. 4, pp. 422–446, 2002.
- [52] J. Pennington, R. Socher, and C. D. Manning, “Glove: Global vectors for word representation,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 2014, pp. 1532–1543.
- [53] M. Deshpande and G. Karypis, “Item-based top- N recommendation algorithms,” *ACM Transactions on Information Systems*, vol. 22, no. 1, pp. 143–177, 2004.
- [54] G. Kostopoulos, S. Karlos, S. Kotsiantis, and O. Ragos, “Semi-supervised regression: A recent review,” *Journal of Intelligent and Fuzzy Systems*, vol. 35, no. 2, pp. 1483–1500, 2018.

A METHOD FOR MIDI VELOCITY ESTIMATION FOR PIANO PERFORMANCE BY A U-NET WITH ATTENTION AND FiLM

Hyon Kim

Universitat Pompeu Fabra
hyon.kim@upf.edu

Xavier Serra

Universitat Pompeu Fabra
xavier.serra@upf.edu

ABSTRACT

It is a well known fact that the dynamics in piano performance gives significant effect in expressiveness. Taking the polyphonic nature of the instrument into account, analysing information to form dynamics for each performed note has significant meaning to understand piano performance in a quantitative way. It is also a key element in an education context for piano learners.

In this study, we developed a model for estimating MIDI velocity for each note, as one of indicators to represent loudness, with a condition of score assuming educational use case, by a Deep Neural Network (DNN) utilizing a U-Net with Scaled Dot-Product Attention (Attention) and Feature-wise Linear Modulation (FiLM) conditioning. As a result, we prove that effectiveness of Attention and FiLM conditioning, improved estimation accuracy and achieved the best result among previous researches using DNNs and showed its robustness across the various domain of test data.

1. INTRODUCTION

In the realm of piano performance, the loudness of each note plays a pivotal role, alongside other factors such as tempo and precise keystrokes [1]. When analyzing piano performances, the loudness of each note is quantitatively represented by MIDI velocity. Given the polyphonic nature of the piano, measuring the overall loudness within a specific timeframe fails to provide meaningful insights into the performance's quality. Loudness can be observed at various granularities, ranging from note-level loudness and frame-level aggregated loudness to the transcription of symbolic loudness representations. Each note in a piano performance can exhibit varying loudness levels, contingent on the music's texture [2, 3]. The unique loudness of each note, especially in the context of the piano's polyphonic attributes, holds significant meaning. Mastery over the loudness of individual notes is paramount, particularly in educational settings. To hone this control, score information serves as an essential benchmark. Visualization

further enhances this educational endeavor [4]. Consequently, this study operates under the assumption that score information is accessible.

To ensure clarity in our terminology, we define "loudness" as the aggregated MIDI velocities within a designated timeframe, as gauged by an electronic piano device. In contrast, "intensity" refers to the peak value of the frequency sum for a note frame, as delineated in [5]. It is imperative to recognize that MIDI velocity does not directly correspond to the loudness as perceived by the human auditory system. Previous research has probed the relationship between MIDI velocity and perceived loudness in decibels (dB) [6,7]. These investigations consistently reveal a non-linear relationship, where an increase in MIDI velocity corresponds to a rise in perceived loudness.

Furthermore, studies such as those by [8, 9] have explored the mapping from perceptual loudness values in dB scale to dynamic symbols in piano performance, including symbols like *forte*, *mezzoforte*, *piano*, *pianissimo*, *crescendo*, and so forth. The dynamics and expressiveness of a musical composition are shaped by the loudness values attributed to each note in the score [1]. Notably, MIDI velocity offers a more nuanced prediction of loudness compared to traditional dynamic markings found in most music scores. These markings provide relative directives on the loudness with which a piece should be played. The loudness of individual notes in a piano performance can fluctuate based on the texture of the music [2,3]. Given the polyphonic characteristics of piano performances, note-level loudness is of paramount importance.

Recognizing the significance of delving into note-level loudness granularity, this study primarily centers on MIDI velocity estimation, particularly within an educational context where score information is presumed available.

2. RELATED WORK

In this section, we delve into pertinent works within the domain of Machine Learning methods and their applications for the task.

Note Level Intensity Estimation: The task of note-level loudness estimation has been the focus of multiple studies [5, 10–13]. These investigations have utilized both Non-Negative Matrix Factorization (NMF) and DNN methodologies to segregate piano performance audio into 88 distinct keys, subsequently estimating MIDI velocity or intensity for each note. This research domain can be



viewed as an extension of Automatic Music Transcription (AMT) and Music Performance Assessment, with potential applications in modeling performance expressiveness. The task of piano note-level MIDI velocity estimation is multifaceted, encompassing both a regression problem, where MIDI velocity values within the 0-127 range are estimated, and an audio classification challenge, which categorizes audio into one of the typical 88 piano keys. A limited number of studies have tackled the note-level MIDI velocity estimation task for an actual piano performance data, employing techniques like NMF [5] and DNN methods [13, 14]. The study by [5] integrated with score information to estimate note-level intensity, subsequently developing a linear regression model for note-level MIDI velocity estimation. The DNN methods [13] have sought to address the estimation challenge by incorporating AMT techniques and score conditioning. These DNN architectures amalgamate convolution blocks and GRU blocks, introducing FiLM conditioning generated by a fully connected linear layer. A diffusion model together with FiLM conditioning [14] inserts a score and performance audio information for its generative task to express note frames with MIDI velocity information. While the DNN approach did not outperform the NMF method, it marked a pioneering effort to estimate MIDI velocity using DNNs, aiming to create a model that could generalize to unseen classical music inputs, in contrast to the NMF method that optimizes parameters for individual test data. In our research, we juxtapose our findings with these preceding studies.

U-Net: The U-Net architecture incorporates layered residual connections. The concept of a residual network emerged as a solution to counteract the vanishing or exploding gradient issues encountered during the DNN training phase. U-Net has been employed for piano performance transcription, specifically for reconstructing spectrograms [15]. Its efficacy in music source separation tasks within the field of music information retrieval is well-documented. Notably, research has been conducted on a FiLM-conditioned U-Net for music source separation [16]. In our study, we leverage a U-Net structure with convolutional layers to process the mel spectrogram, a two-dimensional representation of audio. We anticipate that the U-Net will enhance classification accuracy, converting audio to the 88 piano keys.

Feature-wise Linear Modulation (FiLM): Our study employs FiLM conditioning to integrate score information, aiming to estimate note-level MIDI velocity for piano performances [17]. Historically, FiLM conditioning has found applications in image processing, yielding enhanced results when conditioned with natural language for tasks like object detection [17]. This concept has been extended to audio source separation tasks, where audio is conditioned with supplementary information such as video and scores [18]. Structurally, FiLM encompasses neural network layers that produce an affine transformation for a specified input layer. It integrates a base DNN, trained in a supervised manner, with a condition generator. This generator processes conditions, such as scores, to produce the

parameters β and γ for an element-wise affine transformation in the latent space of the base DNN. Mathematically, this is represented as: $FiLM(x) = \gamma(z) \cdot x + \beta(z)$. Here, the vector z serves as the conditional vector. Figure 1 visually represents the FiLM conditioning architecture, illustrating how the condition embedding model generates the parameters β and γ for the affine transformation on the latent vector x derived from the base DNN.

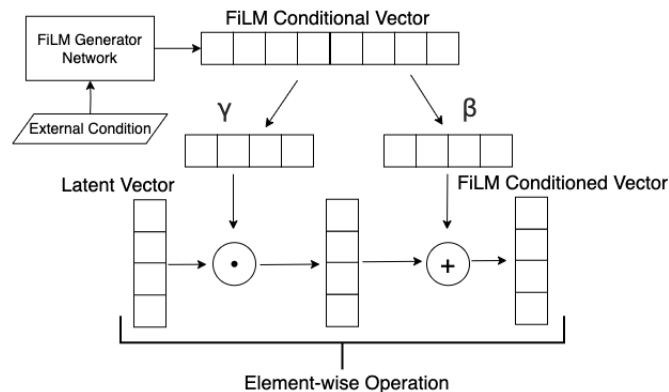


Figure 1. Visualization of FiLM operation

The Scaled Dot-Product Attention (Attention): The Attention, introduced by [19], has been instrumental in advancing the field of deep learning. This mechanism computes attention weights by scaling the dot products of queries and keys, which facilitates a dynamic focusing of the model on relevant parts of the input data. Its efficiency and simplicity allow for significant improvements in model performance by enabling the capture of long-range dependencies within the data, without the constraints imposed by previous sequence processing models. The architecture is utilised in an image processing area [20] and a speech processing area [21] together with U-Nets. This mechanism has also been applied to music information retrieval such as source separation [22] and showed its performance together with computational efficiency for the task. These researches show that the Attention mechanism works for capturing its target information from complex input data.

Our model incorporates this Attention within the U-Net architecture to leverage its proven benefits, thereby enhancing our model’s ability to understand and generate nuanced responses based on the context provided by the input sequence in a musical sense.

3. METHOD

Figure 2 illustrates the comprehensive architecture of our proposed model. Initially, the model processes audio input, transforming it into a Log Mel-frequency Spectrogram. This transformation facilitates the conversion of the waveform into an image-like format. The audio processing parameters include a window length of two seconds, a hop size of one second, and a sampling rate of 16k Hz, resulting in a model output resolution of 100 frames per second. The overarching model architecture can be cate-

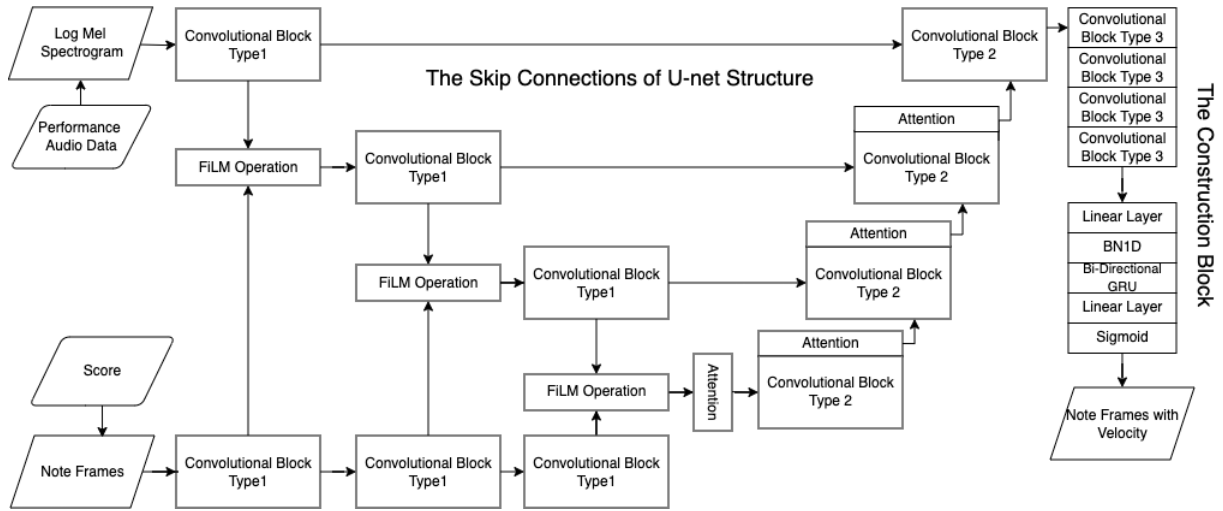


Figure 2. The entire architecture of the proposed model.

gorized into three distinct convolutional blocks, as showcased in Figure 3.

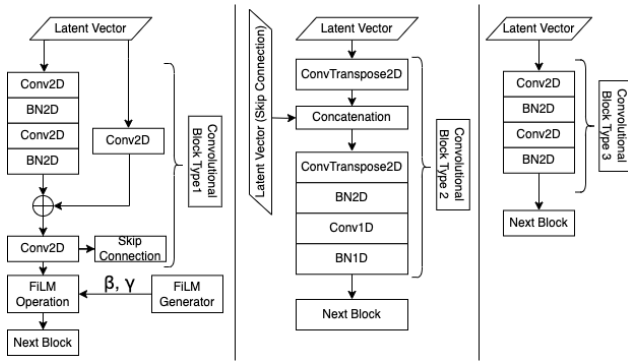


Figure 3. Schematic of the three convolutional blocks utilized in the model.

Convolutional blocks of type 1 and 2 collectively form the U-Net structure. Type 1 blocks also play a pivotal role in encoding note frame information. In this study, note frames are derived from a MIDI roll. Corresponding blocks in the encoding phase generate FiLM conditioning parameters, denoted as β and γ , for each affine transformation. Several methods for inserting FiLM parameters are described in [16]. In our model, through empirical study, we generate parameters to ensure element-wise correspondence for each latent space vector, as depicted in Figure 1. Consequently, each output from the score encoders generates twice as many parameters for the output of each block in the encoder of the U-Net.

To ensure uniformity in the processed latent features, we employ convolutional layers of the same hierarchical level to produce FiLM parameters for each layer within the U-Net. For the skip connections, non-conditioned latent vectors from each block are relayed to the corresponding type 2 block, while FiLM-conditioned latent vectors are channeled to the subsequent layer of the type 1 block.

In the decoder section of the U-Net architecture, Attention modules are incorporated before each convolutional

block type 2. This configuration enhances the network’s ability to focus on relevant features by dynamically adjusting the importance of different areas of the input image. The Attention mechanism, which calculates attention scores by scaling the dot-product of queries and keys, enables the model to prioritize specific features over others, improving the precision of MIDI velocity estimation.

The construction block consists of convolutional block type 3 followed by the block containing bi-directional GRU. It processes inputs through a sequence of layers including linear transformations for dimensionality reduction based on the input feature type, batch normalization, and a bidirectional GRU for capturing temporal dynamics. The network concludes with a fully connected layer applying a sigmoid function to output note frames with velocity information. Dropout and ReLU activations are utilized throughout to enhance performance and prevent overfitting.

For training our model, we employed the MAESTRO dataset [23]. MAESTRO is a dataset composed of about 200 hours of virtuosic piano performances captured with fine alignment (up to 3 ms) between note labels and audio waveforms. Notably, other DNN models targeting MIDI velocity estimation, such as [13] and [14], have also employed this dataset. This usage facilitates a more legitimate comparison of model performance across different studies.

Our chosen loss function, represented by Eq. 1, amalgamates the $l1$ loss and the Binary Cross-Entropy (BCE) loss. This design facilitates back-propagation of losses for both classification and regression tasks.

$$Loss = \theta \cdot l1 \text{ loss} + (1 - \theta) \cdot BCE \text{ loss} \quad (1)$$

Here, $\theta \in [0, 1]$ signifies the weight for the $l1$ and the BCE loss. For our empirical setup, we set θ to 0.5. The $l1$ loss function, as defined in Eq. 2, is articulated as:

$$l1 \text{ loss} = \frac{\sum_i |V(i)_{\text{ground truth}} - V(i)_{\text{model output}}|}{N} \quad (2)$$

In this equation, $V(i)$ represents MIDI velocity with the index i of corresponding notes between the ground truth and the model output within a specified window, while N denotes the total number of notes present in that window. Each input data point spans two seconds, with each frame encompassing 100 segments per second to depict the MIDI roll. The velocities used to compute the loss are normalized to the range $[0, 1]$ to match the scale of the BCE.

For the evaluation phase, we employed the Saarland Music Data (SMD) dataset [24]. SMD provides audio recordings along with perfectly synchronized MIDI files for various piano pieces. The pieces were performed by students of the Hochschule für Musik Saar on a hybrid acoustic/digital piano (Yamaha Disklavier). We selected 49 excerpts from this dataset, consistent with the test sets used in prior studies [5, 13, 14], ensuring a fair and comparable assessment. The model's error is quantified using the formula presented in Eq. 3:

$$Error = \frac{\sum_i |V(i)_{\text{ground truth}} - V(i)_{\text{inference}}|}{N} \quad (3)$$

In this equation, i represents individual notes, and N denotes the total number of notes accurately identified in the score. The inferred MIDI velocity is determined by the peak value within the interval of each detected and categorized velocity frame, juxtaposed with the ground truth velocity frame for the respective note. This approach is adopted because the detected velocity typically exhibits a peak followed by a decline in the estimated MIDI velocity within a note frame, mirroring the attack and decay patterns of each note's loudness. Differently from loss function, the output values are not normalised but are scaled to the range $[0, 127]$. The recall score serves as our primary evaluation metric for classification accuracy, given that the model's output is constrained by the provided score information.

4. RESULTS AND DISCUSSION

Result and Comparison: Table 1 presents the comparative outcomes of our model against previous works in the field. The proposed model consistently outperforms other DNN-based methods across all metrics, demonstrating notable improvements. The enhancements are particularly evident when comparing the best and worst outcomes of our model with those of other models. The results highlight that the U-Net designed with Attention and FiLM conditioning with score information significantly boosts performance.

Among the test set, the most favorable outcome is observed for "Bach BWV875-01 002," which recorded mean error, standard deviation, and recall values of 4.6, 3.3, and 95.6%, respectively. Conversely, "Chopin Op028-17" exhibited the least favorable results for mean error and standard deviation, with values of 16.0 and 11.9 respectively, and a recall of 87.5%. Additionally, "Ravel Jeux d'eau" demonstrated the lowest recall score in the dataset

Model	Mean	SD	Recall
DNN Based Model			
DiffVel [14]	19.7	13.1	53.0%
Convolutional Net [13]	15.1	12.3	85.8%
Proposed Model	9.9	7.8	89.7%
NMF Based Model			
Score-Informed NMF [5]	4.1	5.0	N.A.

Table 1. Comparative results of models for note-level MIDI velocity estimation with score information. SD: Standard Deviation

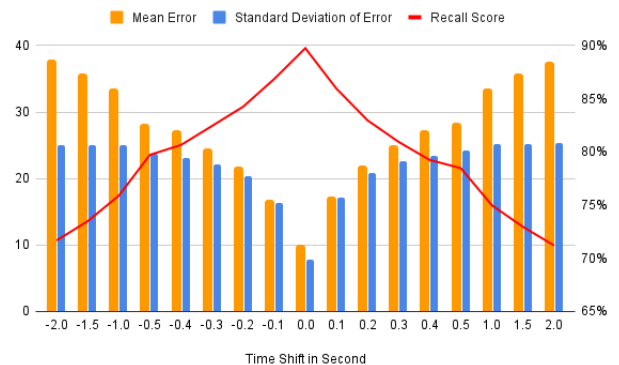


Figure 4. Mean and SD of errors for misaligned score information

at 80.7%, with corresponding mean error and standard deviation values of 12.1 and 10.2. These results illustrate the varied performance of our model across different musical pieces, underscoring its effectiveness as well as areas for potential improvement.

The analysis also highlights the strengths and weaknesses of both DNN and NMF-based methods. DNNs are capable of capturing complex relationships within the training data due to their nonlinear nature, but they are computationally demanding and require extensive data to optimize parameters effectively. In contrast, NMF-based methods, such as the one described by [5], optimize parameters for individual excerpts using score information in the test set, offering a more tailored approach. This specificity, however, can limit their generalizability compared to DNNs, which aim to develop a more generic model suitable for diverse musical excerpts. Notably, the proposed model is trained on a distinct domain, specifically a piano performance dataset different from the test set, to ensure a fair comparison and robust assessment of its performance. This strategy helps in evaluating the model's ability to generalize across different musical contexts effectively.

Misaligned Condition Insertion: In real-world applications, alignment discrepancies frequently occur between scores and their corresponding audio, affecting the accurate feeding of note frames. Figure 4 elucidates the model's sensitivity to temporal misalignments, exhibiting a correlation between the degree of time shift and the model's performance metrics.

These shifts are synthetically generated by inserting the

conditions a specified number of seconds ahead or behind each input frame, with a two-second duration per input. It is clear that misaligned data affects to the model accuracy proportionally. Addressing this misalignment, data augmentation can be employed during the training phase to acclimate the model to varying degrees of data condition misalignments, thereby enhancing its flexibility.

Nonetheless, this alignment challenge may become negligible with the integration of a dedicated note frame detection model, as delineated by models like the one in [25]. Utilizing such models for precise note frame detection allows for a subsequent, more accurate analysis of MIDI velocity estimation by the proposed model, streamlining the workflow and potentially increasing performance accuracy.

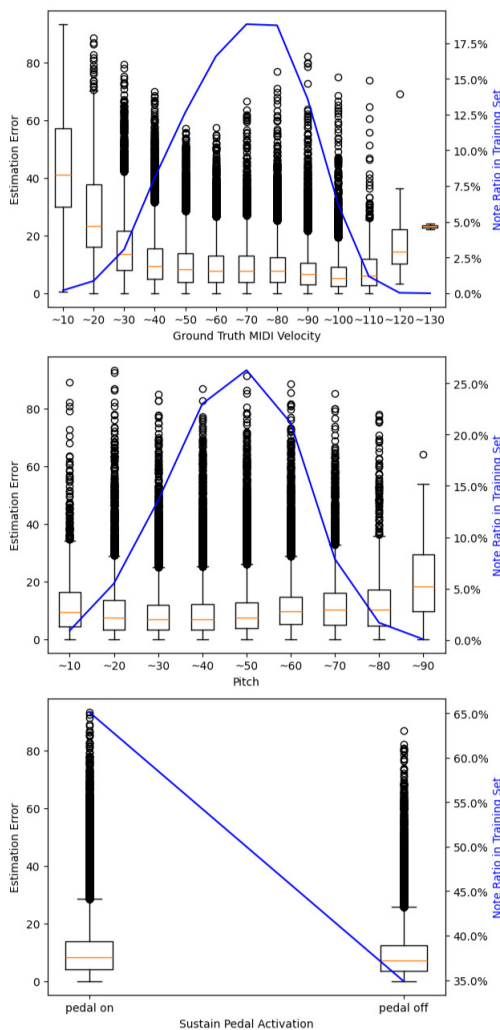


Figure 5. Error distributions based on various ground truth aspects: pitch, sustain pedal activation, and MIDI velocity intervals together with the ratio of notes appeared in the training set.

Error Analysis: Further analysis was conducted to evaluate the error distribution across different pitch groups, ground truth MIDI velocities, and sustain pedal activation states, as depicted in Figure 5. The analysis indicates that error is inversely correlated with the volume of data in the

training set: the greater the quantity of data processed by the model, the more accurate the MIDI velocity estimates, highlighting the benefits of extensive data representation.

The results further reveal that enhanced training data volumes lead to improved estimation outcomes across various data dimensions. This suggests that applying data augmentation to achieve a balanced distribution in pitch and velocity bins can result in higher estimation accuracy. However, such augmentation must maintain the musico-logical context, including harmony and expressiveness, making this a complex yet critical task for effective model training.

Ablation Study: In our ablation study, we evaluate the individual and combined contributions of FiLM conditioning and the Attention modules to our model’s performance, based on the U-Net architecture. These components were chosen for their theoretical abilities to enhance feature representation and focusing mechanisms, respectively. The study aims to clarify their roles within our proposed deep neural network architecture. We examine four configurations of our model: (i) with both FiLM and Attention (proposed model), (ii) with FiLM but without Attention, (iii) with Attention but without FiLM, and (iv) without either FiLM or Attention, as shown in Table 2.

Model Configuration	Mean	SD	Recall
With FiLM:			
With Attention	9.9	7.8	89.7%
Without Attention	10.0	7.8	89.4%
Without FiLM:			
With Attention	12.1	10.5	73.0%
Without Attention	13.0	10.5	68.5%

Table 2. Ablation Study: Detailed Performance Comparison Highlighting the Impact of FiLM Conditioning and Attention.

The ablation study highlights the significant impact of FiLM Conditioning and a relatively lesser contribution from the Attention in enhancing the performance of the proposed model. The observed synergy when integrating these modules indicates a promising avenue for future research and development in deep neural network architectures. While the Attention module improves model performance, its effectiveness is not as pronounced as that of FiLM Conditioning. This suggests that the model’s ability to concentrate on relevant features, and thereby its predictive performance, is significantly enhanced by FiLM Conditioning. Notably, we could see universal improvement on the recall score on all the excepts on the test set in any comparison among combination of (i) to (iv).

According to Table 1, the study also demonstrates that incorporating a U-Net mechanism, particularly its skip connections, can enhance accuracy for the task at hand, outperforming previous models.

Robustness of the Model: We conducted a comparative analysis against the state-of-the-art transcription model that additionally estimates the MIDI velocity [26]. The results, detailed in Table 3, indicate that our proposed

model achieves comparable performance in MIDI velocity estimation. Notably, our model demonstrates enhanced robustness across various test datasets, as evidenced by the recall scores, in comparison to the model proposed by [26] which is also trained on the MAESTRO dataset.

Model	Mean	SD	Recall
Proposed Model	9.9	7.8	89.7%
The hFT Model [26]	9.9	7.3	78.0%

Table 3. Comparison to the SOTA Transcription Model

This comparison highlights the efficacy of our model, particularly in its ability to generalize across different datasets, which is crucial for practical applications. The fact that both models yield identical mean scores for MIDI velocity estimation but our model exhibits a higher recall rate suggests our model’s capability in accurately capturing the nuances of musical expression. Furthermore, despite the slightly higher standard deviation in our model’s performance, the significantly higher recall rate underscores its robustness and reliability in diverse testing scenarios. This finding is particularly relevant for applications requiring high fidelity in musical transcription and velocity estimation, indicating a promising direction for future research and development in music transcription technologies. Also the ablation study indicates that adding FiLM conditioning can improve the model accuracy for the task, after experimental process of designing the parameter generators and methods to insert the parameters, which yields another research topic.

5. CONCLUSION AND FUTURE WORKS

In this study, we explored the complexities of MIDI velocity estimation, leveraging an U-Net architecture enriched with the Attention and FiLM conditioning to integrate score information. Our results underscore the superiority of this approach among DNN methodologies. Our empirical evaluations further attest to the pivotal role of FiLM conditioning in bolstering result accuracy. This enhancement transcends specific model architectures, with FiLM conditioning amplifying precision across various models, ranging from feed-forward designs with convolution and GRU blocks to diffusion models, combining the previous researches [13, 14]. The Attention also contributes to improve on both MIDI velocity estimation and recall score. The model also showed that comparable results towards SoTA transcription model and the robustness across the sources of test set compared to other state of the art transcription models which also estimates MIDI velocity.

Generally, FiLM conditioning has proven effective for MIDI velocity estimation tasks. Enhanced transcription of note onset, offset, and frames could further refine performance, positioning this model as a robust solution for MIDI velocity estimation across diverse datasets. This suggests that utilizing DNN models, such as the onsets and frames model proposed by [25], which demonstrates superior accuracy in note frame detection without FiLM

conditioning, could be advantageous. In situations where score data are not available, a cascaded approach can be employed: first, use a DNN for accurate note frame detection, and then leverage the detected MIDI for FiLM conditioning, circumventing the need for score-to-audio alignment.

As we look to the future, our objective is to expand the range of score information, transitioning from MIDI note frame to more comprehensive formats, MusicXML to be encoded. Such a shift is anticipated to offer increased resilience, especially in situations where achieving precise alignments poses challenges. Data augmentation on training data is also considered as crucial task for obtaining more robust estimation, as mentioned. Additionally, addressing issues such as omitted notes and extraneous notes is essential to tailor the model more effectively for educational applications, catering to both novice learners and seasoned professionals, considering currently the set up only considers the student is god enough to follow the score for performance visualization purposes.

The potential applications of this research are manifold, extending from the development of visualization tools that bolster musical communication to advanced transcription techniques. Such annotations, especially those denoting expressiveness, carry significant implications, particularly in pedagogical contexts where teacher-student interactions are crucial.

The code and model developed for this study are available upon request.

6. ACKNOWLEDGMENTS

"IA y Música: Cátedra en Inteligencia Artificial y Música" (TSI-100929-2023-1), funded by the Secretaría de Estado de Digitalización e Inteligencia Artificial, and the European Union-Next Generation EU, under the program Cátedras ENIA 2022 para la creación de cátedras universidad-empresa en IA.

7. REFERENCES

- [1] M. Grachten and G. Widmer, "Linear basis models for prediction and analysis of musical expression," *Journal of New Music Research*, vol. 41, no. 4, pp. 311–322, 2012.
- [2] W. Goebel, "Melody lead in piano performance: expressive device or artifact?" *The Journal of the Acoustical Society of America*, vol. 110, no. 1, pp. 563–72, 2001.
- [3] S. Kim, J. M. Park, S. Rhyu, J. Nam, and K. Lee, "Quantitative analysis of piano performance proficiency focusing on difference between hands," *PLoS ONE*, vol. 16, 2021.
- [4] L. F. Hamond, G. Welch, and E. Himonides, "The pedagogical use of visual feedback for enhancing dynamics in higher education piano learning and performance," *Opus*, vol. 25, no. 3, pp. 581–601, 2019.

- [5] D. Jeong, T. Kwon, and J. Nam, “Note-intensity estimation of piano recordings using coarsely aligned MIDI score,” vol. 68, no. 1, pp. 34–47, 2020, publisher: Audio Engineering Society. [Online]. Available: <https://www.aes.org/e-lib/browse.cfm?elib=20716>
- [6] R. B. Dannenberg, “The interpretation of midi velocity,” in *International Conference on Mathematics and Computing*, 2006.
- [7] Y. Qu, Y. Qin, L. Chao, H. Qian, Z. Wang, and G. Xia, “Modeling perceptual loudness of piano tone: Theory and applications,” *arXiv preprint arXiv:2209.10674*, 2022.
- [8] K. Kosta, O. F. Bandtlow, and E. Chew, “Outliers in performed loudness transitions: An analysis of chopin mazurka recordings,” in *International Conference for Music Perception and Cognition (ICMPC)*, California, USA, 2016, pp. 601–604.
- [9] K. Kosta, R. Ramírez, O. F. Bandtlow, and E. Chew, “Mapping between dynamic markings and performed loudness: a machine learning approach,” *Journal of Mathematics and Music*, vol. 10, no. 2, pp. 149–172, 2016.
- [10] S. Ewert and M. Müller, “Estimating note intensities in music recordings,” in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2011, pp. 385–388.
- [11] J. Devaney and M. Mandel, “An evaluation of score-informed methods for estimating fundamental frequency and power from polyphonic audio,” in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 181–185. [Online]. Available: <http://ieeexplore.ieee.org/document/7952142/>
- [12] D. Jeong and J. Nam, “Note intensity estimation of piano recordings by score-informed nmf,” in *Audio Engineering Society Conference: 2017 AES International Conference on Semantic Audio*. Audio Engineering Society, 2017.
- [13] H. Kim., M. Miron, and X. Serra, “Score-informed midi velocity estimation for piano performance by film conditioning,” in *Proc. Int. Conf. Sound and Music Computing*, 2023.
- [14] H. Kim and X. Serra, “Diffvel: Note-level midi velocity estimation for piano performance by a double conditioned diffusion model,” in *Proceedings of the 16th International Symposium on Computer Music Multidisciplinary Research*, Tokyo, Japan, 2023. [Online]. Available: <http://hdl.handle.net/10230/57790>
- [15] K. W. Cheuk, Y.-J. Luo, E. Benetos, and D. Herremans, “The effect of spectrogram reconstruction on automatic music transcription: An alternative approach to improve transcription accuracy,” in *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 2021, pp. 9091–9098.
- [16] G. Meseguer-Brocal and G. Peeters, “Conditioned-u-net: Introducing a control mechanism in the u-net for multiple source separations,” *arXiv preprint arXiv:1907.01277*, 2019.
- [17] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville, “Film: Visual reasoning with a general conditioning layer,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.
- [18] O. Slizovskaia, G. Haro, and E. Gómez, “Conditioned source separation for musical instrument performances,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 2083–2095, 2021.
- [19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” 2023.
- [20] O. Oktay, J. Schlemper, L. L. Folgoc, M. Lee, M. Heinrich, K. Misawa, K. Mori, S. McDonagh, N. Y. Hammerla, B. Kainz, B. Glocker, and D. Rueckert, “Attention u-net: Learning where to look for the pancreas,” 2018.
- [21] R. Giri, U. Isik, and A. Krishnaswamy, “Attention wave-u-net for speech enhancement,” in *2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2019, pp. 249–253.
- [22] T. Sgouros, A. Bousis, and N. Mitianoudis, “An efficient short-time discrete cosine transform and attentive multiresunet framework for music source separation,” *IEEE Access*, vol. 10, pp. 119 448–119 459, 2022.
- [23] C. Hawthorne, A. Stasyuk, A. Roberts, I. Simon, C.-Z. A. Huang, S. Dieleman, E. Elsen, J. Engel, and D. Eck, “Enabling factorized piano music modeling and generation with the MAE-STRO dataset,” in *International Conference on Learning Representations*, 2019. [Online]. Available: <https://openreview.net/forum?id=r11YRjC9F7>
- [24] M. Müller, V. Konz, W. Bogler, and V. Arifi-Müller, “Saarland music data (smd),” in *Proceedings of the international society for music information retrieval conference (ISMIR): late breaking session*, 2011.
- [25] C. Hawthorne, E. Elsen, J. Song, A. Roberts, I. Simon, C. Raffel, J. Engel, S. Oore, and D. Eck, “Onsets and frames: Dual-objective piano transcription,” 2018.
- [26] K. Toyama, T. Akama, Y. Ikemiya, Y. Takida, W.-H. Liao, and Y. Mitsufuji, “Automatic piano transcription with hierarchical frequency-time transformer,” in *Proceedings of the 24th International Society for Music Information Retrieval Conference*, 2023.

MUSICONGEN: RHYTHM AND CHORD CONTROL FOR TRANSFORMER-BASED TEXT-TO-MUSIC GENERATION

Yun-Han Lan^{1,2}

Wen-Yi Hsiao¹

Hao-Chung Cheng²

Yi-Hsuan Yang^{1,2}

¹ Taiwan AI Labs

² National Taiwan University

cyan0731@gmail.com, wayne391@aillabs.tw, {haochung,yhyangtw}@ntu.edu.tw

ABSTRACT

Existing text-to-music models can produce high-quality audio with great diversity. However, textual prompts alone cannot precisely control temporal musical features such as chords and rhythm of the generated music. To address this challenge, we introduce MusiConGen, a temporally-conditioned Transformer-based text-to-music model that builds upon the pretrained MusicGen framework. Our innovation lies in an efficient finetuning mechanism, tailored for consumer-grade GPUs, that integrates automatically-extracted rhythm and chords as the condition signal. During inference, the condition can either be musical features extracted from a reference audio signal, or be user-defined symbolic chord sequence, BPM, and textual prompts. Our performance evaluation on two datasets—one derived from extracted features and the other from user-created inputs—demonstrates that MusiConGen can generate realistic backing track music that aligns well with the specified conditions. We open-source the code and model checkpoints, and provide audio examples online, https://musicongen.github.io/musicongen_demo/.

1. INTRODUCTION

The realm of text-to-music generation has seen significant progress over the recent years [1–11]. These models span various genres and styles, largely leveraging textual prompts to guide the creative process. There have been two primary methodological frameworks so far. The first employs *Transformer* architectures to model audio tokens [12] derived from pre-trained audio codec models [13–15]; noted examples include MusicLM [1] and MusicGen [2]. The second employs *diffusion* models to represent audio through spectrograms or audio features, such as AudioLDM 2 [4] and JEN-1 [5].

Text-to-music generation model generally relies on the global textual conditions to guide the music generation process. Textual prompts serving as high-level conceptual guides, however, introduce a degree of ambiguity and verbosity into the music generation for describing the musi-

Model	Chord control	Rhythm control	Do not need reference audio
Coco-Mulla [6]	✓	✓	
Music ControlNet [7]		✓	✓
Ours	✓	✓	✓

Table 1. The comparison for conditions and condition type of related temporally-conditioned text-to-music models.

cal features [7]. This inherent vagueness poses a challenge in precisely controlling temporal musical features such as melody, chords and rhythm, which are crucial for music creation. Building on the success of MusicGen-melody [2] in melody control, our focus now shifts to enhancing chord and rhythm control, aiming to create a more integrated approach to music generation that captures the full spectrum of musical elements.

Table 1 tabulates two existing studies that have explored the incorporation of time-varying chord- and rhythm-related attributes in text-to-music generation. Coco-Mulla [6] is a Transformer-based model that employs a large-scale, 3.3B-parameter MusicGen model, finetuned with an adapted LLaMA-adapter [16] for chord and rhythm control. For rhythm control in particular, Coco-Mulla uses drum audio codec tokens extracted from a reference drum audio signal as a condition for guiding the music generation, thereby demanding *reference audio* for control. While it is appropriate to assume the availability of such *reference audio* in some scenarios, for broader use cases we desire to have a model that can take user-provided *text*-like inputs as well, such as the intended beats-per-minute (BPM) value (for rhythm) and the chord progression as a series of chord symbols (for chords). This function is not supported by Coco-Mulla.

The other model, Music ControlNet [7], leverages a diffusion model architecture and the adapter-based conditioning mechanism of ControlNet [17] to manipulate text-like, symbolic melody, dynamics, and rhythm conditions. This diffusion model creates a spectrogram based on the provided conditions, which is then transformed into audio using their pretrained vocoder. For musical conditions, a 12-pitch-class chromagram representation is used for the melody, combined with beat and downbeat probability curves concatenation for rhythm control, and an energy curve to adjust the dynamic volume. However, Music ControlNet does not deal with chord conditions.



© Y. Lan, W. Hsiao, H. Cheng and Y. Yang. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Y. Lan, W. Hsiao, H. Cheng and Y. Yang, “MusiConGen: Rhythm and Chord Control for Transformer-Based Text-to-Music Generation”, in *Proc. of the 25th Int. Society for Music Information Retrieval Conf.*, San Francisco, United States, 2024.

In view of the limits of the prior works, we introduce in this paper MusiConGen, a Transformer-based text-to-music model that applies temporal conditioning to enhance control over rhythm and chord. MusiConGen is finetuned from the pretrained MusicGen framework [2]. We design our temporal condition controls in a way that it supports not only musical features extracted from reference *audio* signals, but also the aforementioned user-provided *text-like* symbolic inputs such as BPM value and chord progression. For effective conditioning of such time-varying features, we propose “adaptive in-attention” conditioning by extending the in-attention mechanism proposed in the MuseMorphose model [18]. Table 1 includes a conceptual comparison of MusiConGen with existing models in terms of the conditions and their types.

In our implementation, we train MusiConGen on a dataset of *backing track music* comprising 5,000 text-audio pairs obtained from YouTube. This training utilizes beat tracking and chord recognition models to extract necessary condition signals without the need for manual labeling. We note that rhythm and chord controls are inherently critical for backing tracks, for backing tracks often do not include the primary melody and their purpose is mainly to provide accompaniment for a lead performer.

Moreover, instead of using the adapter-based finetuning methods [16, 17, 19], we apply the straightforward “direct finetuning” approach to accommodate the domain shift from general instrumental music (on which MusicGen was trained) to the intended backing track music. We leave the use of adapter-based finetuning as future work. To make our approach suited for operations on consumer-grade GPUs, we propose a mechanism referred to as “jump finetuning” instead of finetuning the full MusicGen model.

We present a comprehensive performance study involving objective and subjective evaluation using two public-domain datasets, MUSDB18 [20] and RWC-pop-100 [21]. Our evaluation demonstrates MusiConGen’s enhanced ability to offer nuanced temporal control, surpassing the original MusicGen model in producing music that aligns more faithfully with the given conditions.

The contributions of this work are two-fold. First, to our best knowledge, this work presents the first Transformer-based text-to-music generation model that follows user-provided rhythm and chords conditions, requiring no reference audio signals. Second, we present efficient training configuration allowing such a model to be built by finetuning the publicly-available MusicGen model with customer-level GPU, specifically 4x RTX-3090 in all our experiments. We open-source the code, checkpoint, and information about the training data of MusiConGen on GitHub.¹

2. BACKGROUND

2.1 Codec Models for Audio Representation

In contemporary music generation tasks, audio signals are typically compressed into more compact representations

using two main methods: Mel spectrograms and codec tokens. Mel spectrograms provide a two-dimensional time-frequency representation, adjusting the frequency axis to the Mel scale to better align with human auditory perception. Codec tokens, on the other hand, are often residual vector quantization (RVQ) tokens that are encoded from audio signals by a codec model [13–15]. Following MusicGen, we employ in our work the Encodec (32k) [14] as the pretrained codec model to encode audio data at a sample rate of 32,000 Hz. This Encodec model comprises 4 codebooks, each containing 2,048 codes, and operates at a code frame rate f_s of 50 Hz.

2.2 Classifier-Free Guidance

Classifier-free guidance [22] is a technique initially developed for diffusion models in generative modeling to enhance the quality and relevance of the outputs without the need for an external classifier. This approach involves training the generative model in both a conditional and an unconditional manner, combining the output score estimates from both methods during the inference stage. The mathematical expression is as $\nabla_x \log \tilde{p}_\theta(x|c) = (1-\gamma)\nabla_x \log p_\theta(x) + \gamma\nabla_x \log p_\theta(x|c)$. Here, γ represents the guidance scale, which adjusts the influence of the conditioning information. We perform a weighted average of $f_\theta(x, c)$ and $f_\theta(x)$ when sampling from the output logits.

2.3 Pretrained MusicGen Model

The pretrained model used in our study is a MusicGen model with 1.5B parameters, equipped with melody control (i.e., MusicGen-Melody). The melody condition employs a chromagram of 12 pitch classes at a frame rate f_M , denoted as $\mathcal{M} \in \mathbb{R}^{T_{f_M} \times 12 \times 1}$, derived from the linear spectrogram of the provided reference audio. For text encoding, the model leverages the FLAN-T5 [23] as a text encoder to generate conditioning text embeddings, represented as $\mathcal{T} \in \mathbb{R}^{T_{t5} \times d_{t5} \times 1}$. Both the melody and text conditions undergo linear projection into a D -dimensional space before being prepended to the input audio embedding. Regarding the input audio for training, audio signals are initially encoded into RVQ tokens, $X_{rvq} \in \mathbb{R}^{T_{f_s} \times 1 \times 4}$, using the pretrained Encodec model. These tokens are then formatted into a “delay pattern” [2], maintaining the same sequence length. Subsequently, an embedding lookup table, $W_{emb} \in \mathbb{R}^{N \times D \times 4}$, where N represents for numbers of codes in a codebook, is used to represent the associated codes, summing contributions from each codebook of X_{rvq} to form the audio embedding $X_{emb} \in \mathbb{R}^{T_{f_s} \times D \times 1}$. The input representation is then fed to the self-attention layers via additive sinusoidal encoding.

3. METHODOLOGY

Our method seeks to efficiently finetune the foundational MusicGen model using time-varying symbolic rhythm and chord conditions as guiding conditions. To achieve this, we must carefully consider both the representation of these conditions and the finetuning mechanism as follows:

¹ <https://github.com/Cyan0731/MusiConGen>

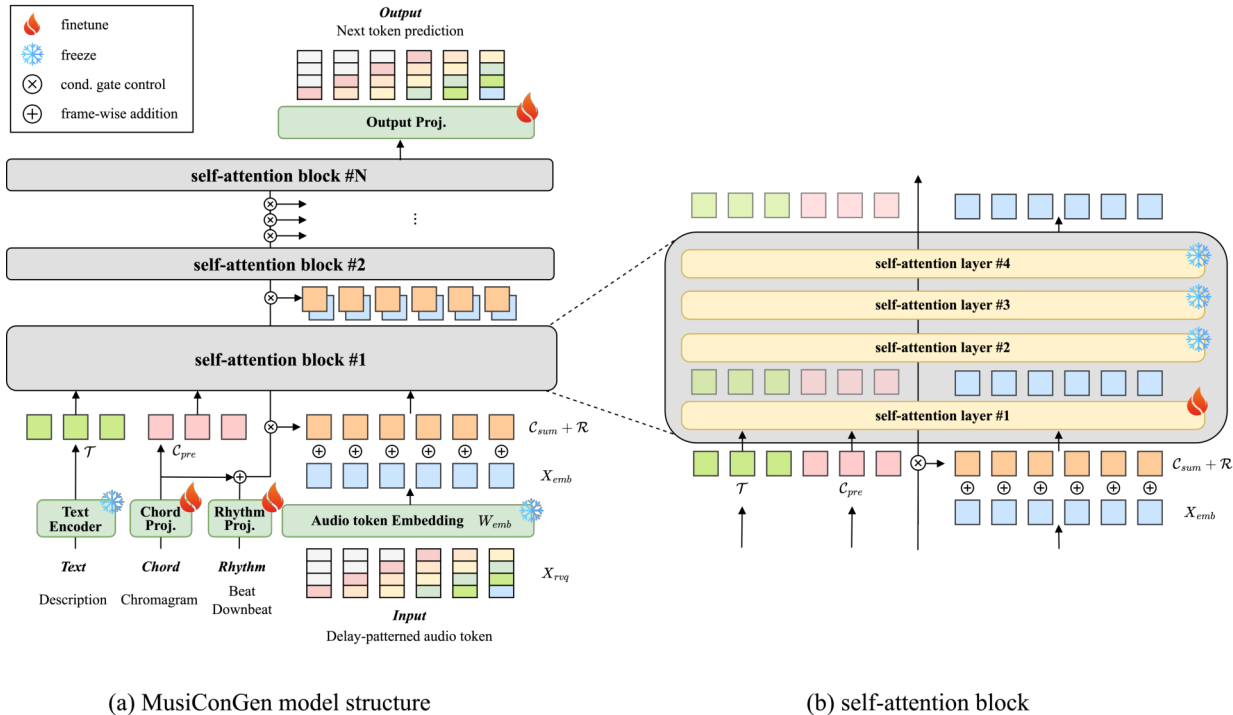


Figure 1. The model structure of MusiConGen and the self-attention block. a) MusiConGen takes text \mathcal{T} , downsampled chord C_{pre} as prepended condition and frame-wise chord C_{sum} and rhythm \mathcal{R} as additive condition. The addition operation of frame-wise conditions to each self-attention block is regulated by the condition gate control (\otimes). b) Each self-attention block consists of four layers. In our proposed model, only the first layer is finetuned, which is also called jump finetuning.

3.1 Representing Temporal & Symbolic Conditions

Chords. For chord condition, we employ two methods. The first **prepend** method is similar to the melody control method of MusicGen, denoted as $C_{pre} \in \mathbb{R}^{T_{f_M} \times 12}$ where C_{pre} maintains the same resolution (i.e. frame rate f_M and sequence length) as MusicGen’s melody condition \mathcal{M} . This allows us to utilize the pretrained melody projection weights from MusicGen as initial weights. Furthermore, we have noted that chord transitions can lead to asynchronization issues. To address this, we introduce a second **frame-wise** chord condition, $C_{sum} \in \mathbb{R}^{T_{f_s} \times 12 \times 1}$, which matches the resolution of the audio codec tokens, thus providing a solution for the synchronization problem.

Rhythm. To control rhythm, we derive conditions from both the beat and the downbeat. The beat represents the consistent pulse within a piece of music, and the downbeat signifies the first and most emphasized beat of each measure, forming the piece’s rhythmic backbone. We encode beat and downbeat information into one-hot embedding each at a frame rate of f_s . For the beat embedding, a soft kernel is applied to allow for a tolerance of 70ms. Subsequently, the beat and downbeat arrays are summed to yield the **frame-wise** rhythm condition $\mathcal{R} \in \mathbb{R}^{T_{f_s} \times 1}$.

3.2 Finetuning Mechanisms

The finetuning mechanism we employ consists of two parts: 1) jump finetuning, and 2) an adaptive in-attention mechanism. As illustrated in Figure 1, our proposed model activates condition gates at the “block” level, treating four

consecutive self-attention layers as a block.

Jump finetuning is designed to specifically target the *first* self-attention layer *within each block* for finetuning, while freezing the remaining three self-attention layers of the same block, as shown in Figure 1 (b). Doing so reduces the number of parameters of finetuning while maintaining the flexibility to learn to respond to the new conditions by refining the first self-attention layer per block.

The adaptive in-attention mechanism is designed to improve control over chords and rhythm. It is an adaptation of the in-attention technique of MuseMorphose [18], whose main idea is to augment every intermediate output of the self-attention layers with copies of the condition. Unlike the original implementation that augment all the self-attention layers, we selectively apply it to the first three-quarters of self-attention blocks (e.g., for a model with 12 blocks, in-attention is applied to first 9 blocks) to relax the control in the last few blocks for better balancing on rhythm and chords. This leads to better result empirically, as will be shown in Section 5.2 and Table 3.

4. EXPERIMENTAL SETUP

4.1 Datasets

We finetuned the model using a dataset of ~ 250 hours backing track music sourced from YouTube, comprising 5K songs across five genres: Rock, Funk, Jazz, Blues, and Metal, with 1K songs per genre. After preprocessing (see Section 4.2), the training data contained 80,871 clips.

For evaluation, we used the rhythm and chords from two public-domain datasets—MUSDB18 [20] and RWC-pop-100 [21]. For MUSDB18, the rhythm and chords are *extracted* from the audio signals, so this dataset reflects the case where the condition signals are from a *reference audio*. There are 150 songs with four isolated stems: vocal, bass, drum, and others. For each song, we dropped the vocals and divided the mix of the remaining tracks into 30-second clips, resulting in a total of 1,089 clips.

The RWC comprises 100 Japanese pop songs with human annotated chord progressions and BPM labels. We simply use the *human labels* as the conditions here, reflecting the case where the condition signals are user provided *in a text-like format*. We similarly divided each song into 30-second clips, leading to 755 clips in total.

4.2 Dataset Pre-processing Details

The training and evaluation datasets consist of full-song data, with durations ranging from 2 to 5 minutes per song. Below are the preprocessing details for each type of input:

Audios: All audio data have vocals removed. For the training and RWC dataset, we employed the source separation model Demucs [24, 25] to eliminate the vocal stem. In the MUSDB18 dataset, which already features isolated stems, we combined the bass, drum, and others stems to form the dataset. Each song was segmented into 30-second clips, ensuring each clip starts at a downbeat.

Descriptions: For the training set, the text prompts were simply extracted from the titles of the corresponding YouTube videos. For the two evaluation datasets, we tasked ChatGPT [26] to generate 16 distinct text prompts, covering the five genres included by the training set. Here is an example—“A smooth acid Jazz track with a laid-back groove, silky electric piano, and a cool bass, providing a modern take on Jazz. Instruments: electric piano, bass, drums.” At inference time, we randomly selected one of the 16 text prompts in a uniform distribution.

Chords: The RWC dataset comes with ground truth labeled chords. For both the training set and MUSDB18, we used the BTC model [27] as the chord extraction model to predict symbolic chords with time tags for each clip. The detailed chord quality extends to the seventh note. We then translated the extracted chord symbols with time tags into a 12-pitch chromagram in the order of C, C#, ..., B. The chromagram’s frame rate for the frame-wise condition \mathcal{C}_{sum} is f_s , and for the prepend condition \mathcal{C}_{pre} it is f_M .

Rhythm: Except for RWC, beat and downbeat were extracted using the RNN+HMM model [28] from the Madmom library [29]. The timing format for beats and downbeats was transformed into a one-hot representation matching the audio token frame rate f_s . A soft kernel was applied to the one-hot beat array to create a softened beat array. The rhythm representation \mathcal{R} was the frame-wise summation of the softened beat array and downbeat array.

4.3 Training Configuration

The proposed rhythm and chord-conditioned Transformer was built upon the architecture of the medium-sized (1.5B)

MusicGen-melody, featuring $L = 48$ self-attention layers with dimension $D = 1,536$ and 24 multi-head attention units. The condition dropout rate is 0.5 and guidance scale is set to be $\gamma = 3$ for classifier-free guidance. We finetuned only a quarter of the full model, which corresponds to 352 million parameters, while keeping both the audio token embedding lookup table and the FLAN-T5 text encoder frozen. The training involved 100K finetuning steps, carried out over approximately 2 days on 4 RTX-3090 GPUs, with a batch size of 2 per GPU for each experiment.

4.4 Objective Evaluation Metrics

We employed metrics to evaluate controllability of chords and rhythm, textual adherence and audio fidelity. For the first two metrics, we used the rhythm and chord conditions from a clip in a evaluation dataset to generate music (along with a text prompt generated by ChatGPT; see Section 4.2), applied the Madmom and BTC models on the generated audio to estimate beats and chords, and evaluated how they reflect the given conditions. See Figure 2 for examples.

Chord. We used the *mir_eval* [30] package to measure 3 different degrees of frame-wise chord correctness: **majmin**, **triads** and **tetrads**. The majmin function compares chords in major-minor rule ignoring chord qualities outside major/minor/no-chord. The triads function compares chords along triad (root & quality to #5), while the tetrads compares chords along tetrad (root & full quality).

Rhythm F1 measurement follows the standard methodology for beat evaluation. We measured the beat accuracy also via *mir_eval*, assessing the alignment between the beat timestamps of the generated music and the reference rhythm music data, with a tolerance window of 70ms.

CLAP [31, 32] score examines the textual adherence by the cosine similarity between the embedding of the text prompt and that of the generated audio in a text-audio joint embedding space learned by contrastive learning. Here, we used the LAION CLAP model trained for music [33], *music_audioset_epoch_15_esc_90.14.pt*.

FAD is the Fréchet distance between the embeddings distribution from a set of reference audios and that from the generated audios [34, 35]. The metric represent how realistic the generated audios are compared to the given reference audios. The audio encoder of FAD we used is VGGish [36] model which trained on an audio classification task. The reference set of audios was from MUSDB18 or RWC depending on the evaluation set.

4.5 Subjective Evaluation Metrics

We also did a listening test to evaluate the followings aspects: text relevance, rhythm consistency, and chord relevance. Text relevance concerns how the generated audio clips reflect the given text prompts. Rhythm consistency is about how steady the beats is within an audio clip. (We found that, unlike the case of objective evaluations, minor out-of-sync beats at the beginning of a clip were deemed acceptable here perceptually.) Chord relevance concerns how a generated clip follows the given chord progressions.

Model	Evaluation dataset	Rhythm	Chord			FAD	CLAP
		F-measure(%)	majmin(%)	triads(%)	tetrads(%)		
proposed ($\mathcal{C}_{pre} + \mathcal{C}_{sum} + \mathcal{R}$)	MUSDB18	69.76	67.03	66.19	56.91	1.29	0.34
	RWC	79.40	73.03	68.42	54.12	0.96	0.34
chords only ($\mathcal{C}_{pre} + \mathcal{C}_{sum}$)	MUSDB18	39.47	73.25	72.29	60.89	1.91	0.34
	RWC	49.85	73.30	68.50	50.66	2.18	0.34
rhythm only (\mathcal{R})	MUSDB18	61.37	5.84	5.76	3.84	1.95	0.32
	RWC	58.39	5.40	5.08	2.90	2.67	0.32
no frame-wise chords ($\mathcal{C}_{pre} + \mathcal{R}$)	MUSDB18	61.68	57.39	56.65	47.17	1.44	0.35
	RWC	69.30	60.95	57.19	44.21	1.29	0.35
baseline (no finetuning; \mathcal{M} for \mathcal{C}_{pre})	MUSDB18	26.14	53.13	52.31	44.83	2.01	0.34
	RWC	30.67	51.90	48.54	35.81	2.30	0.35

Table 2. Objective evaluation results for models with different conditions on two different test sets MUSDB18 and RWC. With the proposed condition representation, we can achieve better performance both in rhythm and chord controls.

Model	Evaluation dataset	Rhythm	Chord			FAD	CLAP
		F-measure(%)	majmin(%)	triads(%)	tetrads(%)		
proposed (jump+adaptive in-attn)	MUSDB18	69.76	67.03	66.19	56.91	1.29	0.34
	RWC	79.40	73.03	68.42	54.12	0.96	0.34
ablation 1 (jump finetuning only)	MUSDB18	42.28	71.06	70.21	61.58	1.39	0.36
	RWC	53.14	76.04	71.33	57.52	1.27	0.36
ablation 2 (jump+full in-attn)	MUSDB18	67.23	66.47	65.60	56.37	1.59	0.35
	RWC	71.13	64.82	60.77	48.07	1.47	0.35
finetuned baseline (jump only; no \mathcal{C}_{sum} no \mathcal{R})	MUSDB18	40.15	55.65	54.88	45.52	1.94	0.36
	RWC	49.25	56.49	52.66	38.07	2.24	0.36

Table 3. Objective evaluation results for models trained with different finetuning mechanisms. We see that the proposed jump finetuning with adaptive (partial) in-attention achieves better result on rhythm and chord controls.

5. EXPERIMENTAL RESULTS

5.1 Objective Evaluation: Temporal Conditions

We assessed the audio generated under various condition combinations applied to the training model, including the proposed method and its **ablations** with either chord- or rhythm-only as the temporal condition, or using both but without the frame-wise chord condition. The finetuning configurations and mechanisms for these models were the same. Moreover, we considered the **baseline** as follows. The pretrained MusicGen-melody model originally processes text and melody conditions \mathcal{T} , \mathcal{M} . We simply used the prepend chord condition \mathcal{C}_{pre} as input to the linear projection layers originally pretrained to take the melody condition, without finetuning the entire model at all. In addition, we appended to the end of the text prompt BPM information (e.g., “at BPM 90”) as the rhythm condition.

Result shown in Table 2 leads to many findings. Firstly, a comparison between the result of the proposed model (first row) and the baseline (last row) demonstrates nicely the effectiveness of the proposed design. The proposed model leads to much higher scores in almost all the met-

rics. Moreover, it performs similarly well for the two evaluation datasets, suggesting that MusiConGen can deal with both conditions extracted from a reference audio signals or provided by creators in a symbolic text-like format.

Secondly, although the baseline model does not perform well, it still exhibits some level of chord control, showing the knowledge of melody can be transferred to chords.

Finally, from the ablations (middle three rows), chord-only and rhythm-only did not work well for rhythm and chord control respectively, which is expected. Compared to the proposed model, excluding per-frame chord condition degrades both chord and rhythm controllability, showing that chord and rhythm are interrelated.

5.2 Objective Evaluation: Finetuning Mechanisms

Besides the proposed finetuning method, we evaluated the following alternatives. **Finetuned baseline** is a baseline model that was finetuned using the prepended chords (\mathcal{C}_{pre}) instead of melody \mathcal{M} the frame-level conditions, employing the jump finetuning mechanism but no in-attention. **Jump finetuning without in-attention** (ablation 1) and **jump finetuning with full in-attention** (abal-

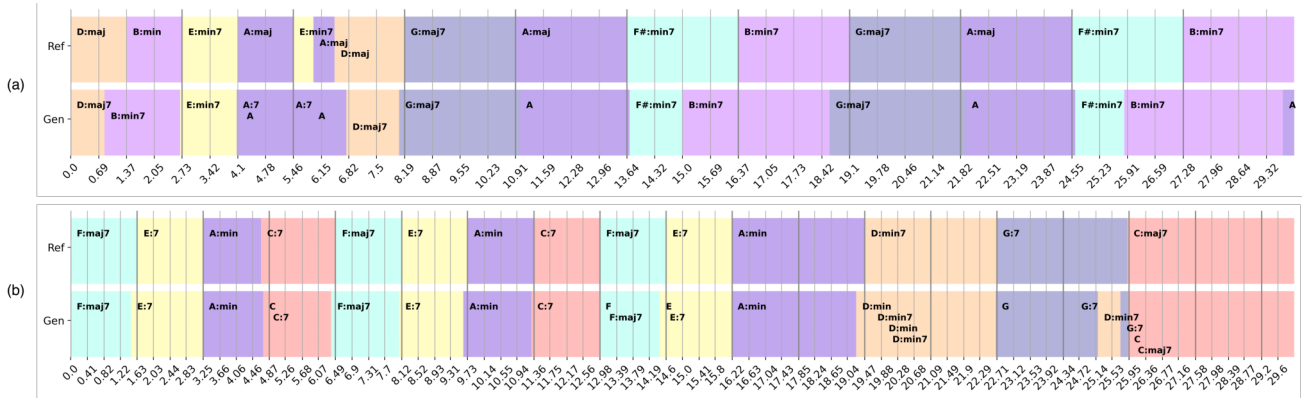


Figure 2. Comparison on chord progression and beats of ground truth and generated samples, using the conditions from RWC. For each example (a) or (b), the top row is ground truth chords and the bottom row is extracted chords from generated samples. The thick and light gray lines indicate the times of the downbeat and the beat, respectively.

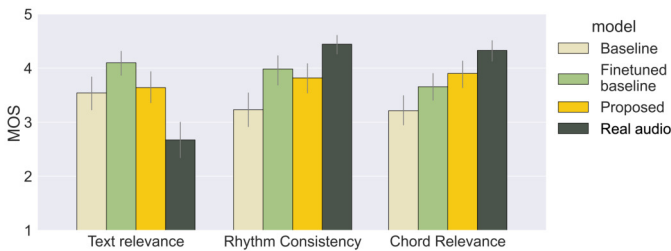


Figure 3. Subjective evaluation of condition controls—5-scale mean opinion score with 95% confidence interval.

ation 2) are ablations which use full conditions (prepended chord C_{pre} , frame-wise chord C_{sum} , and rhythm \mathcal{R}), but we either dropped in-attention entirely, or employed in-attention to every self-attention block, instead of only the first three-quarter blocks as done by the proposed method.

The result is tabulated in Table 3. Among the four methods, the proposed method leads to the best rhythm control and very competitive chord control. Comparing the results of the proposed method and the two ablations reveals a trade-off in rhythm and chord control when we go from no in-attention, adaptive (partial) in-attention, to full in-attention. The proposed method strikes an effective balance between rhythm and chord controls.

Comparing the last row of Table 2 and that of Table 3 shows that the finetuned baseline outperforms the baseline (with no finetuning at all) mainly in the rhythm control. This is notable as the finetuned baseline is actually trained with only the prepend chord condition C_{pre} , not using the rhythm condition \mathcal{R} , suggesting again the interrelation of chord and rhythm. Moreover, although the finetuned baseline is better than the baseline, it is still much inferior to the proposed method in both chord and rhythm controls.

5.3 Subjective Evaluation

We evaluated three models in the listening test: the baseline, the finetuned baseline, and the proposed model. Each model generates a music clip using the ChatGPT-generated text prompts, along with the BPM and chords

from the RWC dataset, namely considering text-like symbolic rhythm and chord conditions. Besides the audios generated by the three models, we also included real audios from the RWC dataset as the **real audio**. We note that the real audios would have perfect rhythm and chord controllability (for they are where the conditions are from), but the textual adherence would be bad because RWC songs are J-Pop rather than any of the five genres (i.e., Rock, Funk, Jazz, Blues, and Metal) described by the text prompts.

We had 23 participants in the user study, 85% of whom have over three years of musical training. Each time, we displayed the given text, rhythm and chord conditions, and asked a participant to rate the generated audio and the real audio (anonymized and in random order) on a five-point Likert scale. The result is shown in Figure 3.

Several findings emerged. Firstly, the proposed model demonstrated superior chord control compared to the other two models, although it still fell short of matching the real audio. Secondly, the proposed model has no significant advantage on rhythm consistency against the finetuned baseline. As suggested by the examples on our demo page, we found that being on the precise beat onset does not significantly impact rhythm perception. Thirdly, our model had lower text relevance than the finetuned baseline, suggesting that our model may have traded text control for increased temporal control of rhythm and chords.

6. CONCLUSION AND FUTURE WORK

This paper has presented conditioning mechanisms and finetuning techniques to adapt MusicGen for better rhythm and chord control. Our evaluation on backing track generation shows that the model can take condition signals from either a reference audio or a symbolic input. For future work, our user study shows room to further improve the rhythm and chord controllability while keeping the text relevance. This might be done by scaling up the model size, better language model, or audio codecs. It is also interesting to incorporate additional conditions, such as symbolic melody, instrumentation, vocal audio, and video clips.

7. ACKNOWLEDGEMENTS

We are grateful to the discussions and feedbacks from the research team of Positive Grid, a leading global guitar amp and effect modeling company, during the initial phase of the project. We also thank the comments from the anonymous reviewers and meta-reviewer. The work is also partially supported by grants from the National Science and Technology Council (NSTC 112-2222-E-002-005-MY2), (NSTC 113-2628-E-002-029), and from the Ministry of Education of Taiwan (NTU-112V1904-5).

8. REFERENCES

- [1] A. Agostinelli, T. I. Denk, Z. Borsos, J. H. Engel, M. Verzetti, A. Caillon, Q. Huang, A. Jansen, A. Roberts, M. Tagliasacchi, M. Sharifi, N. Zeghidour, and C. H. Frank, “MusicLM: Generating music from text.” *arXiv preprint arXiv:2301.11325*, 2023.
- [2] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Défossez, “Simple and controllable music generation,” in *Proc. NeurIPS*, 2023.
- [3] Q. Huang, D. S. Park, T. Wang, T. I. Denk, A. Ly, N. Chen, Z. Zhang, Z. Zhang, J. Yu, C. Frank, J. Engel, Q. V. Le, W. Chan, Z. Chen, and W. Han, “Noise2Music: Text-conditioned music generation with diffusion models,” *arXiv preprint arXiv:2302.03917*, 2023.
- [4] H. Liu, Q. Tian, Y. Yuan, X. Liu, X. Mei, Q. Kong, Y. Wang, W. Wang, Y. Wang, and M. D. Plumbley, “AudioLDM 2: Learning holistic audio generation with self-supervised pretraining,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 32, pp. 2871–2883, 2024.
- [5] P. Li, B. Chen, Y. Yao, Y. Wang, A. Wang, and A. Wang, “JEN-1: Text-guided universal music generation with omnidirectional diffusion models,” *arXiv preprint arXiv:2308.04729*, 2024.
- [6] L. Lin, G. Xia, J. Jiang, and Y. Zhang, “Content-based controls for music large language modeling,” *arXiv preprint arXiv:2310.17162*, 2023.
- [7] S.-L. Wu, C. Donahue, S. Watanabe, and N. J. Bryan, “Music ControlNet: Multiple time-varying controls for music generation,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 32, pp. 2692–2703, 2024.
- [8] K. Chen, Y. Wu, H. Liu, M. Nezhurina, T. Berg-Kirkpatrick, and S. Dubnov, “MusicLDM: Enhancing novelty in text-to-music generation using beat-synchronous mixup strategies,” in *Proc. ICASSP*, 2024.
- [9] J. Melechovsky, Z. Guo, D. Ghosal, N. Majumder, D. Herremans, and S. Poria, “Mustango: Toward controllable text-to-music generation,” in *Proc. NAACL*, 2024.
- [10] Y. Zhang, Y. Ikemiya, G. Xia, N. Murata, M. Martínez, W.-H. Liao, Y. Mitsufuji, and S. Dixon, “MusicMagus: Zero-shot text-to-music editing via diffusion models,” in *Proc. IJCAI*, 2024.
- [11] F.-D. Tsai, S.-L. Wu, H. Kim, B.-Y. Chen, H.-C. Cheng, and Y.-H. Yang, “Audio Prompt Adapter: Unleashing music editing abilities for text-to-music with lightweight finetuning,” in *Proc. ISMIR*, 2024.
- [12] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, and I. Sutskever, “Jukebox: A generative model for music,” *arXiv preprint arXiv:2005.00341*, 2020.
- [13] N. Zeghidour, A. Luebs, A. Omran, J. Skoglund, and M. Tagliasacchi, “SoundStream: An end-to-end neural audio codec,” *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, vol. 30, p. 495–507, 2021.
- [14] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, “High fidelity neural audio compression,” *arXiv preprint arXiv:2210.13438*, 2022.
- [15] R. Kumar, P. Seetharaman, A. Luebs, I. Kumar, and K. Kumar, “High-fidelity audio compression with improved RVQGAN,” in *Proc. NeurIPS*, 2023.
- [16] R. Zhang, J. Han, C. Liu, P. Gao, A. Zhou, X. Hu, S. Yan, P. Lu, H. Li, and Y. Qiao, “LLaMA-Adapter: Efficient fine-tuning of language models with zero-init attention,” *arXiv preprint arXiv:2303.16199*, 2023.
- [17] L. Zhang, A. Rao, and M. Agrawala, “Adding conditional control to text-to-image diffusion models,” in *Proc. ICCV*, 2023.
- [18] S.-L. Wu and Y.-H. Yang, “MuseMorphose: Full-song and fine-grained piano music style transfer with one Transformer VAE,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 1953–1967, 2023.
- [19] S. Mangrulkar, S. Gugger, L. Debut, Y. Belkada, S. Paul, and B. Bossan, “PEFT: State-of-the-art parameter-efficient fine-tuning methods,” 2022. [Online]. Available: <https://github.com/huggingface/peft>
- [20] Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimilakis, and R. Bittner, “The MUSDB18 corpus for music separation,” 2017. [Online]. Available: <https://sigsep.github.io/datasets/musdb.html>
- [21] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, “RWC Music Database: Popular, classical, and jazz music databases,” in *Proc. ISMIR*, 2002. [Online]. Available: <https://staff.aist.go.jp/m.goto/RWC-MDB/>
- [22] J. Ho and T. Salimans, “Classifier-free diffusion guidance,” in *Proc. NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.

- [23] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, Y. Li, X. Wang, M. Dehghani, S. Brahma *et al.*, “Scaling instruction-finetuned language models,” *arXiv preprint arXiv:2210.11416*, 2022.
- [24] S. Rouard, F. Massa, and A. Défossez, “Hybrid Transformers for music source separation,” in *Proc. ICASSP*, 2023. [Online]. Available: <https://github.com/facebookresearch/demucs>
- [25] A. Défossez, “Hybrid spectrogram and waveform source separation,” in *Proc. ISMIR 2021 Workshop on Music Source Separation*, 2021.
- [26] J. Schulman, B. Zoph, C. Kim, J. Hilton, J. Menick, J. Weng *et al.*, “Introducing ChatGPT,” 2022.
- [27] J. Park, K. Choi, S. Jeon, D. Kim, and J. Park, “A bi-directional Transformer for musical chord recognition,” in *Proc. ISMIR*, 2019. [Online]. Available: <https://github.com/jayg996/BTC-ISMIR19>
- [28] S. Böck, F. Krebs, and G. Widmer, “Joint beat and downbeat tracking with recurrent neural networks,” in *Proc. ISMIR*, 2016, pp. 255–261.
- [29] S. Böck, F. Korzeniowski, J. Schlüter, F. Krebs, and G. Widmer, “madmom: a new Python audio and music signal processing library,” in *Proc. ACM Multimedia*, 2016, pp. 1174–1178. [Online]. Available: <https://github.com/CPJKU/madmom>
- [30] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, and D. P. W. Ellis, “Mir_eval: A transparent implementation of common MIR metrics,” in *Proc. ISMIR*, 2014, pp. 367–372. [Online]. Available: https://github.com/craffel/mir_eval
- [31] Y. Wu, K. Chen, T. Zhang, Y. Hui, T. Berg-Kirkpatrick, and S. Dubnov, “Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation,” in *Proc. ICASSP*, 2023.
- [32] K. Chen, X. Du, B. Zhu, Z. Ma, T. Berg-Kirkpatrick, and S. Dubnov, “HTS-AT: A hierarchical token-semantic audio Transformer for sound classification and detection,” in *Proc. ICASSP*, 2022.
- [33] “LAION CLAP.” [Online]. Available: <https://github.com/LAION-AI/CLAP>
- [34] K. Kilgour, M. Zuluaga, D. Roblek, and M. Sharifi, “Fréchet Audio Distance: A metric for evaluating music enhancement algorithms,” *arXiv preprint arXiv:1812.08466*, 2018.
- [35] A. Gui, H. Gamper, S. Braun, and D. Emmanouilidou, “Adapting Fréchet audio distance for generative music evaluation,” in *Proc. ICASSP*, 2024, pp. 1331–1335.
- [36] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold *et al.*, “CNN architectures for large-scale audio classification,” in *Proc. ICASSP*, 2017.

END-TO-END PIANO PERFORMANCE-MIDI TO SCORE CONVERSION WITH TRANSFORMERS

Tim Beyer

Technical University of Munich
tim.beyer@tum.de

Angela Dai

Technical University of Munich
angela.dai@tum.de

ABSTRACT

The automated creation of accurate musical notation from an expressive human performance is a fundamental task in computational musicology. To this end, we present an end-to-end deep learning approach that constructs detailed musical scores directly from real-world piano performance-MIDI files. We introduce a modern transformer-based architecture with a novel tokenized representation for symbolic music data. Framing the task as sequence-to-sequence translation rather than note-wise classification reduces alignment requirements and annotation costs, while allowing the prediction of more concise and accurate notation. To serialize symbolic music data, we design a custom tokenization stage based on compound tokens that carefully quantizes continuous values. This technique preserves more score information while reducing sequence lengths by $3.5\times$ compared to prior approaches. Using the transformer backbone, our method demonstrates better understanding of note values, rhythmic structure, and details such as staff assignment. When evaluated end-to-end using transcription metrics such as MUSTER, we achieve significant improvements over previous deep learning approaches and complex HMM-based state-of-the-art pipelines. Our method is also the first to directly predict notational details like trill marks or stem direction from performance data. Code and models are available on [GitHub](#).

1. INTRODUCTION

Creating structured musical scores from human performance recordings is a challenging task with a significant number of downstream applications in areas such as alignment [1], score-following, education, and archiving.

Human performances are typically represented as performance-MIDI (P-MIDI) files, as they can be easily recorded from MIDI instruments or generated from audio by automated transcription systems [2, 3]. In contrast, high-quality scores in standard sheet music formats such as MusicXML [4] are much less commonly available and generally require creation by human experts.

Performance-MIDI-to-Score conversion (PM2S) is complex, encompassing several lower-level tasks like note

value prediction, tempo regression, rhythm quantization, voice assignment, and typesetting details, including ornaments and note stems. As a result, PM2S and its sub-tasks have remained an active research topic and popular application of computational methods for over 30 years [5].

While early approaches relied on classical modeling and hand-crafted processing [6], research gradually shifted towards statistical methods based on Hidden Markov Models (HMMs) augmented with heuristics.

Cogliati et al. [7] run an HMM-based meter estimation [8] with beat-snapping heuristics to quantize note timings, before outputting LilyPond [9] notation. HMM variants were also used for estimating staff placement [10] and rhythm quantization [11, 12], where the outputs of multiple models are merged into a final prediction with improved accuracy. To complement onset timing quantization, a method for note value recognition based on Markov random fields was introduced by Nakamura et al. [13]. Building upon these advances, Shibata et al. [14] combined prior systems with hand-crafted non-local statistics to improve estimates of global attributes such as piece tempo and time signatures. While these approaches yield state-of-the-art performance, they are composed of a complex web of interdependent components, rely on human-designed priors, and are not trained end-to-end. Thus, more recent work has attempted to tackle PM2S using deep learning.

So far, the scarcity of high-quality labeled data has limited its use to scenarios with cheaper labels, leading to a focus on synthetic data [15, 16], sub-tasks like pitch-spelling [17], note value quantization and voicing [18], or beat tracking [19, 20]. Beat tracking, in particular, has seen significant progress by combining CRNNs with beat in-filling via dynamic programming [19]. The CRNN also predicts other score attributes such as key and time signatures. Unfortunately, beat-tracking makes overly restrictive assumptions about the regularity of the underlying performance data and struggles with real-world human recordings.

Another challenge is the representation of symbolic music data for machine learning models. Monophonic sequences are often captured as Lilypond [9, 15], ABC [21], Humdrum-derived [16, 22], or custom CTC-friendly [16] character sequences. For polyphonic data, piano rolls [2, 23], MIDI-derived tokens [24–26], and custom MusicXML tokens [27] are the most common representations. To create more compact encodings, Zeng et al. [28] and Dong et al. [29] use compound tokens and represent MIDI attributes in separate streams, shortening sequence lengths.



Our proposed approach continues the progression of PM2S systems towards end-to-end learned approaches and overcomes several limitations of prior systems based on deep learning, making the following key contributions:

- We cast PM2S as an end-to-end sequence-to-sequence translation task, developing a transformer to enable accurate prediction of global attributes (e.g., meter) that require understanding of long-term dependencies.
- Relaxed annotation requirements compared to prior deep learning methods, using only beat-level alignment for training. We can additionally leverage unmatched MusicXML data without corresponding P-MIDI.
- We introduce a compact and extensible tokenization scheme for P-MIDI and MusicXML data, allowing the backbone model to directly translate tokenized P-MIDI into MusicXML tokens and enabling the generation of detailed score features such as ornaments.
- We demonstrate superior performance on quantitative error metrics like MUSTER, with our approach surpassing prior deep learning models and the highly optimized, complex state-of-the-art.

2. METHODOLOGY

2.1 Task definition

Our end-to-end PM2S system directly converts an unstructured P-MIDI file into a highly readable MusicXML score. P-MIDI files only contain information about note timing (onsets, offsets), pitch, and velocity. The input to a PM2S system is thus defined by the following sequence:

$$\mathbf{X} = \{(p_i, o_i, d_i, v_i)\}_{i=1}^{N_{\text{perf}}}, \quad (1)$$

with MIDI pitch p_i , onset o_i and duration d_i in seconds, and velocity v_i for each of the N_{perf} performance notes.

In contrast to existing methods, which often cast PM2S as a note-wise classification task [18, 19], we do not assume a one-to-one correspondence between notes in the performance and the score. This is crucial in scenarios with trills or misplayed notes, where one-to-one matchings are impossible. Consequently, we predict a new output note sequence from scratch that includes a full set of MusicXML attributes for each note in the score:

$$\mathbf{Y}_{\mathbf{q}} = \{(p_j, mo_j, md_j, ml_j)\}_{j=1}^{N_{\text{score}}} \quad (2)$$

$$\mathbf{Y}_{\mathbf{v}} = \{(h_j, vo_j)\}_{j=1}^{N_{\text{score}}} \quad (3)$$

$$\mathbf{Y}_{\mathbf{o}} = \{(t_j, s_j, sd_j, g_j, a_j)\}_{j=1}^{N_{\text{score}}} \quad (4)$$

$$\mathbf{Y} = (\mathbf{Y}_{\mathbf{q}}, \mathbf{Y}_{\mathbf{v}}, \mathbf{Y}_{\mathbf{o}}), \quad (5)$$

where $\mathbf{Y}_{\mathbf{q}}$ comprises attributes related to pitch p_j and quantized timings for the musical onset time mo_j , musical duration md_j , and measure length ml_j . $\mathbf{Y}_{\mathbf{v}}$ collects vertical positioning information, such as staff placement/hand h_j , and MusicXML voice number vo_j . Finally, $\mathbf{Y}_{\mathbf{o}}$ covers performance annotations, ornamentation, and typesetting details like trill t_j , staccato s_j , stem direction sd_j , grace note g_j , and accidentals a_j . Predicting these additional attributes enables creating more concise and accurate notation. \mathbf{X} and \mathbf{Y} are sorted by ascending onset/offset, pitch, and duration, yielding a unique serialized representation even for complex polyphony.

2.2 Tokenization scheme

To efficiently represent input \mathbf{X} and output \mathbf{Y} , we introduce a systematic tokenization for P-MIDI files and MusicXML scores. The key objective of a tokenization algorithm is to retain as much information from the original sequence as possible within a compact sequence length and vocabulary size. Thus, we adopt a parallel token stream paradigm [28]; a separate token stream is constructed for each of the four input attributes given in Eq. (1) and the eleven output attributes in Eq. (5). As a result, each note occupies only one timeslot. The final vocabulary sizes and parameter ranges are shown in Table 1.

For P-MIDI, we adopt a strategy similar to [19] and use 128 pitch tokens, 8 quantized velocity tokens, and quantize delta onsets and durations into 200 buckets. To achieve high resolution for small values while covering times up to 8 seconds without clipping, we apply a *log*-transform before bucketing onsets and durations, implementing a continuous version of multi-resolution quantization [28].

We pay particular attention to the MusicXML tokenization. While binary and categorical attributes, such as stem direction and staff assignment, are easily tokenized, continuous values like onsets and durations demand more care. The encoding of musical timing and positioning significantly impacts score quality. To find a good trade-off between vocabulary size and the ability to correctly represent common note durations and onsets, we conducted a search over bucket sizes. Consequently, we opt to quantize musical time into $\frac{1}{24}$ th fractions, diverging from previous approaches, which rely on powers of 2 or smaller denominators [18, 28, 30]. Our parameterization accurately represents 98.6% of notes in the ASAP dataset with 97 tokens, compared to just 85.4% using powers of 2 up to 256.

Encoding absolute musical onsets into tokens poses another challenge. Direct quantization of absolute positions is infeasible due to the large range of positions required, while delta encoding similar to MIDI quickly leads to drift issues and misalignment with measure boundaries and the

Parameter	N_{vocab}	Range/Values
<i>Input Parameters</i>		
Pitch (p_i)	128	[0, 127]
Onset (o_i)	200	[0, 8]
Duration (d_i)	200	[0, 8]
Velocity (v_i)	8	[0, 127]
<i>Output Parameters</i>		
Pitch (p_j)	128	[0, 127]
Musical Onset (mo_j)	145	[0, 6]
Musical Duration (md_j)	97	[0, 4]
Measure Length (ml_j)	146	[0, 6] \cup {false}
Hand/Staff (h_j), Trill (t_j), Grace (g_j), Staccato (s_j)	2 each	boolean
Voice (vo_j)	8	[1, 8]
Stem (sd_j)	3	{up, down, none}
Accidental (a_j)	6	{bb, b, ♮, #, x, none}

Table 1. Parameter specifications for input/output representations. The rightmost column details the range or set of representable values for each attribute. Continuous values outside the range are clipped before tokenization.

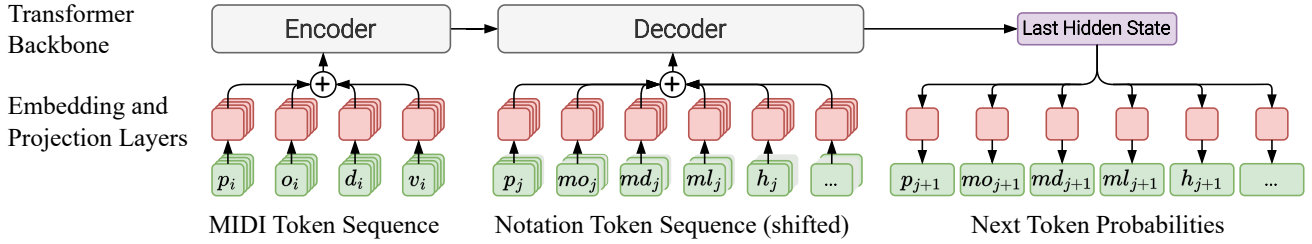


Figure 1. Model architecture overview. We use a standard Roformer encoder-decoder model [31] with custom token embedding and projection layers. Each token stream is embedded separately, then a constant-size shared embedding is created via summation. The backbone model architecture remains unchanged compared to models applied to NLP or other sequence-to-sequence learning tasks. In this illustration, depth symbolizes the time direction.

musical grid. Thus, we adopt a hybrid approach representing absolute positions using two tokens; mo_j encodes the note’s position relative to the start of the current measure, and ml_j stores the preceding measure’s length for the first note in each measure or is set to `false` otherwise. Combined, mo_j and ml_j enable the reconstruction of absolute musical times, including correct bar lines and most time signatures. During score creation, bar lines are used to split and tie notes crossing measure boundaries, recovering most ties, and rests are added to fill any gaps in voices.

Since input and output streams are not necessarily the same length, we also insert space tokens (sp_j) where required for alignment (see also Section 3.1.2). The space tokens differ from typical end-of-sequence padding or masking tokens in Transformers, as they are predicted during inference and attention to these positions is not masked.

2.3 Model architecture

By adopting a unified autoregressive Transformer encoder-decoder model [32] to directly translate tokenized P-MIDI into MusicXML tokens (as depicted in Figure 1), we diverge from existing deep learning models for PM2S, which used subtask-specific LSTMs [33] or CRNNs. Our choice is driven by the transformer’s ability to scale to large datasets and to handle long-range dependencies, which are crucial for predicting piece-wide attributes like meter.

To interface with parallel token streams, our model introduces custom embedding and projection modules. First, each attribute-specific token stream is mapped into a constant-size 512-dimensional embedding space. The results are then summed and normalized using LayerNorm [34] to form a constant-size shared embedding, independent of the token stream count.

The backbone model itself follows the original architecture described by Vaswani et al. [32] and consists of symmetrically arranged encoder and decoder stacks. Each stack comprises four layers, eight attention heads, and a model dimension of 512. To optimize performance, we adopt rotary positional encodings [31], pre-norm [35], and SwiGLU activations [36] with an inner dimension of 3072 for the position-wise feed-forward network. At the end of the decoder, a set of linear layers projects the final hidden state into one output logit distribution per token stream.

2.4 Training & inference details

To optimize our model parameters, we break down the loss computation into two stages and first compute per-timestep

losses \mathcal{L}_j , before summing along the sequence position. At each timestep j , our model performs 12 separate classification tasks, one for every token stream in \mathbf{Y} and one for the space token sp_j .

We compute the cross entropy (CE) loss for each output token stream y and the space token stream. For timesteps with spacing token sp_j , the loss is calculated only for the space token stream since the labels for all other tokens are undefined. The full loss computation is thus:

$$\mathcal{L}_{y,j} = \text{CE}(\hat{y}_j, y_j) \quad (6)$$

$$\mathcal{L}_j = \begin{cases} \text{CE}(\hat{sp}_j, 1) & \text{for } sp_j = 1, \\ \sum_{y \in \mathbf{Y}} \mathcal{L}_{y,j} + \text{CE}(\hat{sp}_j, 0) & \text{otherwise.} \end{cases} \quad (7)$$

$$\mathcal{L} = \sum_{j=1}^{N_{\text{score}}} \mathcal{L}_j. \quad (8)$$

We train our model for 40,000 steps using the AdamW optimizer [37]. The learning rate follows a cosine learning rate decay schedule with linear warmup over the first 4,000 steps to a maximum learning rate of $3e-4$. Gradients are clipped to a maximum value of 0.5. We use a batch size of 32 and the training sequence length is 512 timesteps.

To parallelize training, transformers are often trained with teacher forcing. However, exposure bias [38] can lead to lower-than-expected performance at inference time, especially in the low-data regime. We find that heavy dropout [39] during training to expose the model to only 25% of the preceding output tokens addresses this problem.

During inference, we employ greedy top-1 decoding as it provides better performance than alternatives. To handle songs with more than 512 notes, we partition the input into chunks of 512 notes each, ensuring a 64-note overlap between consecutive segments. Sufficient overlap eliminates abrupt changes at the segment boundaries and is essential for generating temporally coherent scores.

3. EXPERIMENTS

3.1 Data

3.1.1 Datasets

Unlike prior PM2S systems [14, 18, 19], we do not use the MAPS [40] dataset in our experiments, as performance data contained therein is not representative of real-world P-MIDI¹. Furthermore, its musical scores are only available

¹ The underlying data is score-derived and has been manually adjusted to represent aspects of performance and score at the same time. This leads

in the MIDI format, which lacks representational capacity compared to MusicXML. Thus, MAPS scores do not effectively capture many aspects of musical notation.

To overcome these limitations, we use the ASAP dataset [41] for training and evaluation. It contains 1067 pieces of P-MIDI recorded from expert piano performances and corresponding high-fidelity MusicXML scores. Performance and scores are aligned with beat-level annotations, which are significantly cheaper to obtain than note-level alignments. We also observed that, on average, MusicXML scores contain 2.6% fewer notes than associated P-MIDI. These discrepancies are typically caused by misplayed notes and trills, again highlighting the importance of a flexible approach that is not reliant on one-to-one correspondences and can handle a wide variety of notation features.

After manually inspecting the dataset, we reject 100 instances due to poor alignment or data corruption, leaving 967 performances. We perform only minimal preprocessing, focusing on removing non-sounding notes from the score. This includes merging tied notes into a single, longer note and removing notes with the MusicXML `print-object=no` attribute, as they would not be visible to a human performer.

To guarantee non-overlapping sets with robust evaluation across all composers in the dataset, dataset splits are created using the following procedure:

For each composer, we select one piece as a test piece and use all performances of this piece for the test set, yielding 59 instances. 90% of all remaining pieces are used in the training set and 10% in the validation set. Table 2 shows the resulting full dataset split statistics.

To complement this labeled dataset, we also construct an unpaired dataset consisting of 58,646 public domain MusicXML files from Muscores, without corresponding P-MIDI. These scores are filtered for overlap with the labeled dataset to avoid data leakage.

Dataset	Train	Validation	Test	Total
Performances	822	86	59	967
Distinct pieces	176	16	14	206
P-MIDI Notes (10^3)	2510	300	220	3030
Score Notes (10^3)	2462	295	215	2972

Table 2. Dataset statistics for ASAP [41] after excluding instances with mismatched annotations.

3.1.2 Training batch construction

All training batches consist of 32 sequences of 512 notes each, equally split between labeled and unpaired datasets. To sample instances from these heterogeneous datasets, we adopt two different procedures.

Labeled data. We first use the beat-level correspondences to coarsely align input and output sequences by sorting notes into inter-beat intervals according to their onset time. Although this correspondence is exact for the Mu-

to information leakage as highly regular onset/offset alignment remains in the ‘performance’ data. Details of the laborious alignment process are available at <http://www.piano-midi.de/technic.htm>.

sicXML score data, human performances introduce variations to the P-MIDI data, causing some notes to not align perfectly with annotated beats. As a result, performance notes that occur shortly before the annotated beat time may musically belong into the next inter-beat interval and vice-versa. To solve this issue, we follow a greedy optimization strategy that minimizes mismatched pitches between performance and score in each beat interval. If a performed note occurred within 50ms of a beat, and moving it to the previous/next inter-beat-interval reduces the number of mismatched pitches in both intervals, the move is performed. Where necessary for alignment, we add spacing tokens (sp_j) at the end of inter-beat intervals. Given correct beat annotations, this procedure yields good alignment even in non-trivial situations like trills, where multiple MIDI notes correspond to just one MusicXML note.

Unpaired data. In this case, only MusicXML data is available. This could be used to simply pre-train the decoder stack in an autoregressive fashion; however, we found this procedure to be ineffective. We thus aim to incorporate the encoder into the training process and construct a surrogate input token stream by reusing the output pitch tokens p_j as input for the encoder model p_i and mark the input sequence using conditioning tokens c_i . All other input tokens (o_i , d_i , and v_i) are masked. As demonstrated in Section 3.4, this significantly enhances the effectiveness of training on unpaired data. Without input timing and velocity streams, the model has far less information to make predictions. To make the learning objective more feasible, we decrease the prior-token dropout probability to 50% (compared to 75% for paired data), improving training efficiency without compromising inference time behavior (see also Section 2.4). Similar to conditioning masks in diffusion models, we also feed a binary token (c_i) to the encoder which indicates that no real P-MIDI conditioning information from the labeled dataset is provided, resolving ambiguity about whether input tokens are masked/dropped out or simply not available. The addition of this token improves the effectiveness of training on unlabeled data (see Table 6). When training on labeled data and during inference, its embedding is set to 0 and can thus be omitted.

3.1.3 Data augmentation

During training, four types of data augmentation are used to combat overfitting:

- **Transposition:** Shift all pitches in the input and output up or down by up to 12 semitones; notes falling outside the MIDI pitch range are shifted inward by one octave. Accidentals are modified accordingly, following [17].
- **Global tempo:** Change the timing data of the input MIDI notes by a factor of $\lambda \sim \mathcal{U}(0.8, 1.2)$.
- **Duration jitter:** To simulate human performance variations, performed note durations are additionally rescaled by a small amount of noise $\sim \mathcal{U}(0.95, 1.05)$.
- **Onset jitter:** All between-note intervals of the input MIDI are changed according to $\tilde{o}_{i+1} - \tilde{o}_i = (o_{i+1} - o_i) \cdot \mathcal{N}(1, 0.05^2)$.

Method	MUSTER [14]						ScoreSimilarity [27,42]					
	\mathcal{E}_p	\mathcal{E}_{miss}	\mathcal{E}_{extra}	\mathcal{E}_{onset}	\mathcal{E}_{offset}	\mathcal{E}_{avg}	\mathcal{E}_{miss}	\mathcal{E}_{extra}	$\mathcal{E}_{dur.}$	\mathcal{E}_{staff}	\mathcal{E}_{stem}	$\mathcal{E}_{spell.}$
Neural Beat Tracking (improved) [19]	2.02	6.81	9.01	68.28	54.11	28.04	17.10	17.67	66.98	6.86	-	9.71
MuseScore [43]	2.41	7.35	9.64	47.90	49.44	23.35	16.17	16.74	55.23	21.87	29.87	9.69
Finale [44]	2.47	10.10	13.46	31.85	45.34	20.64	14.72	16.43	53.35	21.79	26.74	15.34
HMMs + Heuristics (J-Pop) [14] [†]	2.09	6.38	8.67	25.02	29.21	14.27	10.80	11.39	71.38	-	-	-
HMMs + Heuristics (classical) [14] [†]	2.11	6.47	8.75	22.58	29.84	13.95	10.74	11.28	64.73	-	-	-
Ours	3.11	7.56	6.44	15.55	23.84	11.30	12.69	9.06	51.86	6.62	25.03	8.69

Table 3. Comparative quantitative evaluation on the ASAP test set. All prior methods produce quantized MIDI and require MuseScore 4 to perform typesetting and conversion to MusicXML. [†]: the reported metrics are slightly optimistic as some pieces of the test set appeared in the training data for subcomponents of this method only.

3.2 Metrics

To conduct fine-grained comparisons, we use both MUSTER [12, 18] and ScoreSimilarity [27, 42] as evaluation metrics for PM2S performance².

MUSTER especially focuses on high-level accuracy and rhythmic structure, with sub-metrics for note-level edit-distance (\mathcal{E}_p , \mathcal{E}_{miss} , \mathcal{E}_{extra}), rhythm correction (\mathcal{E}_{onset}), defined by the amount of scale and shift operations required to correctly align every note’s onset with the ground truth sequence, and \mathcal{E}_{offset} , which measures the accuracy of the predicted note’s musical durations. While edit-distance metrics primarily reflect the melodic correctness of a score, \mathcal{E}_{onset} and \mathcal{E}_{offset} serve as good indicators of rhythmic understanding and visual clarity of the resulting notation.

ScoreSimilarity also tracks edit-distances (\mathcal{E}_{miss} , \mathcal{E}_{extra}) but additionally allows the evaluation of notational details such as stem direction (\mathcal{E}_{stem}), pitch spelling ($\mathcal{E}_{spell.}$), or hand/staff assignment (\mathcal{E}_{staff}). We extend ScoreSimilarity to ornaments by adding F1-scores for grace, staccato, and trill. To harmonize the scores reported by both metrics, we opt to report normalized error scores and F1-scores instead of absolute error counts as originally proposed in [42].

	ScoreSimilarity [27, 42]					
	\mathcal{E}_{staff}	\mathcal{E}_{stem}	$\mathcal{E}_{spell.}$	$F1_{grace}$	$F1_{staccato}$	$F1_{trill}$
SOTA [19, 43, 44]	6.86	26.74	9.69	-	-	-
Ours	6.62	25.03	8.69	27.80	18.19	54.64

Table 4. Predicting score ornaments and visual details.

Method	L	ScoreSimilarity [27, 42]				
		\mathcal{E}_{miss}	\mathcal{E}_{extra}	$\mathcal{E}_{duration}$	\mathcal{E}_{staff}	\mathcal{E}_{stem}
Suzuki [27]	12954	12.53	4.21	0.53	0.03	5.40
Octuple [28]	3697	3.56	4.74	17.51	16.34	30.94
Ours	3697	2.64	0.40	3.72	0.01	1.54
MIDI score ³	-	3.04	4.64	13.63	3.41	-
MusicXML	-	0.00	0.00	0.00	0.00	0.00

Table 5. Comparison of score representation schemes by sequence lengths and representation error rates.

² MV2H [45] was also considered, but its alignment procedure was prohibitively slow on real-world scores with thousands of notes. Alignment is necessitated by the lack of one-to-one correspondence labels.

³ The MIDI files were created from MusicXML with MuseScore 4.0.

3.3 Comparative experiments

PM2S. In Table 3, we compare our model to the best publicly available PM2S systems. Our baselines include the popular commercial programs MuseScore [43] and Finale [44], the strongest HMM-based approach [14], and the highest-performance deep learning model [19], which relies on neural beat tracking. Where necessary for evaluation, MuseScore 4 is employed to convert quantized score MIDI predictions to MusicXML. We also compare with an improved version of the reference implementation of [19], which removes the time-signature⁴ and note-value prediction modules. However, as noted in Section 3.1.1, beat-tracking still struggles on real-world P-MIDI, lagging behind [14] and other options in rhythm quantization.

In contrast, our method predicts notation with significantly more accurate rhythm ($\mathcal{E}_{onset/offset}$), note values ($\mathcal{E}_{offset/duration}$), and fewer extraneous notes (\mathcal{E}_{extra}). In practice, this is reflected in better alignment of notes with barlines and more concise notation than alternative approaches. While all baselines pass the input pitch sequence directly to the output, our setup requires the model to rebuild the full sequence from scratch, leading to more missed notes ($\mathcal{E}_{p/miss}$). Decoupling the output pitch sequence from the input is key to our method, enabling training without one-to-one correspondences and predicting many-to-one relationships like trills. In fact, many ‘misses’ occur because our approach notated a trill where the ground truth score contains multiple alternating notes, with minimal impact on the resulting score’s quality from a human perspective.

For sample scores and visual comparisons with baseline approaches, we refer to the supplementary material.

Notation details. To our knowledge, our method is the first PM2S system to predict note-level attributes beyond timing, pitch, and staff assignment. The model also estimates staccato, grace notes, and trill marks, which are crucial for human performers. Given the data imbalance – for instance, trills account for only 0.15% of notes – achieving high F1 scores is extremely challenging. Table 4 shows that our approach predicts more accurate stem directions, pitch-spelling, and staff assignments, while exhibiting relatively good performance on grace and trill notes.

Tokenization scheme. We evaluate our MusicXML to-

⁴ We found that assuming a fixed $\frac{4}{4}$ time signature improves results.

Modification	MUSTER [14]						ScoreSimilarity [27, 42]					
	\mathcal{E}_p	\mathcal{E}_{miss}	\mathcal{E}_{extra}	\mathcal{E}_{onset}	\mathcal{E}_{offset}	\mathcal{E}_{avg}	\mathcal{E}_{miss}	\mathcal{E}_{extra}	$\mathcal{E}_{duration}$	\mathcal{E}_{staff}	\mathcal{E}_{stem}	
1 no data augmentation	10.96	38.05	38.23	37.89	51.34	35.30	49.99	47.98	32.43	7.30	13.10	
2 BiLSTM backbone	4.01	22.28	16.02	38.30	60.30	28.12	26.75	16.29	55.46	7.55	21.08	
3 BiGRU backbone	3.85	19.49	13.56	30.98	49.59	23.55	23.84	14.39	43.69	7.50	22.33	
4 no beat-alignment	3.57	25.53	10.91	19.25	29.60	17.77	33.40	10.91	41.06	6.03	20.20	
5 no transpose	4.86	12.04	9.89	19.36	29.66	15.16	17.99	12.76	50.54	7.92	24.93	
6 no 24-div quantization	3.02	10.96	8.99	19.71	31.73	14.88	15.43	10.50	74.92	7.54	26.84	
7 ALiBi pos. enc.	3.12	11.22	8.07	17.85	27.66	13.58	16.24	10.11	55.67	7.35	27.92	
8 no surrogate pitch	4.11	9.83	8.20	16.89	26.66	13.14	15.21	10.79	49.41	6.81	25.84	
9 no onset jitter	3.20	8.66	7.23	17.04	27.75	12.78	13.42	9.38	51.33	8.59	25.93	
10 no tempo augmentation	3.18	8.74	7.21	17.25	27.45	12.76	13.60	9.50	54.20	7.78	26.95	
11 no conditioning token	3.09	9.10	7.10	16.82	27.01	12.62	13.98	9.18	52.98	8.33	26.57	
12 no duration jitter	3.39	8.53	7.32	16.54	26.92	12.54	13.56	9.59	50.68	8.19	27.69	
13 sinusoidal pos. enc.	3.50	8.29	6.83	16.71	27.35	12.49	13.41	9.36	51.68	7.35	25.81	
14 Ours	3.11	7.56	6.44	15.55	23.84	11.30	12.69	9.06	51.86	6.62	25.03	

Table 6. Ablation study for key design decisions. Grayed out values do not reflect the true model performance as a large fraction of notes are misaligned during metric computations, leading to incorrect results. Rows are organized by \mathcal{E}_{avg} .

kenization against prior methods and score-derived MIDI files by converting ground-truth scores to a new format and then comparing the reconstructions to the originals.

Table 5 shows that our approach yields $3.5\times$ shorter sequence lengths than prior MusicXML tokenizations while maintaining more detail than alternatives. Furthermore, it highlights MIDI’s shortcomings as a notation format; both ground-truth MIDI scores and MIDI-based tokenizations [28] exhibit lower fidelity than MusicXML-derived tokenizations and particularly high error rates for details like stem directions, which are not supported by MIDI.

3.4 Ablation study

Our ablation study in Table 6 shows the impact of key design choices.

Backbone architecture. The transformer architecture is much stronger than classic recurrent networks like bidirectional LSTMs [33] and GRUs [46] when trained on the same data (rows 3 & 2). We also evaluate the effectiveness of the conditioning token (row 11) and demonstrate the impact of feeding surrogate pitch information to the encoder for unpaired data (row 8).

Positional encoding. We compare rotary embeddings with standard sinusoidal embeddings [32] (row 13) and ALiBi [47] (row 7) and find that sinusoidal embeddings perform slightly weaker than rotary encoding while ALiBi yields significantly worse results.

Tokenization. We demonstrate the impact of our tokenization scheme’s quantization by changing the encoding scheme to quantize durations by 32nd-divisions instead of 24th (row 6). This has a particularly strong impact on metrics for rhythm and note values (\mathcal{E}_{onset} , \mathcal{E}_{offset} , $\mathcal{E}_{duration}$).

Data augmentation & alignment. Data augmentation is crucial to the effectiveness of our approach (row 1). Rows 5, 9, 10, and 12 show that using all 4 augmentation types combined yields the best results. We also demonstrate that using our greedy beat-level note alignment algorithm significantly improves performance compared to unoptimized input and output sequence alignment (row 4).

3.5 Scaling

To assess model performance as datasets grow, we conduct experiments with varying amounts of paired and unpaired data (see Table 7). We observe a clear trend where increasing the amount of paired and unpaired data improves final performance across most metrics. The benefits of adding 10,000 unpaired scores are similar to expanding from 100 paired to 822 paired training pieces. While training on unpaired data is less sample-efficient, the labeling cost-savings may make it worthwhile nonetheless. The results suggest that our method is able to leverage additional data well and that training on larger (paired & unpaired) datasets could lead to significant further improvements.

Number of pieces		ScoreSimilarity [42]				
Paired	Unpaired	\mathcal{E}_{miss}	\mathcal{E}_{extra}	$\mathcal{E}_{duration}$	\mathcal{E}_{staff}	\mathcal{E}_{stem}
100	-	22.04	17.78	54.22	8.77	28.43
822	-	14.44	10.11	50.77	7.78	27.56
100	10,000	15.40	12.03	55.98	7.53	26.93
822	10,000	13.44	9.49	49.18	8.22	27.85
822	58,686	12.69	9.06	51.86	6.62	25.03

Table 7. The effects of training dataset size.

4. CONCLUSION

We presented a flexible, robust, and conceptually simple approach to convert P-MIDI into musical notation and showed that a standard sequence-to-sequence transformer model can outperform existing methods that benefit from extensive domain-specific optimizations. We also introduced a compact tokenization method for symbolic music data that is extensible to further notational elements in the future. Furthermore, we demonstrate that leveraging large unpaired training datasets can improve model performance and enhance the fidelity of predicted scores. Future efforts could add features like explicit key and time signature prediction, tempo marks, and expand to other musical genres, instruments, and notation systems.

5. ETHICS STATEMENT

The proposed system is primarily trained on classical piano music by European composers engraved in standard Western musical notation. While preliminary experiments show that the method generalizes out-of-genre to modern piano pop music, our approach nonetheless excludes a large body of musical work notated in different formats. Future work should aim to address this imbalance and create systems that can be useful to an even wider audience.

6. ACKNOWLEDGEMENTS

We would like to thank Andrew McLeod for his assistance and feedback on a draft of this work via the New-to-ISMIR paper mentoring program. This work was partially supported by the ERC Starting Grant SpatialSem (101076253).

7. REFERENCES

- [1] N. Orio and D. Schwarz, "Alignment of monophonic and polyphonic music to a score," in *International Computer Music Conference (ICMC)*, 2001, pp. 1–1.
- [2] C. Hawthorne, E. Elsen, J. Song, A. Roberts, I. Simon, C. Raffel, J. Engel, S. Oore, and D. Eck, "Onsets and frames: Dual-objective piano transcription," *arXiv preprint arXiv:1710.11153*, 2017.
- [3] E. Benetos, S. Dixon, Z. Duan, and S. Ewert, "Automatic music transcription: An overview," *IEEE Signal Processing Magazine*, vol. 36, no. 1, pp. 20–30, 2018.
- [4] M. Good, "MusicXML for notation and analysis," *The virtual score: representation, retrieval, restoration*, vol. 12, no. 113–124, p. 160, 2001.
- [5] P. Desain and H. Honing, "The quantization of musical time: A connectionist approach," *Computer Music Journal*, vol. 13, no. 3, pp. 56–66, 1989.
- [6] C. Raphael, "Automated rhythm transcription," in *ISMIR*, vol. 2001, 2001, pp. 99–107.
- [7] A. Cogliati, D. Temperley, and Z. Duan, "Transcribing human piano performances into music notation," in *ISMIR*, 2016, pp. 758–764.
- [8] D. Temperley, "A unified probabilistic model for polyphonic music analysis," *Journal of New Music Research*, vol. 38, no. 1, pp. 3–18, 2009.
- [9] H.-W. Nienhuys and J. Nieuwenhuizen, "LilyPond, a system for automated music engraving," in *Proceedings of the XIV Colloquium on Musical Informatics (XIV CIM 2003)*, vol. 1. Citeseer, 2003, pp. 167–171.
- [10] E. Nakamura, N. Ono, and S. Sagayama, "Merged-output HMM for piano fingering of both hands," in *ISMIR*, 2014, pp. 531–536.
- [11] E. Nakamura, K. Yoshii, and S. Sagayama, "Rhythm transcription of polyphonic piano music based on merged-output hmm for multiple voices," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 4, pp. 794–806, 2017.
- [12] E. Nakamura, E. Benetos, K. Yoshii, and S. Dixon, "Towards complete polyphonic music transcription: Integrating multi-pitch detection and rhythm quantization," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 101–105.
- [13] E. Nakamura, K. Yoshii, and S. Dixon, "Note value recognition for piano transcription using markov random fields," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 9, pp. 1846–1858, 2017.
- [14] K. Shibata, E. Nakamura, and K. Yoshii, "Non-local musical statistics as guides for audio-to-score piano transcription," *Information Sciences*, vol. 566, pp. 262–280, 2021.
- [15] R. G. C. Carvalho and P. Smaragdis, "Towards end-to-end polyphonic music transcription: Transforming music audio directly to a score," in *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2017, pp. 151–155.
- [16] M. A. Román, A. Pertusa, and J. Calvo-Zaragoza, "Data representations for audio-to-score monophonic music transcription," *Expert Systems with Applications*, vol. 162, p. 113769, 2020.
- [17] F. Foscarin, N. Audebert, and R. Fournier-S'Niehotta, "PKSpell: Data-driven pitch spelling and key signature estimation," *arXiv preprint arXiv:2107.14009*, 2021.
- [18] Y. Hiramatsu, E. Nakamura, and K. Yoshii, "Joint estimation of note values and voices for audio-to-score piano transcription," in *ISMIR*, 2021, pp. 278–284.
- [19] L. Liu, Q. Kong, G. Morfi, E. Benetos *et al.*, "Performance MIDI-to-score conversion by neural beat tracking," 2022.
- [20] T. Cheng and M. Goto, "Transformer-based beat tracking with low-resolution encoder and high-resolution decoder," in *ISMIR 2023 Hybrid Conference*, 2023.
- [21] C. Walshaw, "The abc music standard 2.1," URL: <http://abcnotation.com/wiki/abc:standard:v2>, vol. 1, 2011.
- [22] D. Huron, "Music information processing using the Humdrum toolkit: Concepts, examples, and lessons," *Computer Music Journal*, vol. 26, no. 2, pp. 11–26, 2002.
- [23] R. Kelz, M. Dorfer, F. Korzeniowski, S. Böck, A. Arzt, and G. Widmer, "On the potential of simple frame-wise approaches to piano transcription," *arXiv preprint arXiv:1612.05153*, 2016.

- [24] Y.-S. Huang and Y.-H. Yang, “Pop music transformer: Beat-based modeling and generation of expressive pop piano compositions,” in *Proceedings of the 28th ACM international conference on multimedia*, 2020, pp. 1180–1188.
- [25] C. Hawthorne, I. Simon, R. Swavely, E. Manilow, and J. Engel, “Sequence-to-sequence piano transcription with transformers,” in *Proc. of the 22nd Int. Society for Music Information Retrieval Conf.*, Online, 2021.
- [26] J. Gardner, I. Simon, E. Manilow, C. Hawthorne, and J. Engel, “MT3: Multi-task multitrack music transcription,” *arXiv preprint arXiv:2111.03017*, 2021.
- [27] M. Suzuki, “Score transformer: Generating musical score from note-level representation,” in *Proceedings of the 3rd ACM International Conference on Multimedia in Asia*, 2021, pp. 31:1–31:7.
- [28] M. Zeng, X. Tan, R. Wang, Z. Ju, T. Qin, and T.-Y. Liu, “MusicBERT: Symbolic music understanding with large-scale pre-training,” *arXiv preprint arXiv:2106.05630*, 2021.
- [29] H.-W. Dong, K. Chen, S. Dubnov, J. McAuley, and T. Berg-Kirkpatrick, “Multitrack music transformer,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023.
- [30] A. Lv, X. Tan, P. Lu, W. Ye, S. Zhang, J. Bian, and R. Yan, “GETMusic: Generating any music tracks with a unified representation and diffusion framework,” *arXiv preprint arXiv:2305.10841*, 2023.
- [31] J. Su, Y. Lu, S. Pan, A. Murtadha, B. Wen, and Y. Liu, “Roformer: Enhanced transformer with rotary position embedding,” *arXiv preprint arXiv:2104.09864*, 2021.
- [32] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, 2017.
- [33] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [34] J. L. Ba, J. R. Kiros, and G. E. Hinton, “Layer normalization,” *arXiv preprint arXiv:1607.06450*, 2016.
- [35] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, “Language models are few-shot learners,” *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.
- [36] N. Shazeer, “Glu variants improve transformer,” *arXiv preprint arXiv:2002.05202*, 2020.
- [37] I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” *arXiv preprint arXiv:1711.05101*, 2017.
- [38] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, “Scheduled sampling for sequence prediction with recurrent neural networks,” *Advances in neural information processing systems*, vol. 28, 2015.
- [39] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting,” *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.
- [40] V. Emiya, N. Bertin, B. David, and R. Badeau, “MAPS - A piano database for multipitch estimation and automatic transcription of music,” 2010.
- [41] F. Foscarin, A. Mcleod, P. Rigaux, F. Jacquemard, and M. Sakai, “ASAP: A dataset of aligned scores and performances for piano transcription,” in *International Society for Music Information Retrieval Conference*, 2020, pp. 534–541.
- [42] A. Cogliati and Z. Duan, “A metric for music notation transcription accuracy,” in *ISMIR*, 2017, pp. 407–413.
- [43] MuseScore B.V., “MuseScore: A free and open-source music notation software,” <https://musescore.org/>, 2002, accessed: 2024-02-28.
- [44] MakeMusic, Inc., “Finale version 27,” <https://www.finalemusic.com/>, 1988, accessed: 2024-02-28.
- [45] A. McLeod, “Evaluating non-aligned musical score transcriptions with MV2H,” *arXiv preprint arXiv:1906.00566*, 2019.
- [46] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling,” *arXiv preprint arXiv:1412.3555*, 2014.
- [47] O. Press, N. A. Smith, and M. Lewis, “Train short, test long: Attention with linear biases enables input length extrapolation,” *arXiv preprint arXiv:2108.12409*, 2021.

FROM REAL TO CLONED SINGER IDENTIFICATION

Dorian Desblancs Gabriel Meseguer-Brocal Romain Hennequin Manuel Moussallam
Deezer Research, Paris, France

research@deezer.com

ABSTRACT

Cloned voices of popular singers sound increasingly realistic and have gained popularity over the past few years. They however pose a threat to the industry due to personality rights concerns. As such, methods to identify the original singer in synthetic voices are needed. In this paper, we investigate how singer identification methods could be used for such a task. We present three embedding models that are trained using a singer-level contrastive learning scheme, where positive pairs consist of segments with vocals from the same singers. These segments can be mixtures for the first model, vocals for the second, and both for the third. We demonstrate that all three models are highly capable of identifying real singers. However, their performance deteriorates when classifying cloned versions of singers in our evaluation set. This is especially true for models that use mixtures as an input. These findings highlight the need to understand the biases that exist within singer identification systems, and how they can influence the identification of voice deepfakes in music.

1. INTRODUCTION

In April 2023, the track “Heart on my Sleeve” by an anonymous TikTok user Ghostwriter977 put the music industry in a frenzy [1, 2]. The artist used artificial intelligence (AI) based cloning technologies to turn their voice into Drake and the Weeknd’s [3], two of the most popular singers in the world. The song became very popular across music streaming platforms, before being removed by demand of the original artists’ right owners. This situation raised the need for singer identification systems that can also identify the original singer a synthetic voice was generated from.

In this paper, we train three embedding models for singer identification using a singer-level contrastive learning scheme, where positive pairs consist of segments with vocals of the same singers whilst negatives come from different singers. These samples can be mixtures for the first model, vocals for the second, and both for the third. The models are then evaluated on real singers using novel splits of two open datasets, the Free Music Archive (FMA) [4, 5]


and MTG-Jamendo (MTG) [6], and a closed dataset consisting of 176,141 songs that span 7500 popular singers. We use this dataset due to its scale and the fact that its singers are often the target of music voice deepfakes, some of which we use in this paper. We demonstrate that all three models are highly capable of classifying real voices, though genres that use effects on vocals, such as hip-hop, pop, and electronic music, and singers with long discographies can be much harder to classify. We then test whether the performance of our models generalizes to cloned voices of singers present in our closed dataset, using songs from YouTube. In this context, singers are often cloned onto famous instrumentals, or instrumentals that differ greatly from their usual environments. We find that the performance of all three models deteriorates quite significantly. This is especially true for models that use mixtures as inputs. We hope that these findings can be useful for future singer identification works. We believe that these should aim to design systems that can identify both a singer’s real and synthetic voice, in the hopes of combating the growing problem of voice deepfakes in music.

We summarize the contributions of this work as follows:

- 1) We evaluate singer identification systems on songs with real singers and cloned voices of some of the same singers.
- 2) We offer a detailed inspection of their performance, and demonstrate that these systems struggle to classify synthetic voices, genres where audio effects are applied to natural voices, and singers with long discographies. For synthetic voices, this decline is even greater when instrumental information is present during training, and highlights the need to understand the biases that exist within singer identification systems.
- 3) We open source singer identification splits of two open datasets, the FMA and MTG, that can serve as future performance benchmarks for the task of singer identification in polyphonic mixtures.¹

2. RELATED WORK

Singer identification, has been a staple of the music information retrieval (MIR) community for more than twenty years [7, 8]. Early approaches aimed to attenuate the instrumental parts of a song through the use of vocal melody or pitch extraction and voice re-synthesis and detection algorithms [9–11]. Classic features, such as mel-frequency cepstral coefficients (MFCCs), were then computed on these signals and used as inputs for a classifier. The improvements in music source separation [12, 13] then led re-

 © D. Desblancs, G. Meseguer-Brocal, R. Hennequin, and M. Moussallam. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** D. Desblancs, G. Meseguer-Brocal, R. Hennequin, and M. Moussallam, “From Real to Cloned Singer Identification”, in *Proc. of the 25th Int. Society for Music Information Retrieval Conf.*, San Francisco, United States, 2024.

¹ <https://github.com/deezer/real-cloned-singer-id>

searchers to build models that classify singers using the vocals of each song [14, 15]. More recently, self-supervised methods that process the vocal stem of each track have been shown to be effective for singer identification [16, 17]. However, source separation is computationally costly. As such, these algorithms are hard to deploy on catalogues that span millions of tracks. Several works have attempted to build embedding models for singer identification that use mixtures as inputs [18, 19], most notably by using triplet learning where anchors and positives come from the same singers in different instrumental environments.

The work in this paper focuses on testing whether singer identification systems trained on real voices generalize to cloned voices of the same singers. This task must not be confused with the task of singing voice deepfake detection, which has very recently emerged in the signal processing community [20, 21]. Both works introduce datasets for Chinese singing voice spoofing detection, and demonstrate that state-of-the-art speech deepfake detectors fail to accurately predict whether the songs in their datasets are deepfakes. After supervised training, their performance is improved. However, [21] also finds that the classifiers are not robust to unseen singers, languages, or musical contexts, suggesting the need for more complex methods.

Finally, audio embeddings learned using artist-based sampling scheme have been used in [22]. The authors used metric learning, with anchors and positives coming from the same artists, to train a neural network for artist disambiguation. More recently, [23] used sampling at the artist level for contrastive learning and downstream tasks such as genre and mood classification or music tagging.

3. EXPERIMENTAL SETUP

In this section, we first present the datasets used throughout this paper. We then present the setup used to train an embedding model for singer disambiguation using contrastive learning. Finally, we present how this model is used for singer identification.

3.1 Datasets

We collect a vast number of popular, commercial, and annotated songs for both training and evaluating the embedding models. The data and their singer annotations come from four sources: Deezer, MusicBrainz, Wikidata, and Discogs [24]. The latter three are publicly available. In total, we collect more than four million tracks that span ~ 2.6 million artists. We then filter out all tracks that are not comprised of vocal segments at least 75% of the time. For this, we use a simple deep learning model that classifies three-second segments into either an instrumental class or a vocal class across all songs. We then filter out all unique singers that do not have at least two tracks. This leaves us with 37,525 singers. 7500 of the ones that have at least seven tracks are used for our singer identification task. The remaining 30,025 are used to train and validate our embedding models using contrastive learning.

We then gather 377 tracks, from YouTube, with cloned

Dataset	No. Singers	No. Songs	Songs/Singer
Train	25929	181989	≥ 2
Validation	4096	8192	2
Closed	7500	176141	≥ 7
FMA	1019	11676	≥ 5
MTG	572	7710	≥ 5
Cloned	67	377	N.A.

Table 1. Attributes of each dataset used in this paper. The train and validation sets are used for training the embedding models. Upon initial collection, validation singers can have more than two mostly-vocal tracks; we however randomly select a segment with vocals from two tracks to keep the set constant. The closed, FMA, and MTG datasets are used for real singer identification. The cloned dataset contains songs collected from YouTube in which synthesized voices of real singers are used. The original singers in this dataset are present in the closed dataset.

voices from 67 singers in our closed dataset. These are used to test our embedding models on music voice deepfakes. We also test our models on two open music tagging datasets for real singer identification: the FMA and MTG. For these, we first gather their artist tags. We then filter out songs that are not comprised of segments with vocals at least 50% of the time. Artists with less than five songs are also removed. This leaves us with 1019 artists for the FMA and 572 artists for the MTG. Unlike the commercial, closed dataset, each song can contain more than one singer. We however postulate that the trends observed in the results are highly indicative of our models’ performances on the singer identification task. We publish the subsets of data we used on these datasets for reproducibility and to serve as future benchmarks. To the best of our knowledge, other open singer identification datasets, such as the VocalSet [25] and M4Singer [26], only contain snippets of a capella singing voices. We hope future singer identification systems will also be evaluated on our proposed, more authentic musical data: singers singing to an instrumental. Table 1 displays the attributes of each of these sets of data.

3.2 Singer-Level Contrastive Learning

We train the embedding models in a contrastive learning way to predict whether two songs are from the same singer. During each training iteration, we begin by drawing a batch of $B = 128$ positive pairs, which correspond to pairs of segments with singing. These pairs are drawn on the fly from different songs of the same singer. For our Mixture model, these segments come from songs’ mixtures. For our Vocal model, these segments come from the vocal stem generated by Demucs [27, 28]. Finally, for our Hybrid model, these segments are randomly sampled from either; the following positive pairings are possible during the contrastive learning task: vocal-vocal, vocal-mixture, mixture-vocal, and mixture-mixture. This is done to better disambiguate singing voices, without the need for source separation during the downstream singer identification task. All segments are sampled at 16000 Hz and have a six-

second duration. We then compute their mel-spectrograms and pass these through the small version of the transformer model from [29]. We use a FFT size of 800, a hop length of 400, and a total of 128 mel bins for our mel-spectrogram operation. The transformer model then maps the resulting 128×240 tensor to an embedding of size 2048. Similarly to [30–32], these embeddings are passed through a fully-connected projector head. In our case, this head maps our embeddings to outputs of dimension 2048, 1024, and 2048, and uses batch-normalization [33] and ReLU activations. Let us denote the resulting projections by y_i , where $i \in [1, 2B]$. For each positive pair (i, j) , we compute, and aim to minimize, the normalized temperature-scaled cross-entropy, or NT-Xent [30], loss function, defined as:

$$\ell_{i,j} = -\log \frac{\exp\left(\frac{1}{T} S(y_i, y_j)\right)}{\sum_{k=1}^{2B} \mathbb{1}_{[k \neq i]} \exp\left(\frac{1}{T} S(y_i, y_k)\right)}, \quad (1)$$

where the indicator function $\mathbb{1}_{[k \neq i]}$ evaluates to 1 iff $k \neq i$. T is a temperature parameter that helps the model learn from hard negatives. In our case, we set $T = 0.2$ following [34]. $S(u, v)$ denotes the cosine similarity between vectors u and v . The average loss value is then backpropagated to the model. We use the ADAM optimizer [35] throughout training, with an initial learning rate of 0.0001. This value is decreased by a factor of 0.5 at every 25-epoch validation loss plateau. Note that one epoch corresponds to 32 training and validation iterations. We stop training when the validation loss has plateaued for 100 epochs.

3.3 Singer Identification

We then train classifiers with the same architecture as the projector head from the previous section upon the frozen transformer models. We evaluate these classifiers on two sets of data for real singer identification: the FMA and MTG open datasets and the closed set. For all of these, we randomly set aside one track for testing and one track for validation per singer. These are constant throughout all our experiments for reproducibility purposes. Note that four segments from each validation track are selected randomly at the beginning of each singer identification experiment to keep the validation set constant. At least three tracks per singer are then used for training. During each training iteration, we select a segment with vocals on the fly to construct batches of size 100. We minimize a Cross Entropy loss [36] using the ADAM optimizer with an initial learning rate of 0.01. This value is decreased by a factor of 0.1 every 10-epoch validation loss plateau. Here, one epoch is, again, equal to 32 training and validation iterations. We stop training when the validation loss has plateaued for 20 epochs. For each dataset’s test tracks, the final singer prediction is obtained using a majority vote scheme, where each segment with vocals is passed through the frozen embedding model and classification head. The singer with the most “votes” is then used as the track’s final output.

We report all our results using 10 runs. For both open datasets, we report results using all singers. On the other hand, for our 7500-singer closed set, we report results from

100 to 1000 classes. During each run, we randomly sample a subset of singers, on which we then train and evaluate a classifier. Finally, for the cloned singers dataset, we: 1) train models to classify 100 to 1000 singers using the closed dataset; 67 of these are cloned singers, whilst the remainder are randomly sampled from the rest of our closed dataset. 2) try to classify the cloned singer of our deepfake tracks. Our goal is to evaluate whether singer identification systems trained on real singer data can correctly classify the singers’ voice deepfakes.

4. RESULTS

4.1 Open Datasets

Dataset	Model	Top-1 Acc.	Top-5 Acc.
FMA	CLMR	73.2 +/- 0.6	73.6 +/- 0.6
	Mixture	76.6 +/- 0.5	84.1 +/- 0.6
	Hybrid	77.6 +/- 0.3	85.1 +/- 0.6
	Vocal	79.9 +/- 0.4	85.7 +/- 0.3
MTG	CLMR	67.9 +/- 1.1	68.0 +/- 1.1
	Mixture	78.5 +/- 0.5	88.4 +/- 0.6
	Hybrid	79.3 +/- 1.1	88.7 +/- 0.6
	Vocal	83.2 +/- 0.6	91.3 +/- 0.6

Table 2. Singer identification results obtained on the open datasets (%). For each dataset, we report the top-1 and top-5 accuracies generated by the three models we train using singer-level contrastive learning. We also use the embeddings from [37], called CLMR, as a baseline. These embeddings are trained in a similar fashion to [31], but on $\sim 4M$ tracks, and are used for both training and testing our classification heads to generate these results. We display the means and standard deviations over 10 runs.

The results obtained on open datasets can be visualised in Table 2. One can immediately notice that the CLMR [37] results are inferior to the singer-level embedding models’ results by at least a few percentage points. On the FMA dataset, we notice a 3.4% top-1 gap between the CLMR and Mixture models. This gap grows to more than 10 percentage points using a top-5 accuracy and is even more exacerbated on the MTG dataset and with the Hybrid and Vocal models. This highlights the fact that sampling at the singer-level is much more adapted than classic, high-performing self-supervised learning methods for pre-training a model for singer identification. We observe these gaps even though the contrastive model from [37] is trained on more than $20\times$ more tracks than the embedding models we trained for this paper.

We can also notice that the Vocal model outperforms the models that use mixtures as an input by a few percentage points. More specifically: for the MTG dataset, we observe a 3.9% gap compared to the Hybrid model on top-1 accuracy and a 2.6% gap on top-5 accuracy. The gap is less pronounced on the FMA data. We can also notice that the Hybrid model, which samples both mixtures and vocal stems during pre-training, is slightly better-performing than the mixture model, though the performance gap never exceeds

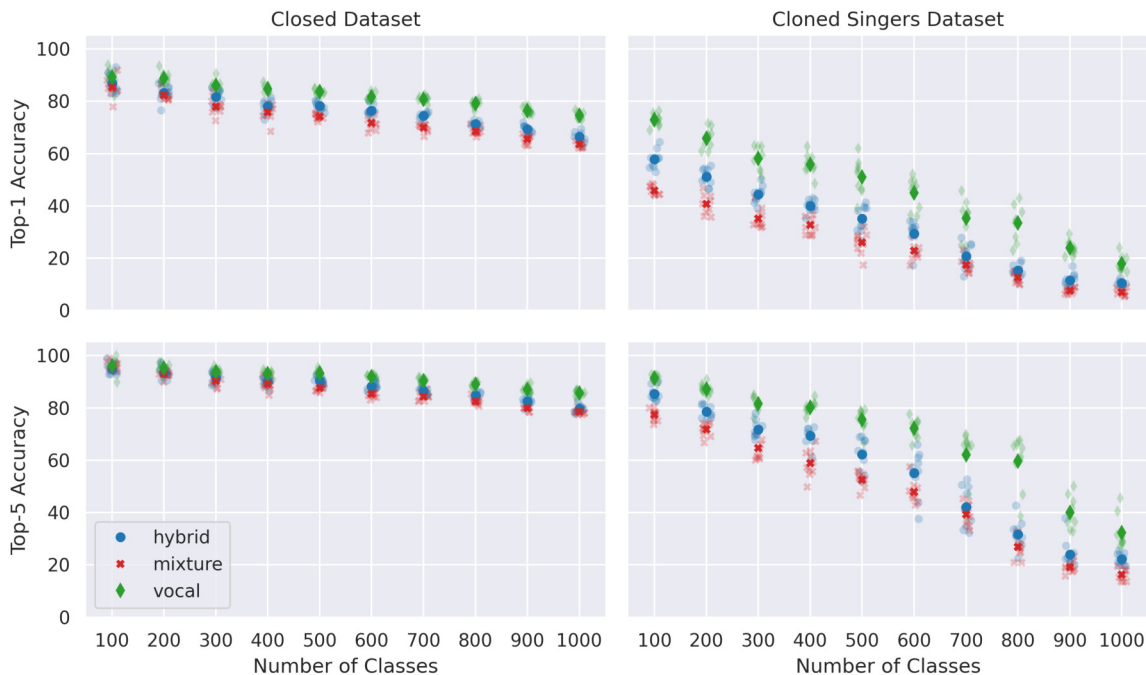


Figure 1. Singer identification results obtained on the closed (left) and cloned (right) datasets. We display results over 10 runs for 100 to 1000 singer classes. For each number of classes, we display the top-1 and top-5 accuracies for each run (pale markers), and the mean results between all runs (prominent markers). On the closed dataset, we randomly sample a subset of the 7500 singers on every run and display results on their test tracks. For the cloned dataset, we train our models to classify the 67 cloned singers and other randomly selected singers. We then display the results on the 377 spoofed tracks.

a percentage point. This highlights one of our main findings: separating vocals from the rest of the track clearly helps our models disambiguate singers between each other. However, we can obtain good performance using mixtures too. In the realm of production, where source separation can be costly memory and time-wise, the results obtained using the Hybrid or Mixture models may suffice; they may not justify the need to separate vocal stems beforehand. The results obtained on the closed dataset in the next section further emphasize this idea.

4.2 Closed Dataset

The results obtained on the closed dataset can be found on the left side of Figure 1. We observe a similar trend to the one observed on the open datasets: the Vocal model outperforms both models that use mixtures as inputs by a few percentage points. Then, the Hybrid model outperforms the Mixture model, though the gap is narrow. For example, for 400 classes, we observe mean top-5 accuracies of 89.2% for the Mixture model, 90.1% for the Hybrid model, and 93.2% for the Vocal model. For 700 classes, we observe mean top-1 accuracies of 69.8% for the Mixture model, 74.5% for the Hybrid model, and 80.9% for the Vocal model. As the number of classes grows, the gaps in performance are more pronounced, especially on the top-1 accuracy metric. We however suggest that the gap between the Vocal model and models that work on mixtures does not warrant the need for source separation in production-like environments for real-singer identification.

No. Singers	Method	Dataset	Top-1	Top-5
300	[19]	MSD	39.5	69.2
	Mixture	Closed	78.0	90.4
500	[16, 18]	MSD	47.9	71.2
	[16]	MSD	63.1	82.2
	Mixture	Closed	74.2	87.6

Table 3. Singer identification results for the same number of singers in this and previous works of the field (%).

Comparing our results to previous works in the field of music singer identification is quite difficult. These report results on private datasets [16] or on the Million Song Dataset (MSD) [18, 19, 38], a dataset whose audio is not publicly available. That is why we hope future works in singer identification will also be evaluated on the open splits we report results on in Section 4.1. We however report our worst and previous works’ results for the same number of singers in Table 3. These highlight that our methodology is at the very least on par with previous works and validate sampling at the singer level for contrastive learning when the downstream task is singer identification.

We should however point out that, even though our models identify real singers quite well, there remain open challenges. As displayed in Figure 2, our performance over musical genres is not uniform. For example, for Country and Folk music, the mean top-5 accuracies are 86.3% and 86.6%. On the other hand, for Hip-hop and Pop music, the

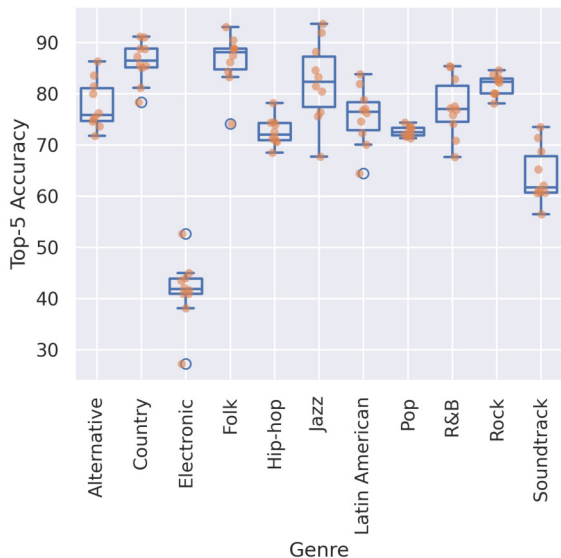


Figure 2. Vocal model performance by genre when trying to classify 1500 singers. The macro genre tags are gathered from Deezer and are unique for each test track. We display the mean top-5 accuracy for each run with the orange dots. The boxes then display the median and interquartile range (IQR) between runs. The whiskers extend to points that lie within 1.5 IQRs of the lower and upper quantiles. Finally, outlier runs have circles drawn around them. Genres containing less than 100 test tracks are omitted from this plot.

mean top-5 accuracies are 72.7% and 72.7%. The performance drops even further for Electronic music, where we observe a mean top-5 accuracy of 41.6%. The same trends can be observed for top-1 accuracy, our other models, and different numbers of classes. Hip-hop, Pop, and Electronic genres tend to employ effects such as reverb and vocoder on singing voices. These effects can change a voice’s timbre quite substantially, and seem to have an effect on our singer identification performance. On the other hand, Folk and Country tend to have natural-sounding singing voices. We suggest that future singer classification works should aim to lessen the gap between these genres, perhaps by introducing augmentations during either the embedding or classifier’s training. We also did experiments on the influence of language on performance, and did not find our results to be biased towards any of these. We found all 10, commonly-represented languages to have a median top-5 accuracy between 68 and 82% for 1500-singer identification, with substantial overlaps in distribution.

One can also notice the following trend from Figure 3: when we are trying to classify a small number of singers, having more tracks per singer for training leads to higher performance; on the other hand, when we are trying to classify a large number of singers, having fewer tracks for training leads to higher performance. For example, for 500-singer identification, we merely observe a top-1 accuracy of 78.5% when singers have 5 to 9 training tracks. This top-1 accuracy grows to 88.5% when singers have 20

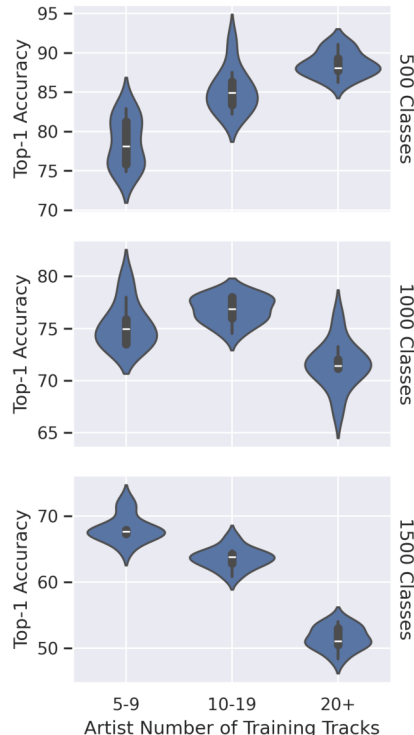


Figure 3. Vocal model performance over 500, 1000, and 1500-singer identification. We report results from each run in buckets that describe the number of training tracks per singer, that are used to train our classifiers. In the first, we display the top-1 accuracies observed for singers with only 5 to 9 training tracks. In the second, we display the top-1 accuracies observed for singers with 10 to 19 training tracks. Finally, in the last, we display the top-1 accuracies observed for singers with 20 or more training tracks. We report results using violin plots, where, for each bucket, the inner figure is a box plot similar to that in Figure 2 and the outer figure is a kernel density estimation of the data.

or more training tracks. On the other hand, for 1500-singer identification, we observe top-1 accuracies of 68.1% and 51.5% for these same buckets. These trends can also be observed on our other models and for top-5 accuracy. They suggest that, as the singer identification task gets harder, singers with more songs to their name, and most likely much longer careers, get harder to correctly classify than singers with just an extended play (EP) or album to their name. This could be due to changes in style, mixing effects, or even singing voice. We hope that future works in the field will design systems that are more robust to singing voice evolution over a variety of musical projects.

4.3 Cloned Voices

The results on our cloned dataset can be found on the right side of Figure 1. One can immediately notice a sharp decline between the performance we observed on real singers and synthetic ones. For 200-singer identification, the worst-performing model on real singers, the Mixture one, has a mean top-1 accuracy of 82.3%. In compar-

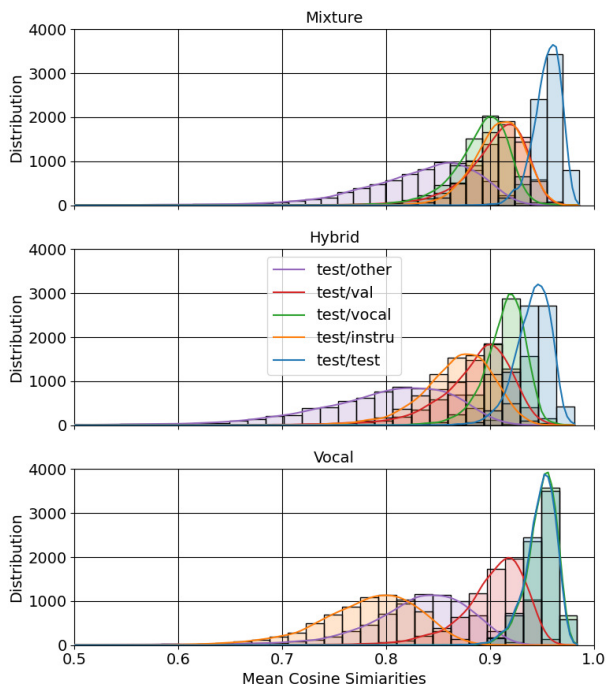


Figure 4. Mean all-pairs cosine similarity between each of the closed set singers’ test track embeddings and: in purple (test/other), the embeddings from a random track from another singer; in red (test/val), their validation track embeddings; in green (test/vocal), their test track’s vocal stem embeddings; in orange (test/instru), their test track’s instrumental stem embeddings; in blue (test/test), the other embeddings from the same track. All embeddings are generated on segments with vocals.

ison, the best-performing model on synthetic voices, the Vocal one, has a top-1 accuracy of 65.8%. For 600-singer identification, their respective top-5 accuracies are 85.3% and 72.3%. The decline in performance on cloned singers is hence quite dramatic. However, in a lot of ways, it is to be expected. Synthetic voices can sometimes be quite unrealistic depending on the voice conversion or generation techniques used, which should obviously lead to deterioration in singer identification performance.

The more striking decline is that which we observe between models themselves. More notably, the Vocal model performs substantially better than the models that use mixtures as inputs. Starting at 100 classes, the mean top-1 accuracy of the Vocal model is 65.8% versus 51.2% for the Hybrid model and 40.1% for the Mixture model. For 800 classes, we observe accuracies of 33.4% versus 15.3% and 12.7%. On the one hand, for real singers, we found the gap in performance between the Vocal and Hybrid to be minimal enough to justify using mixtures over vocal stems, and hence avoid using source separation pre-processing. Here, however, the answer is much more clear cut: the Vocal model is the only one with decent performance on cloned singer identification task, whilst the models that use mixture inputs see a very significant drop in performance.

The reason behind the performance drop between models is illustrated in Figure 4. When comparing the embed-

dings of each closed set test track to other embeddings of the same track, we see that these are very similar, with cosine similarities of $\sim 95\%$. However, the comparison with the test tracks’ stem embeddings can differ significantly. For the Mixture model, we see that the instrumental embeddings are actually more similar to the “ground truth” test track embeddings (GTEs) than the vocal embeddings, with mean similarities of 90.8% and 89.1%. Even worse, the instrumental embeddings are closer to the GTEs than the validation track’s. Hence, even though our Mixture model, like the Hybrid and Vocal models, is pre-trained to disambiguate singers, we find that its embeddings are more suitable for finding similar songs based on instrumental information than vocal information. This problem is partly solved in the Hybrid model and fully solved in the Vocal model. Note that these results extend to other vector similarity measures such as Euclidean distance.

These findings outline why the models that are trained using mixtures drop off significantly on spoofed versions of famous artists. On these tracks, singers are often used on an instrumental which is either from another famous track, or an instrumental which is very different from their usual environment. Some of the cloned tracks’ instrumentals are even present in their original tracks during training on real singers, which leads to obvious misclassifications. As such, models that bias singers towards certain types of musical backgrounds fail to correctly identify them in altered contexts. Source separation allows us to better disambiguate voices only during training, and thus classify synthetic versions of performers. In the future, perhaps reintegrating mashups to alter a singer’s context on the fly, such as was done in [18], could lead to more robust singer identification models. These could solve the two main remaining problems in the field: 1) the need for source separation pre-processing and 2) the identification of cloned versions of existing singers.

5. CONCLUSION

In this paper, we train three models using singer-level contrastive learning. The first is only trained using mixtures, the second is only trained using vocal stems, while the third is trained using both. We find that all three models are highly capable of classifying real singers, though there remain open challenges, such as classifying genres that use more vocal effects and singers with long discographies. However, all three models’ performance decreases drastically when trying to identify cloned voices of existing singers. This decrease is much more pronounced for models that are trained using mixtures. These models bias singers towards certain types of instrumentals. They therefore struggle to correctly classify them in different background music environments, such as those offered by singing voice deepfakes. By publishing our results and novel, singer identification splits of the FMA and MTG datasets, we aim to generate more research in this field of MIR. Future works could notably incorporate cloned voices in a few-shot fashion in the hopes of minimizing the gap between real and synthetic singer identification.

6. ETHICS STATEMENT

Our work offers a glimpse into how we, as a field, can identify the original singers in music voice deepfakes. It is important that outputs of systems like ours not be used as justification to make important decisions, however, such as content removal from platforms. As demonstrated in this paper, singer identification systems are often wrong; they often return false positives. This is even more true on deepfakes. As such, human emotion and decision-making should still be at the heart of the music deepfake battle. Creative, talented singers should never see their work deplatformed because a machine learning model falsely said so. The outputs of these models should always be interpreted with caution, as an indication but not a truth.

7. REFERENCES

- [1] J. Coscarelli, "An a.i. hit of fake 'drake' and 'the weeknd' rattles the music world," *The New York Times*. [Online]. Available: <https://www.nytimes.com/2023/04/19/arts/music/ai-drake-the-weeknd-fake.html>
- [2] H. H. S. Josan, "Ai and deepfake voice cloning: Innovation, copyright and artists' rights," *Artificial Intelligence*, 2024.
- [3] K. Robinson, "Ghostwriter, the mastermind behind the viral drake ai song, speaks for the first time," *billboard*. [Online]. Available: <https://www.billboard.com/music/pop/ghostwriter-heart-on-my-sleeve-drake-ai-grammy-exclusive-interview-1235434099/>
- [4] M. Defferrard, K. Benzi, P. Vandergheynst, and X. Bresson, "FMA: A dataset for music analysis," in *Proc. ISMIR*, 2017.
- [5] M. Defferrard, S. P. Mohanty, S. F. Carroll, and M. Salathé, "Learning to recognize musical genre from audio," in *Proc. Web Conf.*, 2018.
- [6] D. Bogdanov, M. Won, P. Tovstogan, A. Porter, and X. Serra, "The mtg-jamendo dataset for automatic music tagging," in *Proc. ICML Machine Learning for Music Discovery Workshop*, 2019.
- [7] Y. E. Kim and B. Whitman, "Singer identification in popular music recordings using voice coding features," in *Proc. ISMIR*, 2002.
- [8] H. Fujihara, T. Kitahara, M. Goto, K. Komatani, T. Ogata, and H. G. Okuno, "Singer identification based on accompaniment sound reduction and reliable frame selection," in *Proc. ISMIR*, 2005.
- [9] A. Mesaros, T. Virtanen, and A. Klapuri, "Singer identification in polyphonic music using vocal separation and pattern recognition methods," in *Proc. ISMIR*, 2007.
- [10] H. Fujihara, M. Goto, T. Kitahara, and H. G. Okuno, "A modeling of singing voice robust to accompaniment sounds and its application to singer identification and vocal-timbre-similarity-based music information retrieval," *IEEE Transactions on Audio, Speech, and Language Processing*, 2010.
- [11] M. Lagrange, A. Ozerov, and E. Vincent, "Robust singer identification in polyphonic music using melody enhancement and uncertainty-based learning," in *Proc. ISMIR*, 2012.
- [12] R. Hennequin, A. Khlif, F. Voituret, and M. Moussallam, "Spleeter: a fast and efficient music source separation tool with pre-trained models," *Journal of Open Source Software*, 2020.
- [13] A. Défossez, N. Usunier, L. Bottou, and F. Bach, "Music source separation in the waveform domain," *arXiv preprint arXiv:1911.13254*, 2019.
- [14] B. Sharma, R. K. Das, and H. Li, "On the importance of audio-source separation for singer identification in polyphonic music," in *Proc. Interspeech*, 2019.
- [15] T.-H. Hsieh, K.-H. Cheng, Z.-C. Fan, Y.-C. Yang, and Y.-H. Yang, "Addressing the confounds of accompaniments in singer identification," in *Proc. IEEE ICASSP*, 2020.
- [16] H. Yakura, K. Watanabe, and M. Goto, "Self-supervised contrastive learning for singing voices," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2022.
- [17] B. Torres, S. Lattner, and G. Richard, "Singer identity representation learning using self-supervised techniques," in *Proc. ISMIR*, 2023.
- [18] K. Lee and J. Nam, "Learning a joint embedding space of monophonic and mixed music signals for singing voice," *Proc. ISMIR*, 2019.
- [19] K. L. Kim, J. Lee, S. Kum, and J. Nam, "Learning a cross-domain embedding space of vocal and mixed audio with a structure-preserving triplet loss," in *Proc. ISMIR*, 2021.
- [20] Y. Xie, J. Zhou, X. Lu, Z. Jiang, Y. Yang, H. Cheng, and L. Ye, "Fsd: An initial chinese dataset for fake song detection," in *Proc. IEEE ICASSP*, 2024.
- [21] Y. Zang, Y. Zhang, M. Heydari, and Z. Duan, "Singfake: Singing voice deepfake detection," in *Proc. IEEE ICASSP*, 2024.
- [22] J. Royo-Letelier, R. Hennequin, V.-A. Tran, and M. Moussallam, "Disambiguating music artists at scale with audio metric learning," *Proc. ISMIR*, 2018.
- [23] P. Alonso-Jiménez, X. Serra, and D. Bogdanov, "Music representation learning based on editorial metadata from discogs," in *Proc. ISMIR*, 2022.

- [24] Y. Kong, V.-A. Tran, and R. Hennequin, “Strada: A singer traits dataset,” *arXiv preprint arXiv:2406.04140*, 2024.
- [25] J. Wilkins, P. Seetharaman, A. Wahl, and B. Pardo, “Vocalset: A singing voice dataset.” in *Proc. ISMIR*, 2018.
- [26] L. Zhang, R. Li, S. Wang, L. Deng, J. Liu, Y. Ren, J. He, R. Huang, J. Zhu, X. Chen *et al.*, “M4singer: A multi-style, multi-singer and musical score provided mandarin singing corpus,” *Proc. NeurIPS*, 2022.
- [27] S. Rouard, F. Massa, and A. Défossez, “Hybrid transformers for music source separation,” in *Proc. IEEE ICASSP*, 2023.
- [28] A. Défossez, “Hybrid spectrogram and waveform source separation,” in *Proc. ISMIR Workshop on Music Source Separation*, 2021.
- [29] Y. Gong, Y.-A. Chung, and J. Glass, “Ast: Audio spectrogram transformer,” in *Proc. Interspeech*, 2021.
- [30] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, “A simple framework for contrastive learning of visual representations,” in *Proc. ICML*, 2020.
- [31] J. Spijkervet and J. A. Burgoyne, “Contrastive learning of musical representations,” *Proc. ISMIR*, 2021.
- [32] A. Saeed, D. Grangier, and N. Zeghidour, “Contrastive learning of general-purpose audio representations,” in *Proc. IEEE ICASSP*, 2021.
- [33] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *Proc. ICML*, 2015.
- [34] F. Wang and H. Liu, “Understanding the behaviour of contrastive loss,” in *Proc. IEEE/CVF CVPR*, 2021.
- [35] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *Proc. ICLR*, 2015.
- [36] C. M. Bishop, “Pattern recognition and machine learning,” *Springer*, 2006.
- [37] G. Meseguer-Brocal, D. Desblancs, and R. Hennequin, “An experimental comparison of multi-view self-supervised methods for music tagging,” in *Proc. IEEE ICASSP*, 2024.
- [38] T. Bertin-Mahieux, D. P. Ellis, B. Whitman, and P. Lamere, “The million song dataset,” in *Proc. ISMIR*, 2011.

EMOTION-DRIVEN PIANO MUSIC GENERATION VIA TWO-STAGE DISENTANGLEMENT AND FUNCTIONAL REPRESENTATION

Jingyue Huang¹ Ke Chen¹ Yi-Hsuan Yang²

¹Department of Computer Science and Engineering, UC San Diego

²Department of Electrical Engineering, National Taiwan University

{jih150, knutchen}@ucsd.edu, yhyangtw@ntu.edu.tw

ABSTRACT

Managing the emotional aspect remains a challenge in automatic music generation. Prior works aim to learn various emotions at once, leading to inadequate modeling. This paper explores the disentanglement of emotions in piano performance generation through a two-stage framework. The first stage focuses on valence modeling of lead sheet, and the second stage addresses arousal modeling by introducing performance-level attributes. To further capture features that shape valence, an aspect less explored by previous approaches, we introduce a novel functional representation of symbolic music. This representation aims to capture the emotional impact of major-minor tonality, as well as the interactions among notes, chords, and key signatures. Objective and subjective experiments validate the effectiveness of our framework in both emotional valence and arousal modeling. We further leverage our framework in a novel application of emotional controls, showing a broad potential in emotion-driven music generation.

1. INTRODUCTION

With the recent advancements in symbolic music generation [1–6], there has been a growing interest in controlling high-level musical features throughout the generation process. Among these features, *emotion-driven music generation* [7–13] aims to generate music that conveys specific emotions, representing a crucial aspect for music appreciation and analysis. The downstream applications of such models have also been explored, such as music therapy for healthcare and educational purposes [14] and soundtrack generation for videos and movies [15].

Emotion could be represented in two dimensions from the literature [16]: *valence* and *arousal*. Valence refers to the positiveness of an emotion and arousal refers to energy or activation [17–19]. These two dimensions can be further divided into four quadrants (4Q), namely high valence high arousal (Q1), low valence high arousal (Q2), low valence low arousal (Q3), and high valence low arousal (Q4).

In this paper, we focus on the emotion-driven *piano performance generation* of these four quadrants. Throughout prior works, we observe crucial challenges from the perspectives of both model design and musical inductive bias.

First, previous emotion-driven piano performance generation models [7, 8] attempt to learn emotion quadrants and expressions in an *end-to-end* paradigm. In terms of model design, this approach poses training difficulty on the generation model, leading to the instability in achieving results of desired emotions. For example, many existing works [7, 9, 11] could effectively control the arousal levels of music, while their performance of *valence modeling*, especially in generating low valence (i.e., negative) music, is still poor. In terms of music, the creation process of music typically involves multiple stages, such as the *lead sheet composition* for melodies and chord progressions, and *performance generation* for textures and expressiveness. Consistently, emotion can be evoked through a combination of musical elements (e.g., melody, chord, texture). For example, major/minor chords have been found to seize different valence trends in psychological studies [20] and performance-level attributes like articulation, tempo, and velocity are more related to arousal [21, 22]. It is worth to explore the potential relation between the *disentanglement* of the generation process and the emotion expression.

Second, previous emotion-driven generation models have received limited attention regarding the influence of *tonality* on emotion modeling. It has been widely shown that major-minor tonality in composition is highly related to valence perception [22–25]. For example, as depicted in Figure 1, the histogram of musical keys derived from the emotion-labeled music dataset EMOPIA [7] supports the distribution skews to major keys for high valence clips and opposite trend for low valence ones. Furthermore, different tonalities may reveal similar patterns in the relative relationships between melodies and chords, while the distribution of melodies, chords, and tonalities can exhibit distinct shapes across different emotions. Current representations of symbolic music, such as REMI [2] and CP-Word [26], do not explicitly incorporate such interactions nor address its connection to emotion adequately. Therefore, it is necessary to consider a functional format of symbolic music representation considering the relationships between notes, chords and key signatures to better model the tonality in the emotion-driven music generation process.

In this paper, we contribute to combat above challenges:



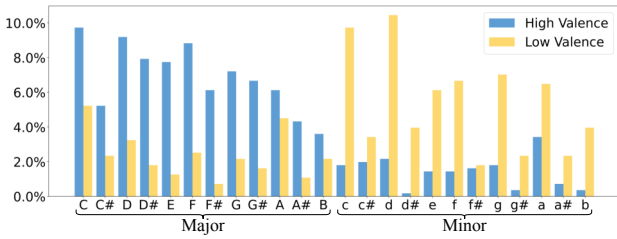


Figure 1. Key histogram of high/low valence clips from the emotion-labeled piano music dataset EMOPIA [7].

- We employ a two-stage Transformer-based model on emotion-driven piano performance generation. The first stage focuses on valence modeling via lead sheet composition, while the second stage addresses arousal modeling by introducing performance-level attributes.
- We propose a novel functional representation for symbolic music, encoding both melody and chords with Roman numerals relative to musical keys, to consider the interactions among notes, chords and tonalities [27].
- Experiments demonstrate the effectiveness of our framework and representation on emotion modeling. Additionally, our method enables new capabilities to control the arousal levels of generation under the same lead sheet, leading to more flexible emotion controls.

As a minor contribution, we also refine key signature labels and extract lead sheet annotations for the EMOPIA dataset [7] to ensure the correct training of the two-stage framework. We share the data, open source our code¹ and present generation samples in the demo page.²

2. RELATED WORK

2.1 Emotion-driven Piano Performance Generation

Prior works apply emotion conditions on deep-learning models to guide the generation of piano performance [7, 8, 11], or develop searching methods to generate music of desired emotions [28, 29]. Musical elements via feature disentanglement [9] or supervised clustering [10] can further be regarded as a bridge between emotion labels and performances for generation. In contrast, our framework employs a two-stage generation approach to reduce the complexities of one-stage generation, fostering a more nature process of music creation as well as a better incorporation between emotion labels and generation results.

2.2 Tonality, Functional Harmony, and Emotion

Musical keys and functional harmony have been explored in the field of roman numeral analysis [30–32]. The analysis of how modes and tonalities relate to mid-level perceptual features (e.g., dissonance, tonal stability, minoriness) and affect the emotional perception of music pieces has also been discovered [22, 24].

While some music generation works attempted to combine key information into data representation [33], loss

¹ <https://github.com/Yuer867/EMO-Disentangler>

² <https://emo-disentangler.github.io/>

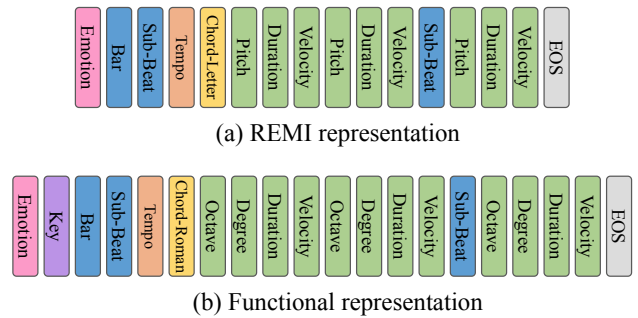


Figure 2. Illustration of (a) REMI [2], (b) the proposed functional representation, and their differences.

function [34] and text conditions [6], none of them explore the relation between musical keys and emotional perception. In this paper, we leverage both functional harmony knowledge and class-octave based pitch representation [35] to design a new data representation, incorporating the relationships between notes, chords and keys for emotion-driven music generation.

3. METHOD

In this section, we will first introduce the functional representation of symbolic music as the main generation unit. Then we introduce the two-stage model as the main component of the emotion disentanglement and generation.

3.1 Functional Representation

Figure 2 illustrates our proposed functional representation. Its design is initially based on REMI [2], a widely used event-based representation for symbolic music. We incorporate different note and chord events assisting to better learn the joint information of emotion and key signature.

3.1.1 Emotion and Key Events

We follow CTRL [36] to set up the condition within the autoregressive generation process in Transformer architecture. To denote distinct emotions and affect overall properties, we begin the event sequence with `<Emotion_*>` event to indicate the emotion label of music clips. The `<Key_*>` event is appended after `<Emotion_*>` to provide the musical key property, with the total of 24 keys (12 tonic notes with two modes in EMOPIA [7]).

3.1.2 Bar, Sub-Beat, Tempo and EOS Events

Similar to REMI, a `<Bar>` event denotes the new start of a bar; a `<Sub-Beat_*>` event denotes one of 16 possible discrete beat locations within a bar; a `<Tempo_*>` event denotes local tempo changes every four beats; and an `<EOS>` event denotes the end of sequence.

3.1.3 Chord Events

A musical chord name typically consists of root note and chord quality. For example, `Fmaj` represents the chord F–A–C with root F and major quality. Such symbols describe correct note information in chord within the tonality,

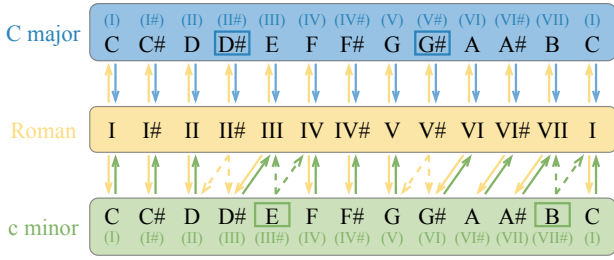


Figure 3. The conversion between letters and Roman numerals in the cases of C major and c minor scales. Solid arrows denote strict one-to-one conversions, and dotted arrows denote optional one-to-either conversions.

but they overlook the variations in *chord functions* of the same chord across different tonalities. For example, while Fmaj serves the tonic function in F major scale, it serves the subdominant function in C major scale. Moreover, the chord progression follows these functional harmony rules to establish tonality and convey musical emotion [27].

To introduce chord functions in the emotion modeling, we adopt Roman numerals from *Roman Numeral Analysis* [31] to notate chord roots in Figure 3. Given the $\langle \text{Key}_* \rangle$ event, root notes in the absolute pitch are directly converted into Roman numerals based on their scale degrees relative to the key (i.e., relative pitch). For roots outside the scale, we employ a direct conversion for I#, II#, IV#, V# and VI# appearing in major keys, but randomly assign III# and VII#, which only appear in minor keys, as one of their neighboring degrees during the encoding and decoding process. This design ensures the notation to be key-independent and make every conversion of notes reasonable to the music theory. The notations of chord qualities remain unchanged, and the chord event $\langle \text{Chord}_* \rangle$ appears every four beats.

3.1.4 Note-related Events

A note is denoted by $\langle \text{Pitch}_* \rangle$, $\langle \text{Duration}_* \rangle$ and $\langle \text{Velocity}_* \rangle$ events, where $\langle \text{Pitch}_* \rangle$ event indicates the onset of pitches from A0 to C8. Inspired by [35, 37], we decompose $\langle \text{Pitch}_* \rangle$ into $\langle \text{Octave}_* \rangle$ and $\langle \text{Degree}_* \rangle$ events according to the note octave and degree in the certain key scale. The conversion rule from $\langle \text{Pitch}_* \rangle$ to $\langle \text{Degree}_* \rangle$ is the same as that of chord roots in Figure 3. For example, pitch D#4 is decomposed into $\langle \text{Octave}_4 \rangle$ and $\langle \text{Degree}_{III} \rangle$ in c minor scale, but $\langle \text{Degree}_I \rangle$ in D# major scale. Such degree-octave pitch representation narrows the difference between melodies, thus improves the learning of connections between emotions, chords, and melodies, as demonstrated in Figure 4.

3.2 Two-stage Emotion Disentanglement

We use the idea of Compose & Embellish [38] to generate music in two stages: lead sheet first, and then piano performance. While Compose & Embellish is emotion-agnostic, we extend it so that the lead sheet model involves *valence modeling* and the performance model *arousal modeling*.

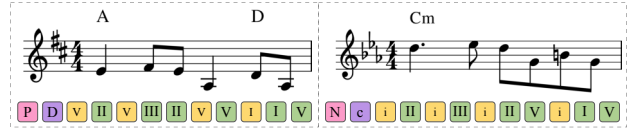


Figure 4. Two lead sheet examples from different songs in EMOPIA. In our functional representation, they have the same melody events (green), but different chord events (yellow) by different emotions (Positive and Negative by pink) and keys (D major or c minor by purple).

3.2.1 Valence Modeling

The top left section of Figure 5 denotes the first stage, where only emotion events $\langle \text{Emotion_Positive} \rangle$ and $\langle \text{Emotion_Negative} \rangle$ are considered as conditions. The former includes music pieces of Q1 and Q4 (high valence) and the latter includes those of Q2 and Q3 (low valence). The lead sheet model first predicts a key event k conditioned on the given emotion event e , and then generates the lead sheet sequence $M = \{m_1, \dots, m_T\}$ of length T , as melody and chord progression, conditioned on previous tokens step-by-step:

$$p(k, M|e) = p(k|e) \prod_{t=1}^T p(m_t|e, k, M_{<t}), \quad (1)$$

where $p(k|e)$ and $p(m_t|e, k, M_{<t})$ are jointly learned through the Transformer-based generation model [26, 38]. Performance-related events $\langle \text{Velocity}_* \rangle$ and $\langle \text{Tempo}_* \rangle$ are removed in the first stage (i.e., lead sheet generation), as we mainly focus on the contributions of key, pitch and chord for valence perception.

3.2.2 Arousal Modeling

The top right section of Figure 5 denotes the second stage. Given the lead sheet M , the performance model generates performance X conditioned on the true emotion label (Q1 to Q4). As the valence aspect has already been modeling in the first stage, this stage focuses on the generation of musical textures for the lead sheet, and more importantly, on how to perform it through variations of tempo, velocity, articulation, and other performance-level attributes that largely influence perceived arousal [21, 22]. During the training and inference phases, with the positions of $\langle \text{Bar} \rangle$ events, M and X are further segmented into $\{M_1, \dots, M_b\}$ and $\{X_1, \dots, X_b\}$, where b is the number of bars. The segmented sequences are “interleaved” in the form of $\{\dots \langle \text{Track}_M \rangle, M_i, \langle \text{Track}_X \rangle, X_i \dots\}$ with additional $\langle \text{Track}_* \rangle$ events to distinguish M and X tracks. In that, the target performance bar X_i is appended to its corresponding conditions M_i , as mapping each lead sheet segment to its corresponding performance segment [26]. With the emotion condition and key event from lead sheet as prefix tokens, the performance model is summarized as

$$p(X|e, k, M) = \prod_{i=1}^b p(X_i|e, k, M_{\leq i}, X_{<i}). \quad (2)$$

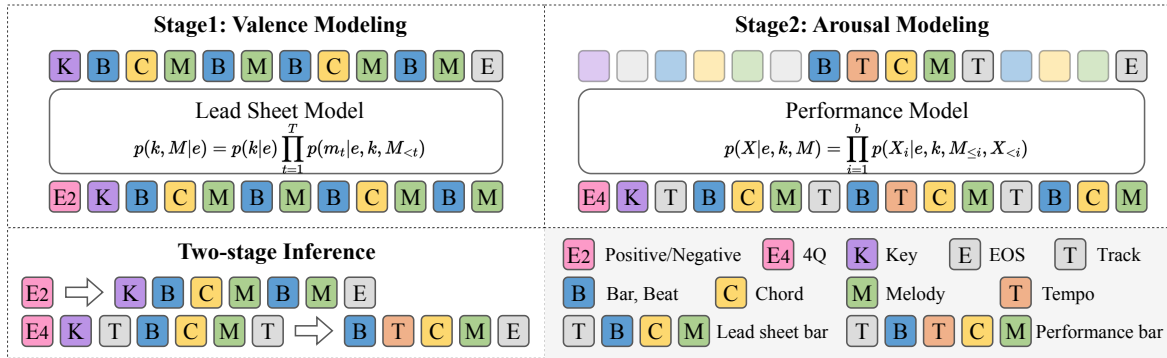


Figure 5. The two-stage framework of emotion-driven piano performance generation. Squares with transparent background denote the tokens that are not included in the loss computation during the training phase.

3.2.3 Training Objectives

Lead sheet and performance models are trained separately by both optimizing the negative log-likelihood loss of the sequence. Since existing emotion-labeled music datasets are not large, we leverage large-scale music datasets without emotion annotations to *pretrain* both models for better music understanding. During pretraining, the emotion event is marked as `<Emotion_None>`. We then finetune two models on the emotion-labeled dataset (detail in Section 4) to learn composition and performance styles specific to different emotion contexts.

3.2.4 Two-stage Inference

The left bottom section of Figure 5 denotes the inference process of both models. In the first stage, the lead sheet model predicts the key event and generates the lead sheet sequence step-by-step given `<Emotion_Positive>` or `<Emotion_Negative>` event, creating a musical motif for the specific valence preference. Even though our framework has the capability to generate any-key music of specific emotions, we observe that some generation results, such as a high-valence and high-arousal song with a minor key scale, may go beyond the current definition of emotion in [16], where the valence naturally has a strong correlation to the major-minor tonality (Figure 1). Therefore, we limit major keys to `<Emotion_Positive>` and minor keys to `<Emotion_Negative>` during the inference stage. We acknowledge that this can be overly simplifying. Since this paper focuses mainly on the valence-arousal disentanglement during the generation process, we leave this exploration of generating any emotion within any key as an advanced topic for future research.

In the second stage, the performance model generates piano performance with desired valence and arousal combination given the lead sheet from the first stage. For example, to generate a music piece of Q3, a “Negative” lead sheet and a “Q3” emotion event are selected as conditions. Additionally, this two-stage framework enables the flexibility to generate different arousal levels of piano performance under the same lead sheet, delivering some scenarios when the music need to shift quickly to complement the scenes in movies or daily videos (detail in Section 4 and the demo page).

4. EXPERIMENTS

4.1 Datasets and Preprocessing

As presented in Table 1, we collect different datasets for pretraining and finetuning phases as mentioned in Section 3.2.3. For pretraining the lead sheet model, we use the HookTheory dataset [39,40], where we choose 18,206 lead sheets with high-quality and human-transcribed melody, chord and key annotations in 4/4 time signature. We simplify 249 chord quality classes into 11 types³ as the same set in the other datasets below. For pretraining the performance model, we use the Pop1k7 dataset [26], consisting of 1747 transcribed pop piano performances. Since Pop1k7 does not contain lead sheet annotations, we refer [38] to extract melodies using the skyline algorithm [41], recognize chords using the chorder library [42], and detect key signatures using [43] in MIDI Toolbox [44].

For finetuning the models with emotion conditions, we use the EMOPIA dataset [7], consisting of 1,071 music clips with human-annotated emotion labels. Similar to Pop1k7, we obtain the lead sheets of EMOPIA by extracting melodies using the algorithm in [45] and recognizing chords using the algorithm in [46]. Empirically, we observe that specifically in the EMOPIA dataset, melodies and chords extracted by these alternative algorithms are more correct compared to the skyline algorithm and the chorder library. Additionally, we found the key signature labels in EMOPIA are not fully correct since they are also obtained by the detection algorithm with error rates. Since the valence modeling is strongly related to the musical keys and modes, we manually correct the key annotations of 367 clips in EMOPIA to ensure a high quality of lead sheets.

All datasets are randomly divided into respective training and validation sets at the ratio of 9:1. As a result in our functional representation, the vocabulary size of events is 215 for lead sheet and 324 for piano performance.

4.2 Model Settings

The lead sheet model is a 12-layer Transformer Decoder [47] with 8 heads, 512 hidden dimensions and relative positional encoding [48]. The performance model is

³ Major, minor, augment, diminish, suspend2, suspend4, major7, minor7, dominant7, diminish7, half-diminish7

Dataset	# clips (major)	# bars	# events
HookTheory [40]	18,206 (9,737)	10.84	282.81
Pop1k7 [26]	1,747 (1,264)	104.82	6794.86
EMOPIA(L) [7]	1,071 (618)	16.94	435.22
EMOPIA [7]	1,071 (618)	17.09	1311.47

Table 1. The datasets. (major) denotes the number of clips in major key (and the left is in minor key). The #bars and #events are average numbers across a dataset. EMOPIA(L) refers to EMOPIA lead sheets.

similar to the lead sheet model except with Performer attention [49]. The total parameter sizes are 41 million and 38 million respectively.

Both models are trained with the batch size of 4, the maximum sequence length of 512 (lead sheet model) or 3072 (performance model), and the Adam optimizer with $\beta = (0.99, 0.9)$. We adopt a 200-step warm-up to achieve the maximum learning rate of $1e-4$ for pretraining and $1e-5$ for finetune. All models are implemented by PyTorch and trained on one NVIDIA Tesla V100 GPU. The lead sheet model took around 180,000 steps to converge and the performance model took around 200,000 steps. Nucleus sampling [50] is employed in the inference phase. We referred [38, 51] to choose the sampling hyperparameters $\tau = 1.2$, $p = 0.97$ for the lead sheet model and $\tau = 1.1$, $p = 0.99$ for the performance model.

4.3 Baseline and Ablations

We consider the emotion-driven piano performance generation model in EMOPIA [7] as our baseline, which generates music in an end-to-end paradigm instead of two stages. To ensure the fair comparison of generation performance, we trained the baseline model under the same datasets in both pretraining and finetune phases, and replaced the original CP-Word representation with REMI as the former usually yields better generation performance and more comparable to our proposed functional representation. Two other related works [8, 9] are not included in comparison due to the main reason that we focus more on the evaluation of the two-stage framework in valence and arousal modeling; and the partial reason that they are not open-source or releasing the reproducible model weights.

We conduct a comprehensive ablation study to evaluate if each proposed design benefits the emotion modeling of music generation. Specially, these designs include: 1) the two-stage generation, 2) the functional representation, and 3) the dataset pretraining. In the following sections, models are denoted as **<representation(stage)>**. For example, REMI(one) denotes the one-stage generation model with REMI representation as the baseline, and REMI(two) denotes the two-stage generation as one variant.

4.4 Objective Evaluation and Results

Even though previous studies [7–9] employ metrics, such as Pitch Range (PR) and Number of Pitch Classes (NPC), to evaluate the generation performance, they do not pro-

	M	C	M+C	P
REMI+key (two)	0.465	0.065	0.075	0.418
–w/o. pretrain	0.350	0.105	0.130	0.343
functional (two)	0.505	0.700	0.735	0.548
–w/o. pretrain	0.400	0.570	0.625	0.430
Real data	0.578	0.695	0.746	0.812

Table 2. Key consistency calculated across all components, including melody (M), chord (C), lead sheet (M+C) and performance (P).

vide any evidences on the superiority of melody development, chord progression, and texture arrangement of music. Therefore, a model with more similar PR and NPC values to those of the target dataset does not necessarily promise a better generation quality than others.

Instead of using such metrics, we wish to evaluate the consistency between the input musical conditions and the generation results. We introduce **key consistency** to assess if a model can generate music pieces that adhere to the desired input key signatures, which is highly correlated to the lead sheet development and valence modeling. Specifically, key consistency measures the match between the key condition $\langle \text{Key}_* \rangle$ and the actual key detected in the generation using the algorithm [44] with an 81% accuracy rate. We compare REMI and our functional representation to determine if the functional representation can improve the key consistency via more close and interactive designs on key, melody, and chord. Since this metric requires the key as conditions, we add the $\langle \text{Key}_* \rangle$ event in REMI after $\langle \text{Emotion}_* \rangle$ (as REMI+key in Table 2) when training the model. The non-pretrained versions are also included for comparison. Each model generates 200 lead sheets (100 high and 100 low valence) and 400 performance samples (100 per emotion quadrant) for evaluation.

From Table 2, the functional representation outperforms REMI (i.e., REMI+key) across all components and even achieves compatible accuracy to real data over the lead sheet component. This demonstrates the effectiveness of the functional representation, by representing notes and chord roots relative to key events for key modeling. In contrast, REMI struggles with associating chord events with keys due to the ignorance of chord labels serving different functions in different key scales. Moreover, pretraining process introduces musical priors to enhance the learning of key relationships with other musical elements, improving key consistency for both representations.

4.5 Subjective Evaluation and Results

We leverage an online listening test to assess the emotion modeling ability of models. The test was conducted to collect user responses on three parts: 1) valence modeling, 2) arousal modeling, and 3) 4Q emotion modeling. During this test, the quality of the generated music has also been assessed implicitly as it is a prerequisite to the emotion expression in the music. 22 participants were engaging in this test, 5 with less than 2 years of musical training, 8 with 2-5 years, 3 with 5-10 years, and 6 with more than 10 years.

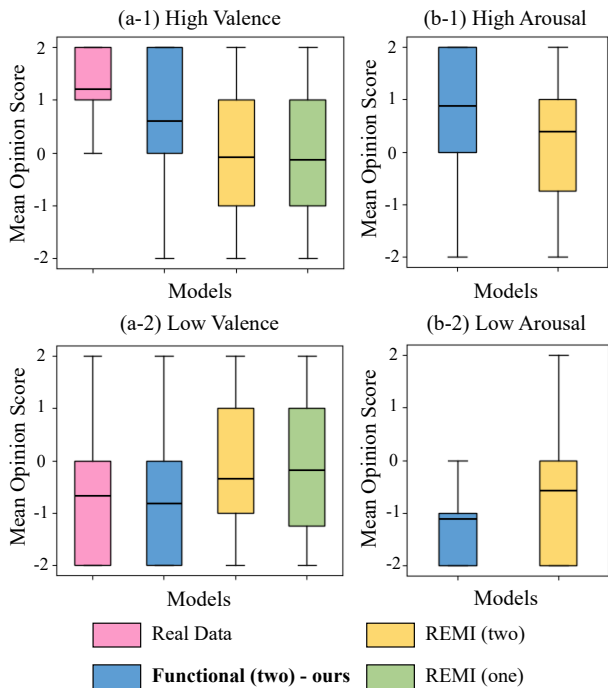


Figure 6. The mean opinion score performance on the valence-oriented and arousal-oriented listening tests. For (a-1) and (b-1), the higher score the better performance; for (a-2) and (b-2), the lower score the better performance.

4.5.1 Valence Modeling

In this part, each participant listened to 16 generated tracks of piano performance from four models [four tracks (two high valence and two low valence) per model]: 1) Real data; 2) REMI (one); 3) REMI (two); 4) Functional (two). For each track, participants rated its positiveness from -2 (low valence) to 2 (high valence) with the step size 1.

The left of Figure 6 (‘a’) presents the mean opinion scores for the valence-oriented test, where the Functional (two) model significantly outperforms both REMI (two) and REMI (one) models. The REMI (two) model shows a slight improvement over REMI (one) due to its two-stage design. Our proposed Functional (two) model even marginally exceeds real data in low valence scores, which could be due to the potential subjective biases in the negative emotion as discussed in [10]. And our model achieves both great performance in high valence and low valence results, demonstrating a good balance in valence modeling.

4.5.2 Arousal Modeling

In the second part, the functional (two) and REMI (two) models are chosen to compare their arousal modeling performance. Specifically, we wish to explore whether they can generate piano performance with either high or low arousal under the same lead sheet based on the given conditions (Q1 and Q4 for positive lead sheets, Q2 and Q3 for negative ones). Two pairs of generated tracks are randomly drawn for each model and each valence level, where every pair includes two tracks of different arousal conditions. For each track, participants rated its arousal level from -2 (low arousal) to 2 (high arousal) with the step size 1.

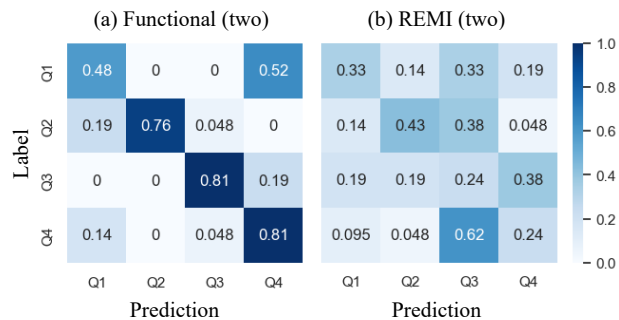


Figure 7. The confusion matrices on the 4Q listening tests.

The right of Figure 6 (‘b’) presents the results. The Functional (two) model surpasses REMI (two) by an average of 0.5 point, highlighting its superior ability to differentiate between the musical features of the two arousal levels through performance. Additionally, it is rare for Functional (two) to be incorrectly identified as high arousal under low arousal conditions (Figure 6 (b-2)).

4.5.3 4Q Emotion Judgement

In the last part, participants needed to choose the best option from four options (4Q) for each track, with 8 tracks in total for the two models the last section (4 tracks per model and 1 track per emotion).

Figure 7 presents the confusion matrices of two models. The Functional (two) model achieves the higher overall accuracy than that of REMI (two) (71.5% vs. 31.0%). When examining each emotion category, Functional(two) demonstrates superior performance in Q3 and Q4 than Q1 and Q2. Furthermore, music pieces generated from it with high valence conditions are misidentified almost based on their arousal levels; for instance, pieces intended for Q1 are almost mistaken for Q4 and vice versa. In contrast, for REMI(two), the misclassifications are across all categories, demonstrating its limitations in modeling the four emotion classes although through two-stage generation.

All above evaluations support that the combination of two-stage framework and functional representation is effective in controlling the emotion of the music it generates to a certain extent.

5. CONCLUSION AND FUTURE WORK

In this paper, we first explore emotion disentanglement through a two-stage Transformer-based framework for emotion-driven piano performance generation. Then we propose a novel functional representation for symbolic music to capture the interactions among musical keys, modes, chords, and melodies in relation to the emotion contexts. An objective metric is designed to qualify the key modeling of the proposed method, and subjective evaluations further confirm its ability to convey desired emotional perception. In the future, we wish to focus on enhancing the flexibility of emotional music generation across all musical keys and investigating new applications fostered by our framework, such as the controls of valence and arousal attributes under similar music motifs.

6. ACKNOWLEDGMENT

The work is supported by a grant from the National Science and Technology Council of Taiwan (NSTC 112-2222-E-002-005-MY2). This work also benefits from the review comments of an unpublished work [52], which is an early version of the present study. While the concept of functional representation has been mentioned in that previous work [52], its focus was on emotion-driven melody harmonization and the effectiveness of the functional representation in that context was unclear, as the melody was pre-given and fixed. In the current work, melody and chords are generated together, allowing more flexibility for emotion control with the proposed functional representation. The source code of that previous work can be found at the GitHub repo: https://github.com/Yuer867/EMO_Harmonizer.

7. REFERENCES

- [1] C.-Z. A. Huang, A. Vaswani, J. Uszkoreit, I. Simon, C. Hawthorne, N. Shazeer, A. M. Dai, M. D. Hoffman, M. Dinculescu, and D. Eck, "Music Transformer: Generating music with long-term structure," in *Proc. ICLR*, 2019.
- [2] Y.-S. Huang and Y.-H. Yang, "Pop Music Transformer: Beat-based modeling and generation of expressive pop piano compositions," in *Proc. ACM Multimed.*, 2020.
- [3] D. von Rütte, L. Biggio, Y. Kilcher, and T. Hofmann, "FIGARO: Generating symbolic music with fine-grained artistic control," in *Proc. ICLR*, 2023.
- [4] K. Chen, C. Wang, T. Berg-Kirkpatrick, and S. Dubnov, "Music sketchnet: Controllable music generation via factorized representations of pitch and rhythm," in *Proc. ISMIR*, 2020.
- [5] S.-L. Wu and Y.-H. Yang, "MuseMorphose: Full-song and fine-grained piano music style transfer with one Transformer VAE," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 31, pp. 1953–1967, 2023.
- [6] P. Lu, X. Xu, C. Kang, B. Yu, C. Xing, X. Tan, and J. Bian, "MuseCoco: Generating symbolic music from text," *CoRR*, vol. abs/2306.00110, 2023.
- [7] H. Hung, J. Ching, S. Doh, N. Kim, J. Nam, and Y.-H. Yang, "EMOPIA: A multi-modal pop piano dataset for emotion recognition and emotion-based music generation," in *Proc. ISMIR*, 2021.
- [8] P. L. T. Neves, J. Fornari, and J. B. Florindo, "Generating music with sentiment using Transformer-GANs," in *Proc. ISMIR*, 2022.
- [9] S. Ji and X. Yang, "MusER: Musical element-based regularization for generating symbolic music with emotion," in *Proc. AAAI*, 2024.
- [10] C. Kang, P. Lu, B. Yu, X. Tan, W. Ye, S. Zhang, and J. Bian, "EmoGen: Eliminating subjective bias in emotional music generation," *CoRR*, vol. abs/2307.01229, 2023.
- [11] L. Ferreira and J. Whitehead, "Learning to generate music with sentiment," in *Proc. ISMIR*, 2019.
- [12] E. Choi, Y. Chung, S. Lee, J. Jeon, T. Kwon, and J. Nam, "YM2413-MDB: A multi-instrumental FM video game music dataset with emotion annotations," in *Proc. ISMIR*, 2022.
- [13] W. Cui, P. Sarmiento, and M. Barthelet, "MoodLoopGP: Generating emotion-conditioned loop tablature music with multi-granular features," in *Proc. EvoMUSART*, 2024.
- [14] K. Zheng, R. Meng, C. Zheng, X. Li, J. Sang, J. Cai, J. Wang, and X. Wang, "EmotionBox: A music-element-driven emotional music generation system based on music psychology," *Frontiers in Psychology*, vol. 13, 2022.
- [15] M. T. Haseeb, A. Hammoudeh, and G. Xia, "GPT-4 driven cinematic music generation through text processing," in *Proc. ICASSP*, 2024.
- [16] J. A. Russell, "A circumplex model of affect," *Journal of Personality and Social Psychology*, 1980.
- [17] Y.-H. Yang, Y.-C. Lin, Y.-F. Su, and H. H. Chen, "A regression approach to music emotion recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 16, no. 2, pp. 448–457, 2008.
- [18] —, "Machine recognition of music emotion: A review," *ACM Trans. Intelligent Systems and Technology*, vol. 3, no. 3, 2012.
- [19] J. S. G. Cañón, E. Cano, T. Eerola, P. Herrera, X. Hu, Y.-H. Yang, and E. Gómez, "Music emotion recognition: Toward new, robust standards in personalized and context-sensitive applications," *IEEE Signal Process. Magazine*, vol. 38, no. 6, pp. 106–114, 2021.
- [20] D. R. Bakker and F. H. Martin, "Musical chords and emotion: major and minor triads are processed for emotion," *Cognitive, Affective, & Behavioral Neuroscience*, 2015.
- [21] Y.-C. Wu and H. H. Chen, "Generation of affective accompaniment in accordance with emotion flow," *IEEE Trans. Audio, Speech, Lang. Process.*, 2016.
- [22] S. Chowdhury and G. Widmer, "On perceived emotion in expressive piano performance: Further experimental evidence for the relevance of mid-level perceptual features," in *Proc. ISMIR*, 2021.
- [23] R. Panda, R. Malheiro, and R. P. Paiva, "Audio features for music emotion recognition: A survey," *IEEE Trans. Affective Computing*, 2020.

- [24] A. Aljanaki and M. Soleymani, “A data-driven approach to mid-level perceptual musical feature modeling,” in *Proc. ISMIR*, 2018.
- [25] Y. Hong, R. K. Mo, and A. Horner, “The effects of mode, pitch, and dynamics on valence in piano scales and chord progressions,” in *Proc. ICMC*, 2018.
- [26] W. Hsiao, J. Liu, Y. Yeh, and Y.-H. Yang, “Compound Word Transformer: Learning to compose full-song music over dynamic directed hypergraphs,” in *Proc. AAAI*, 2021.
- [27] T. Chen and L. Su, “Functional harmony recognition of symbolic music data with multi-task recurrent neural networks,” in *Proc. ISMIR*, 2018.
- [28] L. N. Ferreira, L. Mou, J. Whitehead, and L. H. S. Lelis, “Controlling perceived emotion in symbolic music generation with monte carlo tree search,” in *Proc. of AAAI (AIIDE Workshop)*, 2022.
- [29] L. N. Ferreira, L. H. S. Lelis, and J. Whitehead, “Computer-generated music for tabletop role-playing games,” in *Proc. of AAAI (AIIDE Workshop)*, 2020.
- [30] N. N. López, M. Gotham, and I. Fujinaga, “Augmentednet: A roman numeral analysis network with synthetic training examples and additional tonal tasks,” in *Proc. ISMIR*, 2021.
- [31] G. Micchi, M. Gotham, and M. Giraud, “Not all roads lead to Rome: Pitch representation and model architecture for automatic harmonic analysis,” *TISMIR*, 2020.
- [32] E. Karystinaios and G. Widmer, “Roman numeral analysis with graph neural networks: Onset-wise predictions from note-wise features,” in *Proc. ISMIR*, 2023.
- [33] B. L. Sturm, J. F. Santos, O. Ben-Tal, and I. Korshunova, “Music transcription modelling and composition using deep learning,” *CoRR*, vol. abs/1604.08723, 2016.
- [34] Y. Yeh, W. Hsiao, S. Fukayama, T. Kitahara, B. Genchel, H. Liu, H. Dong, Y. Chen, T. Leong, and Y.-H. Yang, “Automatic melody harmonization with triad chords: A comparative study,” *CoRR*, vol. abs/2001.02360, 2020.
- [35] Y. Li, S. Li, and G. Fazekas, “An comparative analysis of different pitch and metrical grid encoding methods in the task of sequential music generation,” *CoRR*, vol. abs/2301.13383, 2023.
- [36] N. S. Keskar, B. McCann, L. R. Varshney, C. Xiong, and R. Socher, “CTRL: A conditional transformer language model for controllable generation,” *CoRR*, vol. abs/1909.05858, 2019.
- [37] M. Mongeau and D. Sankoff, “Comparison of musical sequences,” *Computers and the Humanities*, vol. 24, no. 3, pp. 161–175, 1990.
- [38] S.-L. Wu and Y.-H. Yang, “Compose & Embellish: Well-structured piano performance generation via a two-stage approach,” in *Proc. ICASSP*, 2023.
- [39] “HookTheory,” <https://www.hooktheory.com/> [Accessed: (September 1, 2023)].
- [40] C. Donahue, J. Thickstun, and P. Liang, “Melody transcription via generative pre-training,” in *Proc. ISMIR*, 2022.
- [41] A. L. Uitdenbogerd and J. Zobel, “Manipulation of music for melody matching,” in *Proc. ACM Multimed.*, 1998.
- [42] J. Chang, “Chorders,” <https://github.com/joshuachang2311/chorder>.
- [43] C. L. Krumhansl, “Cognitive foundations of musical pitch,” *Oxford University Press*, 2001.
- [44] P. Toiviainen and T. Eerola, “MIDI toolbox 1.1,” <https://github.com/miditoolbox/>, 2016.
- [45] “Midi_Toolkit,” https://github.com/RetroCirce/Midi_Toolkit [Accessed: (September 1, 2023)].
- [46] Z. Wang, K. Chen, J. Jiang, Y. Zhang, M. Xu, S. Dai, and G. Xia, “POP909: A pop-song dataset for music arrangement generation,” in *Proc. ISMIR*, 2020.
- [47] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Proc. NeurIPS*, 2017.
- [48] Z. Dai, Z. Yang, Y. Yang, J. G. Carbonell, Q. V. Le, and R. Salakhutdinov, “Transformer-XL: Attentive language models beyond a fixed-length context,” in *Proc. ACL*, 2019.
- [49] K. M. Choromanski, V. Likhoshesterov, D. Dohan, X. Song, A. Gane, T. Sarlós, P. Hawkins, J. Q. Davis, A. Mohiuddin, L. Kaiser, D. B. Belanger, L. J. Colwell, and A. Weller, “Rethinking attention with Performers,” in *Proc. ICLR*, 2021.
- [50] A. Holtzman, J. Buys, M. Forbes, and Y. Choi, “The curious case of neural text degeneration,” in *Proc. ICLR*, 2019.
- [51] Z. Fu, W. Lam, A. M. So, and B. Shi, “A theoretical analysis of the repetition problem in text generation,” in *Proc. AAAI*, 2021.
- [52] J. Huang and Y.-H. Yang, “Emotion-driven melody harmonization via melodic variation and functional representation,” *CoRR*, vol. abs/2407.20176, 2024.

EFFICIENT ADAPTER TUNING FOR JOINT SINGING VOICE BEAT AND DOWNBEAT TRACKING WITH SELF-SUPERVISED LEARNING FEATURES

Jiajun Deng^{1,2}, Yaolong Ju¹, Jing Yang¹, Simon Lui¹, Xunying Liu²

¹ Huawei Technologies Co., Ltd., Shenzhen, China

² The Chinese University of Hong Kong, Hong Kong SAR, China

jjdeng@se.cuhk.edu.hk, yaolongju@huawei.com

ABSTRACT

Singing voice beat tracking is a challenging task, due to the lack of musical accompaniment that often contains robust rhythmic and harmonic patterns, something most existing beat tracking systems utilize and can be essential for estimating beats. In this paper, a novel temporal convolutional network-based beat-tracking approach featuring self-supervised learning (SSL) representations and adapter tuning is proposed to track the beat and downbeat of singing voices jointly. The SSL DistilHuBERT representations are utilized to capture the semantic information of singing voices and are further fused with the generic spectral features to facilitate beat estimation. Sources of variabilities that are particularly prominent with the non-homogeneous singing voice data are reduced by the efficient adapter tuning. Extensive experiments show that feature fusion and adapter tuning improve the performance individually, and the combination of both leads to significantly better performances than the un-adapted baseline system, with up to 31.6% and 42.4% absolute F1-score improvements on beat and downbeat tracking, respectively.

1. INTRODUCTION

Singing voice beat tracking is an important music information retrieval (MIR) task that can serve many downstream applications. For example, singing transcription can utilize beats to finetune the onsets of the transcribed notes for better accuracies [1] as well as automatic accompaniment generation, where the beat information can be instrumental for drum arrangements [2]. However, existing literature on beat tracking mostly focused on music with instrumental accompaniment [3–11], and tracking beats of singing voice is largely unaddressed and remains a key challenge to date. Its difficulty can be attributed to the lack of musical accompaniment that contains rhythmic and harmonic

patterns vital for beat tracking in general. This leads to several challenges in developing effective singing voice beat tracking systems.

First, the existing state-of-the-art music beat tracking systems deliver poor performances on singing voices due to the notable inherent disparities between complete music songs and singing voices [12]. For example, the traditional music beat tracking system often learns latent mapping based on acoustic clues such as the spectrogram magnitude [13–15], which is often caused by the reoccurring drums or bass. Such clues, however, are barely present in singing voices. Inspired by the similarity between the singing voice and speech [16], the self-supervised learning (SSL) speech representations are utilized and demonstrate advantages over spectral features in singing voices [12].

Second, the naturalistic singing voice data is generally highly non-homogeneous due to its inherent variabilities from different conditions, such as genres, singers, recording devices, or languages [17]. The resulting high degree of singing voice heterogeneity may cause a large mismatch between training and test distributions, which can significantly degrade system performances. This issue is particularly prominent with the singing voice beat tracking that lacks musical accompaniment, as opposed to music beat tracking containing rich percussive and harmonic profiles.

To this end, we present a novel singing voice beat and downbeat tracking system using a temporal convolutional network featuring SSL representations and adapter tuning. More specifically, the SSL DistilHuBERT representations are utilized to capture the essential para-linguistics, semantic, and phonemic level characteristics and are further fused with the generic spectral features to facilitate beat estimations. A series of parameter-efficient adapters are performed to compensate for mismatch arising from the inherent variabilities among diverse singing voice datasets. The main contributions of the paper are summarized below:

1) To our knowledge, this paper is the first to investigate the joint beat and downbeat tracking task featuring the fusion of SSL representations and spectral features. In contrast, similar prior research was conducted in the context of only beat estimations [12] or beat/downbeat tracking using spectral features only [18].

2) Extensive experiments show that the train-test



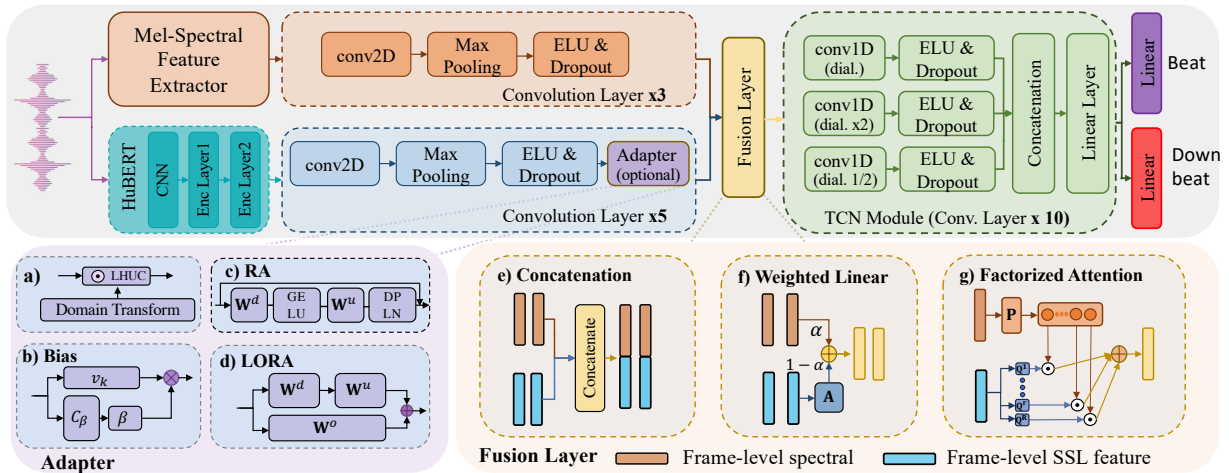


Figure 1. Examples of joint beat and downbeat tracking systems using Temporal Convolutional Network (TCN) shown in the light grey box (top). The pre-trained self-supervised DistilHuBERT model is shown in the deep cyan-blue box (top left). The parameter-efficient adapters described in Section 4 are shown in the light purple colored box (bottom left corner), which includes **a)** Learning Hidden Unit Contributions adaptation, **b)** Bias Adapter adaptation, **c)** Residual Adapter (RA) adaptation and **d)** Low-Rank Adaptation (LoRA). The fusion layer described in Section 3 is shown in the light orange box (bottom right), which includes **e)** Concatenation, **f)** Weighted linear interpolation fusion, and **g)** Factorized attention fusion.

data distribution mismatch issue presented in the non-homogeneous singing voice data significantly degrades the beat-tracking performance, particularly in downbeat estimation. To this end, inspired by the use of parameter-efficient adaptation techniques in machine learning fields [19–24], this paper presents the first work that successfully employs efficient adapter tuning approaches for singing voice beat-tracking tasks to address the mismatch above.

3) The efficacy of the proposed beat tracking approach is consistently demonstrated across various public datasets over the un-adapted baseline beat tracking system. In addition, the inherent generality of the proposed approach and the accompanying implementation details outlined in this paper allow their further application to other beat-tracking systems or MIR tasks.

2. TCN-BASED BEAT TRACKING SYSTEMS

In this paper, we adopt temporal convolutional network (TCN) as the backbone for singing voice beat tracking for two reasons: **1)** TCN has shown solid performances in the traditional beat tracking involving musical accompaniment. First proposed by [25], TCN achieved superior performances to the previous SOTA bi-directional LSTM and has been widely used for beat tracking since then [26, 27]. **2)** Although SpecTNT has recently outperformed TCN [9], TCN is still lightweight with way fewer parameters than SpecTNT, making it easy for deployment and cost-efficient as commercial applications.

2.1 Architecture

The conventional TCN-based beat tracking system consists of a front-end convolution module and a TCN module, connected by a fusion layer shown in Fig. 1 (light grey box, top). Each convolution layer in the front-end module has 20 channels, a stride of one, and kernels with various sizes. Max-pooling, exponential linear unit (ELU) acti-

vation [28], and dropout neural operations are applied to each convolution layer in sequence. The TCN module is stacked by ten dilated convolutional layers with exponentially increased dilation factors $2^0, 2^1, \dots, 2^9$ resulting in a large receptive to capture long temporal contexts. Each dilated convolutional layer contains three dilated convolution blocks, each with different dilation rates (one dilation factor, half the dilation factor, and twice the dilation factor) and 20 channels. ELU activation and dropout operations are applied to each dilated convolution block, followed by an output linear layer shown in Fig. 1 (green box, top right).

2.2 Multi-task Learning

Based on the above TCN-based architecture, the beat-tracking task can be cast as a binary classification through time, for example, classifying the presence or absence of a beat for each frame. To perform the joint beat and downbeat tracking in a single system [26, 27], an auxiliary downbeat tracking task by introducing a separate binary classification linear layer is carried out to produce the downbeat. Thus a multi-task criterion that interpolates between the beat and downbeat binary cross entropy (BCE) costs is adopted for training, which can be formulated as

$$\mathcal{L}_{MTL} = \eta \mathcal{L}_{BEAT} + (1 - \eta) \mathcal{L}_{DBEAT}, \quad (1)$$

where $\eta \in [0, 1]$ is the tunable hyper-parameter for balancing the beat BCE cost \mathcal{L}_{BEAT} and downbeat BCE cost \mathcal{L}_{DBEAT} . Both beat and downbeat prediction outputs are post-processed with a dynamic Bayesian network (DBN) [29–31] to produce the final sequence of predictions [15].

2.3 Input Features

Two types of feature embeddings are fed into the TCN-based beat-tracking system. The first is the traditional 81-dim log-magnitude mel-frequency spectrogram features,

which are widely adopted in the music beat tracking tasks [9, 27, 32]. Another is the SSL DistilHuBERT features. DistilHuBERT [33, 34] is a lightweight self-supervised pre-trained speech foundation model. Its lighter architecture enables faster inference than other pre-trained foundation models. The 768-dim DistilHuBERT feature representations extracted from the last HuBERT layer are proven to serve as beneficial feature embeddings in analyzing singing rhythms [12] due to the acoustic and linguistic similarities between singing voices and speech.

3. FEATURE FUSION

The process of combining diverse feature representations, named feature fusion, plays a vital role in determining the effectiveness of beat-tracking systems [35]. To this end, several fusion approaches are introduced in this section to integrate the traditional spectrogram and pre-trained HuBERT feature representations effectively.

3.1 Early Feature Fusion

Early feature fusion is the combination of diverse feature representations performed early in a neural network [36]. For example, the features are fused at the network's input layer. Let $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T] \in \mathbb{R}^{m \times T}$ and $\mathbf{u} = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_T] \in \mathbb{R}^{n \times T}$ denote the spectrogram and HuBERT feature representations with T frames, respectively. Two forms of early feature fusion are investigated.

a) Input concatenation refers to directly concatenating the spectrogram and HuBERT features at the frame level. The concatenated feature representation \mathbf{z} at t -th frame can be expressed as $\mathbf{z}_t = [\mathbf{x}_t; \mathbf{u}_t] \in \mathbb{R}^{n+m}$.

b) Weighted linear interpolation refers to interpolating the frame-level spectrogram and HuBERT feature representations with a learnable hyper-parameter $\alpha \in [0, 1]$. The interpolated feature representation at t -th frame can be formulated as $\mathbf{z}_t = \alpha \mathbf{x}_t + (1 - \alpha) \mathbf{A} \mathbf{u}_t$, where $\mathbf{A} \in \mathbb{R}^{n \times m}$ is a learnable projection matrix to enable the dimension of HuBERT features to be consistent with that of the spectrogram features.

3.2 Late Feature Fusion

The combination of diverse features at a later network layer leads to late feature fusion [37]. This allows the model to leverage high-level, abstract representations, leading to more informed decisions and improved performance. As shown in Fig. 1, the spectrogram and pre-trained HuBERT features are first fed into a separate CNN module before being further combined using different fusion schemes. Let $\hat{\mathbf{x}} = [\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2, \dots, \hat{\mathbf{x}}_T] \in \mathbb{R}^{k \times T}$ and $\hat{\mathbf{u}} = [\hat{\mathbf{u}}_1, \hat{\mathbf{u}}_2, \dots, \hat{\mathbf{u}}_T] \in \mathbb{R}^{k \times T}$ denote the high-level CNN output hidden representations with T frames using the spectrogram and HuBERT features, respectively. The concatenation and weighted combination operations can also be performed in a late fusion style, which is illustrated as **a)** late concatenation fusion $\hat{\mathbf{z}}_t = [\hat{\mathbf{x}}_t; \hat{\mathbf{u}}_t] \in \mathbb{R}^{2k}$ and **b)** late weighted linear interpolation fusion $\hat{\mathbf{z}}_t = \alpha \hat{\mathbf{x}}_t + (1 - \alpha) \hat{\mathbf{u}}_t$.

3.3 Factorized Attention Fusion

In order to focus on relevant selective representations while suppressing less important ones, factorized attention fusion is performed in a late fusion fashion. The SSL hidden representation at t -th frame $\hat{\mathbf{u}}_t$ are first factorized into R subspace representations $[\mathbf{v}_t^1, \mathbf{v}_t^2, \dots, \mathbf{v}_t^R] \in \mathbb{R}^{k \times R}$ using a series of parallel linear transforms, which is expressed as

$$[\mathbf{v}_t^1, \mathbf{v}_t^2, \dots, \mathbf{v}_t^R] = [\mathbf{Q}^1, \mathbf{Q}^2, \dots, \mathbf{Q}^R] \hat{\mathbf{u}}_t, \quad (2)$$

where $\mathbf{Q}^r \in \mathbb{R}^{k \times k}$ is the linear transformations for r -th subspace. The spectrogram hidden embedding at t -th frame $\hat{\mathbf{x}}_t$ is projected into a R -dim interpolation vector $\mathbf{w}_t = [w_t^1, w_t^2, \dots, w_t^R] \in \mathbb{R}^R$ using a projection matrix $\mathbf{P} \in \mathbb{R}^{R \times k}$, which is given as $\mathbf{w}_t = \text{Softmax}(\mathbf{P} \hat{\mathbf{x}}_t)$. Subsequently, the fused feature representation can be obtained by an attention mechanism [38],

$$\hat{\mathbf{z}}_t = \text{Sigmoid}\left(\sum_{r=1}^R w_t^r \mathbf{v}_t^r\right). \quad (3)$$

4. PARAMETER EFFICIENT ADAPTATION

A straightforward solution to reduce the mismatch between training and evaluation distributions is to directly fine-tune the entire system using the target-domain singing voice data. However, this adaptation scheme not only encounters overfitting problems due to the scarcity of singing voices but also poses key challenges to adaptation parameter storage. Parameter-efficient adaptation approaches [39–41] that introduce limited adaptation parameters with the original model parameters unchanged have been proposed to tail for the above overfitting and parameter storage issues. Inspired by this idea, several prominent parameter-efficient adapters are explored for singing voice beat-tracking systems in this section.

4.1 Learning Hidden Unit Contributions

Learning hidden unit contributions (LHUC) adaptation is an effective speaker adaptation solution that accounts for speaker variation of speech [42]. The basic idea of LHUC adaptation is to modify the amplitudes of activation outputs using a scaling vector. Let $\mathbf{r}_{l,e} \in \mathbb{R}^u$ denote the adaptation parameters for the e -th domain in the l -th hidden layer, where u is the dimension of adaptation parameters. The adapted hidden output $\mathbf{h}_{l,k}$ can be given as

$$\mathbf{h}_{l,k} = \mathbf{h}_l \odot \xi(\mathbf{r}_{l,k}), \quad (4)$$

where \mathbf{h}_l is the original hidden activation output at the l -th hidden layer, \odot is the Hadamard product operation, $\xi(\mathbf{r}_{l,e})$ is the scaling vector parameterized by $\mathbf{r}_{l,e}$, and $\xi(\cdot)$ is the element-wise $2 \times \text{Sigmoid}(\cdot)$ function with a range of $(0, 2)$. During adaptation, the adaptation parameters $\mathbf{r}_{l,k}$ for each domain are initialized as zeros vector. An example of LHUC adaptation is shown in Fig. 1(a).

4.2 Bias Adaptation

The bias adapter adaptation [43] adds frame-level bias to the hidden representation shifts using a domain-dependent shift vector $\mathbf{v}_e \in \mathbb{R}^u$ and a linear layer C_β , which is shown in Fig. 1(b). The frames crucial for beat tracking should be assigned a larger representation shift compared to other frames. The linear layer produces a frame-level weight vector $\boldsymbol{\beta} = C_\beta \mathbf{h}_l = [\beta_1, \beta_2, \dots, \beta_T] \in \mathbb{R}^T$, where β_t denotes the weight of the t -th frame hidden representation. Therefore, the domain-dependent representation shifts \mathbf{v}_e can be enhanced by applying frame-level weights, and the adapted hidden layer output can be expressed as

$$\mathbf{h}_{l,e} = \mathbf{h}_l + \mathbf{v}_e \otimes \boldsymbol{\beta}, \quad (5)$$

where \otimes is the outer product operation, and the outer product of shift-vector \mathbf{v} and the weight $\boldsymbol{\beta}$ can be expressed as $\mathbf{v} \otimes \boldsymbol{\beta} = [\beta_1 \mathbf{v}_e, \beta_2 \mathbf{v}_e, \dots, \beta_T \mathbf{v}_e] \in \mathbb{R}^{u \times T}$.

4.3 Residual Adapter

Inspired by the residual idea [44], a residual adapter (RA) with slight modifications is designed for beat tracking. The adapter starts with a down-linear projection $\mathbf{W}_e^d \in \mathbb{R}^{r \times u}$, followed by a non-linear GeLU activation function $\zeta(\cdot)$, and an up-linear projection $\mathbf{W}_e^u \in \mathbb{R}^{u \times r}$. Let $f_{RA}(\cdot; \boldsymbol{\Theta}_{l,e})$ denote the residual adapter function for e -th domain in the l -th hidden layer, where $\boldsymbol{\Theta}_{l,e}$ is the adaptation parameters for e -th domain. The adapted hidden outputs are given as

$$\mathbf{h}_{l,k} = \mathbf{h}_l + f_{RA}(\mathbf{h}_l; \boldsymbol{\Theta}_{l,e}), \quad (6)$$

$$f_{RA}(\mathbf{h}_l; \boldsymbol{\Theta}_{l,e}) = \text{LN}(\text{DP}(\mathbf{W}_{l,e}^u \zeta(\mathbf{W}_{l,e}^d \mathbf{h}_l))), \quad (7)$$

where $\text{DP}(\cdot)$ and $\text{LN}(\cdot)$ denote the dropout and layernorm operations, respectively. The adaptation capacity can be controlled by managing the number of parameters in each adapter module through controlling the bottleneck dimension r . An example of an RA adapter is shown in Fig. 1(c).

4.4 Low-rank Adaptation

Instead of the non-parallel nature of adapter modules that consumes additional GPU time mentioned above, Low-rank adaptation (LoRA) [45] reduces the number of adaptation parameters by learning rank-decomposition matrix pairs $\{\mathbf{W}^d, \mathbf{W}^u\}$ while freezing the original weights. The LoRA-adapted linear hidden output can be expressed as

$$\mathbf{h}_{l,k} = f_{LoRA}(\mathbf{h}_{l-1}; \boldsymbol{\Theta}_{l,e}), \quad (8)$$

$$= (\mathbf{W}_l^o + \mathbf{W}_{l,e}^u \mathbf{W}_{l,e}^d) \mathbf{h}_{l-1}, \quad (9)$$

where $f_{LoRA}(\cdot; \boldsymbol{\Theta}_{l,e})$ is the LoRA adapter, $\mathbf{W}_l^o \in \mathbb{R}^{n \times u}$ is the original pre-trained weight matrix, $\mathbf{W}_{l,e}^d \in \mathbb{R}^{r \times u}$ and $\mathbf{W}_{l,e}^u \in \mathbb{R}^{n \times r}$ are the trainable low-rank decomposition matrices. It is noted that the rank $r \ll \min(u, n)$ is far less than the dimension of the original matrix, which allows for reducing the number of adaptation parameters. An example of a LoRA adapter is shown in Fig. 1(d).

Table 1. Description of the singing voice beat tracking datasets. † and * represent the music beat tracking dataset and the music source separation dataset, respectively.

Dataset	# Hours	# Excerpts	Genres
GTZAN† [48]	5.9	754	Blues, Country, Disco, Hiphop, etc.
RWC Pop† [49]	5.4	273	Japanese Pop., etc.
Ballroom† [50]	2.8	313	Rumba, Tango, Waltz, Jive, etc.
Hainsworth† [51]	1.9	173	Jazz, Metal, Rock, Opera, etc.
MUSDB18* [52]	6.4	144	Pop., Country, Rock, etc.
URSinger* [53]	3.4	142	Chinese Pop., etc.

4.5 Estimation of Adaptation Parameters

Let $\mathcal{D}_e = \{\mathbf{X}_e, \mathbf{Y}_e\}$ denote the adaptation data for e -th domain, where \mathbf{X}_e and \mathbf{Y}_e stand for the singing voice waveform and the corresponding beat/downbeat sequences, respectively. Without loss of generality and for simplicity, let $\boldsymbol{\Theta}$ denote the original model parameters. In the context of adaptation, the adaptation parameters $\boldsymbol{\Theta}_e$ conditioned on the e -th domain can be estimated by minimizing the loss function in Eqn. (1), which is given by

$$\hat{\boldsymbol{\Theta}}_e = \arg \min_{\boldsymbol{\Theta}_e} \{\mathcal{L}_{MTL}(\mathcal{D}_e; \boldsymbol{\Theta}, \boldsymbol{\Theta}_e)\}. \quad (10)$$

5. EXPERIMENTS

5.1 Datasets and Evaluation Metrics

To the best of our knowledge, there are no publicly available datasets that include pristine vocal audio alongside beats and downbeats annotations. Annotating beats and downbeats based solely on vocal signals can be arduous and subjective, even by human experts, since there are no evident rhythmic cues like percussive instruments to accurately comprehend the singer’s rhythmic intentions.

Therefore, we follow the strategy described in [12] to utilize the existing public MIR datasets and systems to create the singing voice data with beat/downbeat annotations. This includes **a) four music beat tracking datasets** with available beat annotations, where the singing signals are extracted by the Demucs source separation model [46], and **b) two music source separation datasets** with available isolated singing tracks, where the preliminary beats and downbeats annotations are generated by the existing TCN-based beat tracking system [25] using the full music mix (singing with musical accompaniment), then manual revision is further performed to correct the potential annotation errors. Altogether, six datasets are used in this paper as shown in Table 1, where a silence-stripping technique is applied to each dataset to remove the long chunks of silence. The 90% of the whole data randomly selected from a uniform distribution is used for training, while the remaining 10% is used for evaluation. The evaluation metric of F1-score with a tolerance window of ± 70 ms, a typical setting commonly used in the traditional beat tracking [25], is adopted for our performance evaluation. We also adopt P-score, Cemgil, and Goto [47] as additional evaluation metrics to further demonstrate the advantages of the proposed approaches in the final experiments (Table 4).

Table 2. Beat and downbeat tracking performance of TCN systems using different feature fusion methods evaluated on the GTZAN, RWC pop (RWCPO), and MUSDB18 (MUSDB) datasets in terms of F1-score.

ID	Input Features	Fusion Methods	GTZAN	RWCPO	MUSDB
Beat/Downbeat Tracking F1 Scores					
1	Spectrogram	-	0.48/0.26	0.65/0.53	0.31/0.15
2	DistilHuBERT	-	0.74/0.47	0.76/0.68	0.38/0.17
3	Spectrogram & DistilHuBERT	Input Concatenation	0.78/0.56	0.83/0.79	0.41/0.25
4		Input Weighted	0.77/0.55	0.86/0.81	0.43/0.25
5		Late Concatenation	0.81/0.53	0.87/0.81	0.45/0.25
6		Late Weighted	0.79/0.58	0.88/0.84	0.47/0.26
7		Factorized Attention	0.80/0.51	0.91/0.82	0.48/0.23

5.2 Implementation Details

Two feature extractors, including the mel-spectrogram feature extractor and the pre-trained SSL DistilHuBERT feature extractor [34], are employed to generate the 81-dimensional spectral features and 768-dimensional SSL feature representations of vocal signals. In this paper, the vocal signals of all datasets are resampled to 16000 Hz. As illustrated in Section 2, the temporal convolution network consists of a front-end convolution and TCN modules. The front-end convolution module tailored for late feature fusion consists of a 3-layer convolution network¹ and a 5-layer convolution network² for processing the spectral and SSL feature representations, respectively. The kernel size and stride for the Max-pooling operation are 1×3 for all convolution layers. The TCN module consists of ten dilated convolution layers, wherein the dilation factors increase exponentially.

During the TCN-based beat tracking system training, all weights of the system are randomly initialized. The Ranger optimizer [54] with an initial learning rate of 0.001, the ReduceLRonPlateau scheduler with a factor of 0.9 and patience of 5, and a dropout rate of 0.1 are used for training and adaptation. The training and adaptation epochs are set as 100 and 30, respectively. The hyper-parameter of multi-task learning in Eqn. (1) is empirically set as $\eta = 0.2$. Since the ratio of positive and negative examples in the beat-tracking task is often imbalanced, the weighted binary cross-entropy loss is applied, and the weights of positive examples for the beat and downbeat costs are set to be 10 and 20, respectively.

5.3 Performance of Feature Fusion

Table 2 shows the beat and downbeat tracking F1 scores of TCN systems using different feature representation fusion methods. Several important findings can be observed. **a)** The systems using SSL DistilHuBERT features (ID.2) show better F1 performance than those using the traditional spectral features (ID.1) in all three evaluation sets. This demonstrates that the semantic information captured by SSL speech representation is crucial for singing voice beat

¹ The channel, kernel size, stride, and padding of Conv2D used in 3-layer convolution network for each convolution layer are {20, 20, 20}, {3x3, 1x12, 1x3}, {1, 1, 1} and {1x0, 0x0, 1x0} respectively.

² The channel, kernel size, stride, and padding of Conv2D used in 5-layer convolution network for each convolution layer are {20, 20, 20, 20, 20}, {3x3, 1x12, 1x3, 3x3, 1x12, 3x3}, {1, 1, 1, 1, 1} and {1x0, 0x0, 1x0, 0x0, 1x0} respectively.

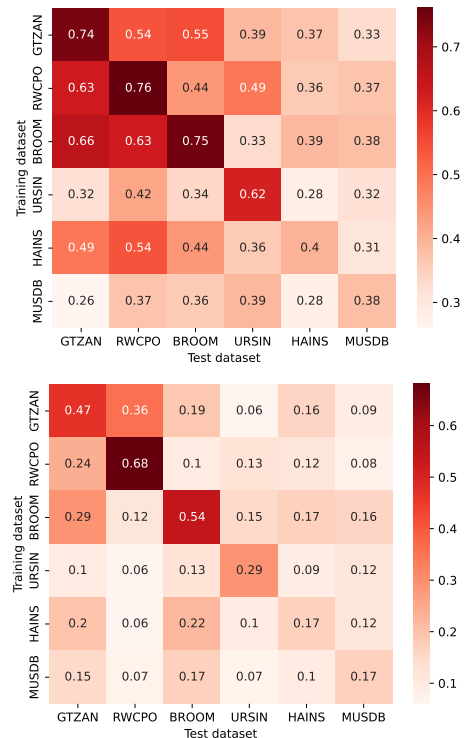


Figure 2. F1-score of the beat (upper) and downbeat (bottom) tracking when training data and test data are collected from the same (diagonal value) or different sources.

tracking. **b)** All systems that leverage feature fusion approaches (ID.3-7) outperform the systems using only one single spectral (ID.1) or SSL feature (ID.2). This confirms our motivation that spectral and SSL features are complementary as they capture different characteristics of vocal signals. **c)** The late weighted linear interpolation fusion method (ID.6) achieves the best F1 results on the downbeat tracking and competitive beat tracking performance among all fusion approaches, therefore we selected it for the following experiments.

5.4 Performance of Adaptation

The mismatch across different datasets is revealed in Fig. 2. **a)** When the system is trained on singing voice data from the same source as the test data, the best beat and downbeat tracking performance are obtained (diagonal value). **b)** The mismatch between training and test distributions (non-diagonal) significantly degrades the beat tracking performance, especially in downbeat tracking. This confirms our assumption that the mismatch across different datasets is particularly prominent in singing voice beat tracking due to the lack of robust rhythmic and harmonic patterns. Therefore, mismatch is an essential issue that needs to be addressed in multi-condition training.

Table 3 shows the beat and downbeat tracking performance of the TCN systems configured with different adapters using only DistilHuBERT features. Several trends can be found. **a)** It is not surprising that multi-condition systems (ID.2) trained on all six datasets do not always outperform the in-domain systems (ID.1) trained on the well-controlled data from the same source as test data because of the mismatch issue. This demonstrates that

Table 3. Beat and downbeat tracking performance of TCN systems configured with different adapters on the GTZAN, RWCPO, Ballroom (BROOM), Hainsworth (HAINS), MUSDB, and URSing (URSIN) datasets in terms of F1-score.

ID	Systems	Adapter			Datasets					
		Method	Location	# Params	GTZAN	RWCPO	BROOM	HAINS	MUSDB	URSIN
Beat/Downbeat Tracking F1-scores										
1	In-domain	-	-	-	0.74/0.47	0.76/0.68	0.75/0.54	0.40/0.17	0.38/0.17	0.62/0.29
2	Multi-condition	-	-	-	0.78/0.57	0.89/0.45	0.72/0.43	0.50/0.33	0.53/0.25	0.61/0.22
3	Multi-condition with Adaptation	Fine-tune	ALL Layers	100%	0.80/0.59	0.91/0.82	0.80/0.57	0.54/0.38	0.55/0.38	0.72/0.34
4		LHUC	First CNN Layer	5%	0.78/0.56	0.88/0.59	0.72/0.46	0.50/0.33	0.53/0.26	0.62/0.25
5		BIAS	First CNN Layer	10%	0.79/0.57	0.89/0.61	0.71/0.47	0.52/0.35	0.55/0.31	0.62/0.24
6		LoRA	First CNN Layer	20%	0.80/0.62	0.90/0.68	0.78/0.56	0.54/0.37	0.56/0.38	0.66/0.33
7		RA	First CNN Layer	20%	0.80/0.65	0.91/0.80	0.80/0.57	0.58/0.43	0.58/0.41	0.68/0.38
8		RA	Second CNN Layer	10%	0.78/0.60	0.90/0.77	0.76/0.56	0.57/0.41	0.55/0.40	0.68/0.35
9		RA	Third CNN Layer	5%	0.79/0.62	0.90/0.78	0.76/0.56	0.57/0.39	0.55/0.40	0.68/0.33

Table 4. Beat and downbeat tracking performance of the proposed TCN systems incorporating fusion and adapters.

ID	Systems	Input Features	Feature Fusion	Adapter	Beat Tracking				Downbeat Tracking			
					F1	P-score	Cemgil	Goto	F1	P-score	Cemgil	Goto
1	Multi-condition	Spectrogram	✗	✗	0.497	0.541	0.410	0.429	0.254	0.420	0.212	0.256
2	Multi-condition	DistilHuBERT	✗	✗	0.656	0.681	0.565	0.561	0.389	0.501	0.351	0.370
3	Proposed	Spec. & HuBERT	✓	✗	0.774	0.756	0.684	0.703	0.524	0.603	0.487	0.520
4	Proposed	Spec. & HuBERT	✓	✓	0.813	0.801	0.713	0.757	0.678	0.692	0.621	0.663

blindly expanding the training data is insufficient to enhance the system’s generalization. **b)** All systems configured with adapters (ID.4-7) improve the performance over both multi-condition systems (ID.2) and in-domain systems (ID.1), which suggests that parameter-efficient adapter tuning methods can address the mismatch issue effectively. **c)** The residual adapter (RA) (ID.7) applied at the first CNN layer achieves the best results relative to other adaptation approaches. It is noteworthy that RA adaptation using only 20% of adaptation parameters shows comparable performance to fully fine-tuned techniques (ID.3). In addition, the observation that the performance gain of downbeat tracking is greater than that of beat tracking is consistent with our finding in Fig. 2 that the downbeat tracking performance is more sensitive to the mismatch issue. **d)** When incorporating adapters into the second or third CNN layer (ID.8,9), with acceptable performance degradation, RA adaptation delivers a much lighter architecture with fewer parameters.

5.5 Performance of The Proposed Method

The advantages of the proposed method incorporating both late weighted linear interpolation feature fusion and RA adapter are demonstrated in Table 4. The evaluation results are the overall performance of all six evaluation sets using micro averaging. Two main observations can be found.

First, the multi-condition system using the proposed feature fusion approaches (ID.3) still outperforms the systems (ID.1,2) using only one spectral or SSL feature. Of particular interest, this system (ID.3) is compared to the existing singing voice beat tracking system [12], where the same evaluation protocol is followed by using the entire 5.9-hrs GTZAN dataset for testing³. As a result, our system achieved beat tracking F1-score of 0.784 on GTZAN, a significant **5.1%** absolute improvement compared to 0.733

³ The remaining five datasets are therefore used for training our system, which is less data compared to [12].

from [12] even using less training data.

Second, consistent performance improvements across all evaluation metrics are observed when the adapter tuning scheme (ID.4) is applied. Overall significant F1-score improvements of up to **31.6%** and **42.4%** absolute were obtained over the baseline un-adapted system using only one feature on the beat and downbeat tracking, respectively. In particular, the beat/downbeat performances of 0.87/0.78, 0.95/0.87, 0.85/0.75, 0.66/0.49, 0.68/0.49, and 0.79/0.41 are achieved by the proposed approach (ID.4) on the test split of GTZAN, RWC pop, Ballroom, Hainsworth, MUSDB, and URSing datasets, respectively.

6. CONCLUSIONS

This paper proposed a temporal convolution network based beat-tracking framework featuring self-supervised learning (SSL) representations and efficient adapter tuning to track the beat and downbeat of singing voices jointly. Feature fusion strategies were performed to leverage the advantages of the generic spectral and SSL speech feature representations. Efficient adapter tuning was utilized to mitigate the sources of variabilities of the non-homogeneous singing voice data. Experimental results showed that the proposed approach significantly outperforms the un-adapted baseline system using only spectral or SSL features. The inherent generality of the proposed approaches allows their further application to other beat-tracking systems or MIR tasks. Future work will focus on solving the data sparsity issue of the singing voice beat tracking task.

7. ACKNOWLEDGMENTS

This research is supported by Hong Kong RGC GRF grant No. 14200021, 14200220, Innovation & Technology Fund grant No. ITS/254/19 and ITS/218/21. Jiajun Deng carried out this research during his internship at Huawei Hong Kong Research Center. The authors would like to thank Huawei for its hardware and software services.

8. REFERENCES

- [1] R. Nishikimi, E. Nakamura, M. Goto, and K. Yoshii, "Audio-to-score singing transcription based on a crnn-hsmm hybrid model," *APSIPA Transactions on Signal and Information Processing*, vol. 10, p. 7, 2021.
- [2] Y.-K. Wu, C.-Y. Chiu, and Y.-H. Yang, "Jukedrummer: Conditional Beat-aware Audio-domain Drum Accompaniment Generation via Transformer VQ-VAE," in *International Society for Music Information Retrieval Conference (ISMIR)*, 2022, pp. 193–200.
- [3] P. E. Allen and R. B. Dannenberg, "Tracking musical beats in real time," in *ICMC*, 1990.
- [4] M. E. Davies and M. D. Plumbley, "Context-dependent beat tracking of musical audio," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 3, pp. 1009–1020, 2007.
- [5] A. Mottaghi, K. Behdin, A. Esmaili, M. Heydari, and F. Marvasti, "Obtain: Real-time beat tracking in audio signals," *International Journal of Signal Processing Systems*, 2017.
- [6] Y.-T. Wu, Y.-J. Luo, T.-P. Chen, I. Wei, J.-Y. Hsu, Y.-C. Chuang, L. Su *et al.*, "Omnizart: A general toolbox for automatic music transcription," *The Journal of Open Source Software*, vol. 6, no. 68, p. 3391, 2021.
- [7] B. Di Giorgi, M. Mauch, and M. Levy, "Downbeat tracking with tempo-invariant convolutional neural networks," *International Society for Music Information Retrieval Conference (ISMIR)*, 2021.
- [8] M. Goto and Y. Muraoka, "Musical understanding at the beat level: real-time beat tracking for audio signals," in *Computational Auditory Scene Analysis*. CRC Press, 2021, pp. 157–176.
- [9] Y.-N. Hung, J.-C. Wang, X. Song, W.-T. Lu, and M. Won, "Modeling beats and downbeats with a time-frequency transformer," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 401–405.
- [10] C. Hernandez-Olivan and J. R. Beltran, "Music composition with deep learning: A review," *Advances in Speech and Music Technology: Computational Aspects and Applications*, pp. 25–50, 2022.
- [11] M. Won, Y.-N. Hung, and D. Le, "A foundation model for music informatics," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 1226–1230.
- [12] M. Heydari and Z. Duan, "Singing beat tracking with self-supervised front-end and linear transformers," *International Society for Music Information Retrieval Conference (ISMIR)*, 2022.
- [13] J. L. Oliveira, F. Gouyon, L. G. Martins, and L. P. Reis, "Ibt: A real-time tempo and beat tracking system," in *International Society for Music Information Retrieval Conference (ISMIR)*, 2010, pp. 291–296.
- [14] A. Gkiokas, V. Katsouros, G. Carayannis, and T. Stajylakis, "Music tempo estimation and beat tracking by applying source separation and metrical relations," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2012, pp. 421–424.
- [15] S. Böck, F. Krebs, and G. Widmer, "Joint beat and downbeat tracking with recurrent neural networks," in *International Society for Music Information Retrieval Conference (ISMIR)*, 2016, pp. 255–261.
- [16] B. Lindblom and J. Sundberg, "The human voice in speech and singing," *Springer Handbook of Acoustics*, pp. 703–746, 2014.
- [17] M. Bunch and M. Bunch, *Dynamics of the singing voice*. Springer, 1982.
- [18] M. Heydari, J.-C. Wang, and Z. Duan, "Singnet: a real-time singing voice beat and downbeat tracking system," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.
- [19] A. Søgaard, *Semi-supervised learning and domain adaptation in natural language processing*. Springer Nature, 2022.
- [20] Y. Wang, S. Agarwal, S. Mukherjee, X. Liu, J. Gao, A. H. Awadallah, and J. Gao, "Adamix: Mixture-of-adaptations for parameter-efficient model tuning," *Empirical Methods in Natural Language Processing (EMNLP)*, 2022.
- [21] N. Ding, Y. Qin, G. Yang, F. Wei, Z. Yang, Y. Su, S. Hu, Y. Chen, C.-M. Chan, W. Chen *et al.*, "Parameter-efficient fine-tuning of large-scale pre-trained language models," *Nature Machine Intelligence*, vol. 5, no. 3, pp. 220–235, 2023.
- [22] P. Bell, J. Fainberg, O. Klejch, J. Li, S. Renals, and P. Swietojanski, "Adaptation algorithms for neural network-based speech recognition: An overview," *Open Journal of Signal Processing*, vol. 2, pp. 33–66, 2020.
- [23] B. Li, D. Hwang, Z. Huo, J. Bai, G. Prakash, T. N. Sainath, K. C. Sim, Y. Zhang, W. Han, T. Strohmaier *et al.*, "Efficient domain adaptation for speech foundation models," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.
- [24] K. C. Sim, Z. Huo, T. Munkhdalai, N. Siddhartha, A. Stooke, Z. Meng, B. Li, and T. Sainath, "A comparison of parameter-efficient asr domain adaptation methods for universal speech and language models,"

- in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 6900–6904.
- [25] E. Matthew Davies and S. Böck, “Temporal convolutional networks for musical audio beat tracking,” in *European Signal Processing Conference (EUSIPCO)*. IEEE, 2019, pp. 1–5.
- [26] S. Böck, M. E. Davies, and P. Knees, “Multi-task learning of tempo and beat: Learning one to improve the other.” in *International Society for Music Information Retrieval Conference (ISMIR)*, 2019, pp. 486–493.
- [27] S. Böck and M. E. Davies, “Deconstruct, analyse, reconstruct: How to improve tempo, beat, and downbeat estimation.” in *International Society for Music Information Retrieval Conference (ISMIR)*, 2020, pp. 574–582.
- [28] S. Singh, “Elu as an activation function in neural networks,” *Deep Learning University*, 2020.
- [29] K. P. Murphy *et al.*, “Dynamic bayesian networks,” *Probabilistic Graphical Models, M. Jordan*, vol. 7, p. 431, 2002.
- [30] D. P. Ellis, “Beat tracking by dynamic programming,” *Journal of New Music Research*, vol. 36, no. 1, pp. 51–60, 2007.
- [31] F. Krebs, S. Böck, and G. Widmer, “An efficient state-space model for joint tempo and meter tracking.” in *International Society for Music Information Retrieval Conference (ISMIR)*, 2015, pp. 72–78.
- [32] J. R. Zapata, M. E. Davies, and E. Gómez, “Multi-feature beat tracking,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 4, pp. 816–825, 2014.
- [33] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, “Hubert: Self-supervised speech representation learning by masked prediction of hidden units,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3451–3460, 2021.
- [34] H.-J. Chang, S.-w. Yang, and H.-y. Lee, “Distilhubert: Speech representation learning by layer-wise distillation of hidden-unit bert,” in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 7087–7091.
- [35] M. Heydari, F. Cwitkowitz, and Z. Duan, “Beatnet: Crnn and particle filtering for online joint beat downbeat and meter tracking,” *International Society for Music Information Retrieval Conference (ISMIR)*, 2021.
- [36] K. Gadzicki, R. Khamsehashari, and C. Zetzsche, “Early vs late fusion in multimodal convolutional neural networks,” in *International Conference on Information Fusion (FUSION)*. IEEE, 2020, pp. 1–6.
- [37] N. Mungoli, “Adaptive feature fusion: Enhancing generalization in deep learning models,” *arXiv preprint arXiv:2304.03290*, 2023.
- [38] Y. Dai, F. Gieseke, S. Oehmcke, Y. Wu, and K. Barnard, “Attentional feature fusion,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 3560–3569.
- [39] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, “Parameter-efficient transfer learning for nlp,” in *International Conference on Machine Learning (ICML)*, 2019, pp. 2790–2799.
- [40] J. He, C. Zhou, X. Ma, T. Berg-Kirkpatrick, and G. Neubig, “Towards a unified view of parameter-efficient transfer learning,” *International Conference on Learning Representations (ICLR)*, 2021.
- [41] H. Liu, D. Tam, M. Muqeeth, J. Mohta, T. Huang, M. Bansal, and C. A. Raffel, “Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning,” *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 1950–1965, 2022.
- [42] P. Swietojanski, J. Li, and S. Renals, “Learning hidden unit contributions for unsupervised acoustic model adaptation,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 8, pp. 1450–1463, 2016.
- [43] C.-L. Fu, Z.-C. Chen, Y.-R. Lee, and H.-y. Lee, “Adapterbias: Parameter-efficient token-dependent representation shift for adapters in nlp tasks,” *Findings of the Association for Computational Linguistics (ACL)*, 2022.
- [44] K. Tomanek, V. Zayats, D. Padfield, K. Vaillancourt, and F. Biadsy, “Residual adapters for parameter-efficient asr adaptation to atypical and accented speech,” *Empirical Methods in Natural Language Processing (EMNLP)*, 2021.
- [45] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, “Lora: Low-rank adaptation of large language models,” *International Conference on Learning Representations (ICLR)*, 2021.
- [46] A. Défossez, “Hybrid spectrogram and waveform source separation,” *ISMIR Workshop on Music Source Separation*, 2021.
- [47] M. E. Davies, N. Degara, and M. D. Plumbley, “Evaluation methods for musical audio beat tracking algorithms,” *Queen Mary University of London, Centre for Digital Music, Tech. Rep. C4DM-TR-09-06*, 2009.
- [48] G. Tzanetakis and P. Cook, “Musical genre classification of audio signals,” *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, pp. 293–302, 2002.

- [49] T. De Clercq and D. Temperley, "A corpus analysis of rock harmony," *Popular Music*, vol. 30, no. 1, pp. 47–70, 2011.
- [50] F. Gouyon, A. Klapuri, S. Dixon, M. Alonso, G. Tzanetakis, C. Uhle, and P. Cano, "An experimental comparison of audio tempo induction algorithms," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 5, pp. 1832–1844, 2006.
- [51] S. W. Hainsworth and M. D. Macleod, "Particle filtering applied to musical tempo tracking," *Journal on Advances in Signal Processing*, vol. 2004, pp. 1–11, 2004.
- [52] Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimilakis, and R. Bittner, "Musdb18-a corpus for music separation," Available: <https://doi.org/10.5281/zenodo.1117372>, 2017.
- [53] B. Li, Y. Wang, and Z. Duan, "Audiovisual singing voice separation." *Transactions of the International Society for Music Information Retrieval*, vol. 4, no. 1, pp. 195–210, 2021.
- [54] L. Wright and N. Demeure, "Ranger21: a synergistic deep learning optimizer," *arXiv preprint arXiv:2106.13731*, 2021.

WHICH AUDIO FEATURES CAN PREDICT THE DYNAMIC MUSICAL EMOTIONS OF BOTH COMPOSERS AND LISTENERS?

Eun Ji Oh¹

¹ Center for Digital Humanities
and Computational Social Sciences,
KAIST, Republic of Korea
muye35@kaist.ac.kr

Hyunjae Kim²

² Graduate School of
Culture Technology,
KAIST, Republic of Korea
present@kaist.ac.kr

Kyung Myun Lee¹²³

³ School of Digital Humanities
and Computational Social Sciences
KAIST, Republic of Korea
kmlee2@kaist.ac.kr

ABSTRACT

Are composers' emotional intentions conveyed to listeners through audio features? In the field of Music Emotion Recognition (MER), recent efforts have been made to predict listeners' time-varying perceived emotions using machine-learning models. However, interpreting these models has been challenging due to their black-box nature. To increase the explainability of models for subjective emotional experiences, we focus on composers' emotional intentions. Our study aims to determine which audio features effectively predict both composers' time-varying emotions and listeners' perceived emotions. Seven composers performed 18 piano improvisations expressing three types of emotions (*joy/happiness*, *sadness*, and *anger*), which were then listened to by 36 participants in a laboratory setting. Both composers and listeners continuously assessed the emotional valence of the music clips on a 9-point scale (1: 'very negative' to 9: 'very positive'). Linear mixed-effect models analysis revealed that listeners significantly perceived the composers' intended emotions. Regarding audio features, the RMS was found to modulate the degree to which the listener's perceived emotion resembled the composer's emotion across all emotions. Moreover, the significant audio features that influenced this relationship varied depending on the emotion type. We propose that audio features related to the emotional responses of composers-listeners can be considered key factors in predicting listeners' emotional responses.

1. INTRODUCTION

Music holds the power to convey emotions and evoke strong emotional responses in its listeners. There is a growing interest in utilizing Music Emotion Recognition (MER) systems for personalized music experiences, such as music recommendations, automated music generation, and diverse multimodal experiences. However, identifying the

variables that effectively predict listeners' emotional experiences is a challenging problem due to the complexity of its mechanisms [1]. While recent MER studies employ machine learning techniques to predict emotions based on dynamic listener annotations [2, 3], they often lack an interpretation of the underlying factors driving emotions.

This study aims to explain the prediction of musical emotions by empirically investigating the relationship between the composer's intended emotion, the listener's perceived emotion, and various audio features of music through time-series data. We specifically focus on the composer's emotional intentions during the music creation process, prior to listener exposure.

1.1 Background

MER tasks are inherently user-centered [4], bringing researchers from interdisciplinary fields such as musicology, cognitive science, and computer science. A range of factors, including individual traits (*e.g.*, personality, mood regulation strategies, etc.) and musical elements (*e.g.*, timbre, rhythm, harmony, etc.) [1], can impact the MER systems, posing challenges for enhancing model performance. Many MER studies rely on emotion datasets where listeners annotate their perceived or felt emotions [5, 6]. Given this, the outcome of the study can be significantly influenced by the taxonomy used to define emotions and the methods used to identify listeners' annotations [4, 7]. While previous studies have often relied on discrete emotion ratings [8–11], the latest trends favor continuous assessments that capture emotional fluctuations during music listening, reflecting the nature of music experiences [2, 3].

Recent studies on Music Emotion Recognition (MER) face several limitations. First, they often overlook the potential influence of emotions expressed by composers or performers on listeners' emotional experiences. Second, MER models commonly encounter challenges in accurately predicting valence compared to arousal [3].

The emotional intentions of composers/performers can play an important role in predicting listeners' emotions, but their significance is often underestimated. Composers or performers express their emotions through musical features such as tempo, dynamics, and timbre [12–14]. Listeners then perceive these cues and interpret the emotions conveyed by the music. When the emotions perceived by



© E.J. Oh, H. Kim, and K.M. Lee. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** E.J. Oh, H. Kim, and K.M. Lee, "Which audio features can predict the dynamic musical emotions of both composers and listeners?", in *Proc. of the 25th Int. Society for Music Information Retrieval Conf.*, San Francisco, United States, 2024.

the listener align closely with those expressed by the composer or performer, it can foster a strong connection between the listener and the composer/performer, which can positively impact listeners' emotional experiences [15]. This connection can also be observed in physiological responses; a previous study [16] has shown that the similarity of brain activity between audiences and violinists can predict the audiences' fondness for the performance.

Taken together, the relationship between listeners' and composers/performers' emotions is highly correlated with the emotional responses that listeners experience from music, such as music engagement and enjoyment. Therefore, we propose that composers' intentions may play an important role in MER systems that seek to predict listeners' emotions. To determine the impact of composers' intentions on predicting listeners' perceived emotions, we aim to compare prediction outcomes using only audio features against those utilizing both audio features and composers' emotional intentions.

Despite the potential importance of this relationship, there is a lack of research exploring the link between listeners and musical intentions. While some studies investigated how accurately emotional intentions were conveyed to listeners through discrete emotion ratings [8, 17, 18], there is a need to investigate the dynamic emotional responses of composer/performer and listener as the music unfolds over time.

1.2 Research Question

To examine the predictive role of audio features and emotional intentions in shaping listeners' perceived emotions, as well as the significance of time-varying emotional data in this context, we set the following research questions:

RQ1. Do the predictors of listeners' perceived emotions (audio features and composer's emotions) vary based on the methodology, discrete vs. dynamic emotional ratings?

RQ2 Which audio features predict the dynamics of the composer/performer's emotional intentions and listener-perceived emotions, respectively?

RQ3 Which audio features reflect both the composer/performer's and listeners' emotions?

To address these questions, we initially recruited composers to create emotionally expressive piano improvisations. We then collected composers' real-time assessments of the emotions they intended to convey during their performances. The emotional valence scale was only used for the assessments to reduce the complexity of predicting emotions. This approach may reduce cognitive overload for lay participants, who might find 2D emotion mapping (*arousal-valence*) unfamiliar.

For listeners' emotional data, we played the composers' music clips and instructed listeners to continuously infer the expressed emotion. Audio features were extracted via the *librosa* library, including root-mean-square (*RMS*), *flatness*, *zero-crossing*, *spectral centroid*, and *roll-off*, chosen based on previous research on audio features and emo-

tions [3, 9, 19].

We employed Linear Mixed-Effects (LME) models for multi-level regression analysis, which are suitable for handling hierarchical, non-independent time-series data. By accounting for variability within and between music clips, we investigated whether listeners effectively captured changes in the composer's intentions, independent of the specific characteristics of individual clips [20].

2. MATERIALS

2.1 Composers' Emotion Data

2.1.1 Participants

We recruited eight composers from various composition departments in the College of Music, Republic of Korea (4 males and 4 females, $M = 26.88$, $SD = 1.73$). These participants were either undergraduate students or recent graduates with a bachelor's degree in Western classical music composition. On average, they had 14.13 years of formal music training ($SD = 5.72$), with an average of 10.25 years of piano experience ($SD = 3.28$). All composers had prior experience in improvised performances.

2.1.2 Music Performance Setting

Composers were instructed to prepare three semi-improvised piano performances, each lasting 1-2 minutes, expressing primary emotions: *joy/happiness*, *sadness*, and *anger*. These emotions were chosen based on prior literature [10, 20–22] for their distinctiveness in conveying or interpreting emotions through music.

Performances took place in a soundproof booth using a Casio Contemporary CDP-120 digital keyboard, with default piano sound and fixed volume settings. Video recordings were made using a Canon EOS 5D Mark IV Full Frame DSLR, capturing audio via the built-in microphone. The recordings were in .mp4 format, with a resolution of 1920 x 1080, 25 fps, and an audio sampling rate of 48 kHz.

2.1.3 Recording Procedure

Composers were briefed about the experiment and provided consent. They had 15 minutes to prepare, followed by a 30-second sample performance for technical setup. The order of recording for the three performances was randomized, with breaks between each to refresh emotions.

After each performance, composers rated their expressed emotions using arousal, valence, and dominance on a 9-point Likert scale (discrete ratings). Following the recording session, they watched videos of themselves in a randomized order, continuously rating the emotions they expressed during the performance on a 9-point valence scale (1: 'very negative' - 9: 'very positive') in real-time, mirroring the setup described in Section 3.2.

2.1.4 Music Selection

Twenty-four music clips were initially recorded, featuring performances of three emotions by eight composers. Five authors and colleagues participated in the decision-making process for music selection. The selection criteria ensured

	joy/happiness	sadness	anger
arousal	6.33 (2.16)	3 (1.87)	7.57 (1.27)
valence	8.33 (0.82)	3.6 (0.55)	2.14 (0.69)
dominance	6.17 (1.94)	4.8 (1.64)	7.71 (1.60)

Table 1. The *mean* (*SD*) scores of emotion provided by composers for the final 18 music clips ($n = 6$ for each emotion). They were assessed with a 9-point Likert scale.

that 1) each clip effectively conveyed its intended emotion (e.g., a performance expressing sadness was excluded since some researchers felt it was positive valenced music) and 2) was free from distracting noise (e.g., the sound of fingernails on keyboards). An equal distribution of male and female composers per emotion was maintained resulting in 18 chosen clips (six per emotion) from four males and three females.

All 18 clips were pre-processed using Adobe Premiere Pro, ensuring .wav format, 44.1 kHz sampling rate, 16-bit depth, stereo, and normalization according to ITU BS.1770-3 standards¹. The mean clip length was 97.5 seconds (*SD* 14.50), ranging from 73 to 125 seconds. The mean scores of discrete emotional ratings provided by composers are shown in Table 1.

2.2 MIR Audio Features

To select the audio features, we reviewed prior research on emotion perception and acoustic features. Studies highlighted the importance of timbre, tempo, mode, harmony, loudness, and pitch in emotional communication [2, 9, 23, 24]. In particular, tonality, pitch, harmony, articulation, and timbre (e.g., brightness, roughness) were crucial for predicting emotion valence [25, 26]. Machine-learning methods have shown that valence emotion prediction models achieve high explanatory power when incorporating spectral [3, 19] and rhythmic features [19] available in the *librosa* package [27]. Based on this, we used *librosa* to extract audio features from 18 music clips, focusing on loudness (root-mean-square; *RMS*), timbre (*flatness*, *zero-crossing*, *spectral centroid*, and *roll-off*), harmony (Mel-Scale Frequency Cepstral Coefficients; *MFCC*, *chroma*, *spectral contrast*), and rhythm (*dynamic tempo*). To compare audio features with 2D data (*time-valence*) of composers' and listeners' emotional ratings, we selected five features: *RMS*, *flatness*, *zero-crossing*, *spectral centroid*, and *roll-off*. These features were computed using non-overlapping 500 ms windows to match the 2 Hz sampling rate of the emotional ratings.

2.3 Linear Mixed-Effect Models

The linear mixed-effects (LME) model to predict the dynamics of listeners' perceived emotions based on composers' emotional intentions is formulated as:

$$y_{ij} = \alpha + \beta x_{ij} + a_i + b_i x_{ij} + \epsilon_{ij} \quad (1)$$

This equation describes how listeners' perceived emotions (y_{ij}) relate to composers' emotional intentions (x_{ij}) for each music clip (i) at each time point (j). α , β , a_i , and b_i represent the intercept, coefficient for composers' emotional intentions, random intercept for each clip, and random slope for composers' emotional intentions within each clip, respectively. Terms (a_i, b_i) follow a bivariate normal distribution, while ϵ_{ij} represents the residual error.

To investigate the influence of a specific audio feature on this relationship, we employed a LME model:

$$y_{ij} = \alpha + \beta_1 x_{ij} + \beta_2 \cdot feature_{ij} + \beta_3 x_{ij} \cdot feature_{ij} + a_i + b_i x_{ij} + \epsilon_{ij} \quad (2)$$

The terms y_{ij} and x_{ij} represent listeners' perceived emotions and composers' emotional intentions, respectively, for each music clip (i) at each time point (j). α , β_1 , β_2 , and β_3 denote the intercept, composer's emotional intention coefficient, audio feature coefficient, and their interaction coefficient. Random intercept (a_i) and slope (b_i) account for variation within each clip, while ϵ_{ij} represents the residual error.

3. EXPERIMENT

3.1 Participants

We recruited 36 participants (19 males, 17 females; *mean* age 26.06, *SD* 3.56) through campus mail and online bulletin boards. Except for one participant, who held a master's degree in piano performance, all others were non-musicians. On average, they had about 5.81 years of musical training (*SD* 3.91).

To minimize cultural influences on emotional judgments, only native Korean speakers were included in the experiment. Participants had to meet certain criteria: aged 20 or older, normal vision and hearing, no hand movement disabilities, no diagnosed neurological or psychiatric conditions, and no current use of psychiatric medications.

3.2 Emotional Ratings

The experiment was primarily designed to examine the modality effects on musical emotion inference [28, 29]. Using a counterbalanced design, each participant rated six out of 18 music clips per modality (audio-only, video-only, and video-and-audio), with two clips per emotion (joy/happiness, sadness, and anger). This resulted in 216 emotional ratings for each modality and a total of 648 ratings collected across all clips. We used the 216 emotional ratings from the audio-only condition for the analysis to investigate listeners' emotional experiences during music listening in a more ecological setting.

The dynamic emotional rating task was conducted using *PsychoPy* software, mirroring the dynamic emotional ratings by composers described in Section 2.1.3. Participants,

¹ Sample music clips and supplementary materials are available at https://osf.io/4dcxu/?view_only=3f1d818e5c4f4e698ebca357daa656cc.

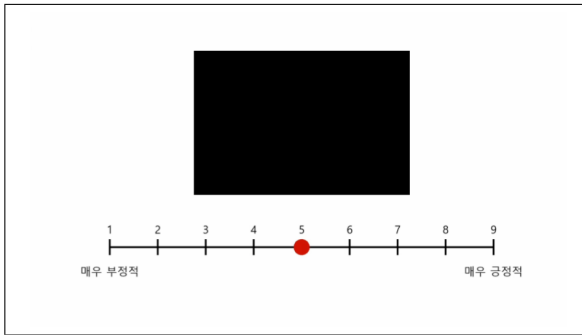


Figure 1. Screenshot of participants' emotional rating task on a valence scale (1: 'very negative' and 9: 'very positive', labels in Korean).

referred to as *listeners*, were instructed to infer the emotional states of composers expressed in the music. While listening, listeners moved a red dot (initially positioned at 5) along a valence scale (1: 'very negative' - 9: 'very positive') whenever they perceived a change in the composer's emotional state [20, 30] (see Figure 1). Ratings were recorded at a sampling rate of 2 Hz, with timestamps every 0.5 seconds.

After each clip, participants evaluated their psychological state using a 9-point Likert scale for arousal, valence, dominance, flow, and empathy. These assessments aimed to minimize the influence of previous emotional experiences on subsequent ratings, and the results were not included in this paper.

3.3 Experiment Procedure

Participants arrived at the lab, completed consent forms, and filled out questionnaires about their music experience. In a soundproof booth, they then performed an emotional inference task. After a practice trial, they listened to six predetermined music clips, inferring the composer's expressed emotion by adjusting a red circle on a scale. Following each clip, they answered five questions about their psychological state and could take breaks. The task was conducted using headphones, with participants adjusting the volume to their preference.

3.4 Data Analysis

All behavior ratings were interpolated using the *scipy* package in Python to maintain consistent time intervals. Silent sections were manually removed from the beginning and end of each audio file before analysis. Audio features were normalized between 0 and 1 at the composer level using min-max normalization. We found strong multicollinearity between spectral centroid and roll-off, so the roll-off feature was excluded from the final analysis to avoid potential overfitting.

For the analysis of listeners' perceived emotions, we selected one representative emotional rating from the responses of 12 listeners for each music clip. The representative value was calculated as the median of 12 ratings for each time point of each clip (see Figure 2). Thus, one time-

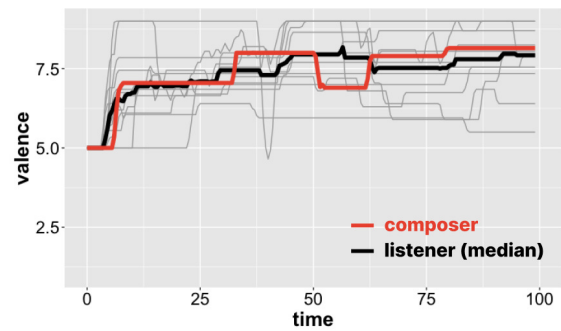


Figure 2. Emotional ratings for a sample joy/happiness music clip: time (x-axis), valence (y-axis, 1: 'very negative' to 9: 'very positive'). Red line shows the composer's ratings, black line the median of 12 listeners, and gray lines individual listener ratings.

series data per music stimulus was used for data analysis as *listeners'* emotions. This means that each music clip retained one composer emotion rating, one listener rating, and four audio features. Additionally, the average of listener ratings was used as discrete emotions for comparison with composers' discrete ratings and audio features, as listeners' discrete ratings were not collected (see section 4.2).

Intra-class correlation (ICC) was computed to assess agreement over time among the listener data using the 'ICC' function in the R package *psych*. A two-way mixed, average score ICC was employed for consistency in the 12 valence ratings, following prior research on continuous emotional annotations [2, 31]. The results of this analysis can be found in the supplementary material.

Linear mixed-effects (LME) models were fitted using the *lme4* [32] and the *lmerTest* package [33] in R. Random effects were included in the model structure, and it was found that the random slope of composers' emotions significantly improved the model fit. The random slope was added since the relationship between listeners' perceived emotions and composers' expressed emotions may vary depending on the music clips.

4. RESULTS

4.1 Composers-Listeners Discrete Emotions

To assess the predictability of listeners' perceived emotions using discrete values, we used an LME model analysis. The dependent variable was the average emotional rating from listeners' representative data per music clip. We compared two models: *Model 1* used four audio features (RMS, flatness, zero-crossing, and spectral centroid; the average value of each music clip) as predictors, while *Model 2* added composers' discrete emotional ratings (arousal, valence, and dominance) with four audio features. P-values for fixed effects were obtained using Satterthwaite's approximations, and confidence intervals were computed using the Wald method. Refer to the supplementary material for detailed results of each model.

Model 1 showed that all four audio features sig-

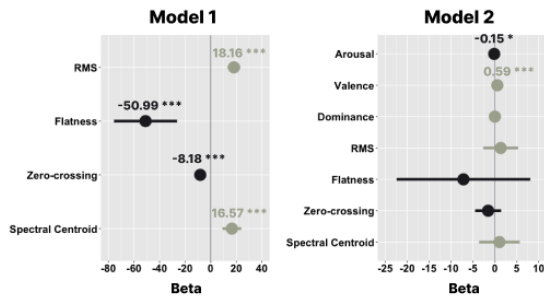


Figure 3. Plot of the effect size of two models.

Metric	Model 1	Model 2	P-value
MAE	2.11	1.17	0.012*
MSE	6.29	2.10	0.009**
RMSE	2.24	1.33	0.009**
MAPE	0.62	0.32	0.012*

Table 2. The mean metric values of each model’s leave-one-song-out cross-validation for 18 music stimuli. The values were compared using the Wilcoxon signed-rank test.

nificantly predicted listeners’ perceived emotions (see Figure 3). In Model 2, only composers’ arousal and valence were significant predictors, with no significant fixed effects for the audio features (see Figure 3). Using leave-one-song-out cross-validation, we confirmed that Model 2 predicted more accurately than Model 1, even for unseen data (see Table 2). This is indicated by its better performance across four metrics: mean absolute error (MAE), mean squared error (MSE), root mean squared error (RMSE), and mean percentage error (MPE).

4.2 Composers-Listeners Continuous Emotions

4.2.1 All Emotions

The relationships between dynamic composers’ emotional intentions and listeners’ perceived emotions were analyzed with an LME model (Equation 1 from Section 2.3). The dependent variable was listeners’ emotional representative ratings, and composers’ emotion ratings served as the predictor across all music clips (total observations, $N = 3438$; music clips, $N = 18$). The LME analysis revealed that composers’ emotional intentions significantly predicted listeners’ perceived emotions ($\beta = 0.26$, $p < 0.001$; see Figure 4). Separate analyses for each emotion indicated a significant association between composer-listener emotions except for *joy/happiness* ($p = 0.144$).

4.3 Audio Features & Musical Emotions

LME models were employed to predict composers’ and listeners’ emotions (see Table 3). For composers’ emotions, spectral centroid significantly predicted *all emotions* ($\beta = 0.74$, $p < 0.001$) and for *joy/happiness*, RMS, zero-crossing, and spectral centroid were significant predictors (RMS: $\beta = 0.60$, $p < 0.001$; zero-crossing: $\beta = -1.11$, $p < 0.001$; spectral centroid: $\beta = 1.09$, $p = 0.020$). For

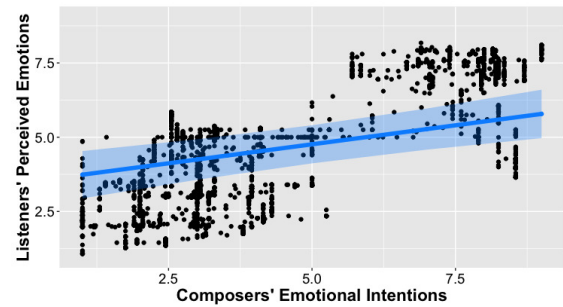


Figure 4. Plot showing listeners’ emotions predicted by composers’ emotions, with a regression line indicating a slope ($\beta = 0.26$) of the fixed effect for composers.

sadness, RMS was significant ($\beta = -0.33$, $p = 0.016$), and for *anger*, RMS and spectral centroid were significant predictors (RMS: $\beta = -0.35$, $p = 0.042$; spectral centroid ($\beta = 1.51$, $p < 0.001$).

In the LME model predicting listeners’ emotions, adding composers’ emotional ratings with audio features significantly improved the model fit across all 18 music clips and for each emotion-specific model. RMS consistently predicted listeners’ emotions, and zero-crossing emerged as a significant predictor for *anger* music ($\beta = 0.38$, $p = 0.045$).

4.3.1 All Emotions

Building on previous findings, we explored whether audio features that significantly predicted composers’ and listeners’ emotions could simultaneously predict both subjects’ emotions (Equation 2 from Section 2.3). An LME model with listeners’ ratings as the dependent variable, and composers ratings, RMS, and their interaction term for predictors (N total observations = 3438; AIC = 5970.1, LogLik = -2977.0), outperformed the model in Section 4.2.1 ($X^2 = 57.24$, $p < 0.001$).

All fixed effects terms were statistically significant in predicting listeners’ emotions, particularly the interaction term between composer ratings and RMS ($\beta = 0.15$, $p < 0.001$). This suggests that the relationship between composer and listener emotions varied significantly with changes in RMS levels (see Figure 5), highlighting RMS’s role in modulating both composer and listener emotions. Conversely, the interaction term with spectral centroid was not statistically significant ($\beta = -0.06$, $p = 0.055$).

4.3.2 Joy/Happiness

As in Section 4.3.1, we assessed interaction terms between audio features and composers in LME models for each emotion, focusing on *joy/happiness*. Using RMS, zero-crossing, and spectral centroid as predictors, each including an interaction term with composer ratings. Results indicated statistically significant interaction effects across all models: *composer x RMS* ($\beta = -0.55$, $p < 0.001$), *composer x zero-crossing* ($\beta = 0.58$, $p < 0.001$), and *composer x spectral centroid* ($\beta = 0.74$, $p < 0.001$).

	Composer	Listener	Composer & Listener
<i>All Emotions</i>	Spectral centroid	Composer, & RMS	RMS
<i>Joy/Happiness</i>	RMS, Zero-crossing, & Spectral centroid	Composer, & RMS	RMS, Zero-crossing, & Spectral centroid
<i>Sadness</i>	RMS	Composer, & RMS	-
<i>Anger</i>	RMS, & Spectral centroid	Composer, RMS, & Zero-crossing	RMS, & Zero-crossing

Table 3. Significant predictors included four audio features for composers’ emotional intentions and listeners’ perceived emotions. Composers’ emotions were added as predictors in the models predicting listeners’ emotions.

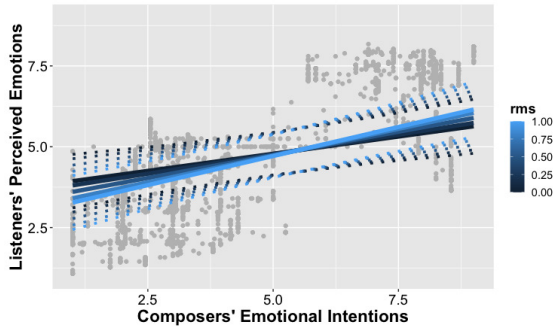


Figure 5. An interaction plot illustrating the model predicting listeners’ emotions with composers’ emotions and RMS. The solid line depicts the slope of the fixed effect of *composer*, varying with changes in RMS values.

4.3.3 Sadness

For *sadness*, an LME model was fitted with the RMS feature and the interaction term between RMS and composer as predictors. The results showed that neither the interaction term ($p = 0.409$) nor the fixed effect of the RMS feature ($p = 0.365$) were statistically significant.

4.3.4 Anger

The LME model for *anger* music included fixed effects and interaction terms for RMS, zero-crossing, and spectral centroid features. Results showed that the interaction terms for RMS (beta = -0.32 , $p < 0.001$) and zero-crossing (beta = 0.53 , $p < 0.001$) were statistically significant. However, the spectral centroid model did not show significant results upon model comparison ($p = 0.222$).

5. DISCUSSION

In this study, we employed linear mixed-effects (LME) models to explore how spectral features of music predict both the composer’s real-time intended emotional expression during piano improvisations and the listener’s perceived emotion. This included gathering emotional ratings empirically from composers and also listeners on a valence scale. We then examined the relationship between these ratings and the features extracted from the music clips.

We found that composers’ emotional intentions were conveyed to listeners’ perceptions of musical emotions. Discrete emotional ratings showed that composers’ intentions were stronger predictors of listeners’ perceived emotions than other audio features. Conversely, continuous emotional data emphasized the importance of both composers’ intentions and RMS features. These results un-

derscored the impact of emotional assessment methodologies, suggesting that discrete emotion ratings may overlook acoustic cues conveying composers’ intentions.

Overall, RMS was identified as a primary predictor for conveying composers’ intentions and also served as an indicator of listeners’ emotional perceptions. While RMS was the key feature for predicting listeners’ emotions, the features that indicated composers’ emotions varied with different emotional categories. For *joy/happiness* and *anger*, the spectral centroid emerged as the main predictor of the composers’ intentions, likely due to its association with timbral brightness, which helps detect changes in the valence [26].

Our findings highlight RMS as a crucial audio feature for predicting the emotions of both composers and listeners. RMS was strongly associated with emotional dynamics in *joy/happiness* and *anger*, but not in *sadness*. This is consistent with prior research [9], which also found RMS to be an effective predictor of *happiness* and *anger*, but not *sadness*. Additionally, zero-crossing emerged as a significant predictor of the emotional relationship between composers and listeners for both *joy/happiness* and *anger*, further aligning with the findings of previous studies on speech emotion recognition [34].

However, we found no audio features capable of predicting composer-listener emotions for *sadness*, which typically involves lower arousal compared to *joy/happiness* and *anger*. In music with low arousal, features related to valence may not be as prominent. For instance, in *sad* music, changes in loudness (i.e., RMS) may not be as pronounced as in *joy/happiness* or *anger*, thus potentially not serving as cues for both composers’ emotional intentions and listeners’ perceptions of valence.

Future research should further explore more audio features related to other musical factors (e.g., tempo, pitch, harmony, etc.) that are known to be associated with emotional experiences in music. Additionally, it is needed to determine whether the features identified in this study enhance the performance of MER models. Expanding the sample size of participants and utilizing a larger pool of music stimuli, while considering individual and cultural variations, will also be essential to enhance generalizability and gain more comprehensive insights.

This study offers insights into factors influencing MER system predictions of emotional valence, enhancing machine-learning models’ ability to predict listeners’ emotions by considering feature importance across different emotions. Additionally, incorporating time-series emotional data, including composers’ intentions, adds further significance to the research.

6. ACKNOWLEDGMENTS

This work was supported by the Ministry of Education of the Republic of Korea and the National Research Foundation of Korea (NRF-2023R1A2C100475512).

7. ETHICS STATEMENT

IRB approval was obtained by the Korea Advanced Institute of Science and Technology (KH2023-070).

8. REFERENCES

- [1] M. Schedl, A. Flexer, and J. Urbano, "The neglected user in music information retrieval research," *Journal of Intelligent Information Systems*, vol. 41, no. 3, pp. 523–539, 2013.
- [2] S. Yang, C. N. Reed, E. Chew, and M. Barthelet, "Examining emotion perception agreement in live music performance," *IEEE transactions on affective computing*, vol. 14, no. 2, pp. 1442–1460, 2021.
- [3] S. Chaki, P. Doshi, S. Bhattacharya, and P. Patnaik, "Explaining perceived emotion predictions in music: An attentive approach." in *ISMIR*, 2020, pp. 150–156.
- [4] J. S. Gómez-Cañón, E. Cano, T. Eerola, P. Herrera, X. Hu, Y.-H. Yang, and E. Gómez, "Music emotion recognition: Toward new, robust standards in personalized and context-sensitive applications," *IEEE Signal Processing Magazine*, vol. 38, no. 6, pp. 106–114, 2021.
- [5] S. Koelstra, C. Muhl, M. Soleymani, J.-S. Lee, A. Yazdani, T. Ebrahimi, T. Pun, A. Nijholt, and I. Patras, "Deap: A database for emotion analysis; using physiological signals," *IEEE transactions on affective computing*, vol. 3, no. 1, pp. 18–31, 2011.
- [6] A. Aljanaki, Y.-H. Yang, and M. Soleymani, "Developing a benchmark for emotional analysis of music," *PloS one*, vol. 12, no. 3, p. e0173392, 2017.
- [7] T. Eerola and J. K. Vuoskoski, "A review of music and emotion studies: Approaches, emotion models, and stimuli," *Music Perception: An Interdisciplinary Journal*, vol. 30, no. 3, pp. 307–340, 2012.
- [8] L. Turchet and J. Pauwels, "Music emotion recognition: intention of composers-performers versus perception of musicians, non-musicians, and listening machines," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 305–316, 2021.
- [9] E. B. Lange and K. Frieler, "Challenges and opportunities of predicting musical emotions with perceptual and automatized features," *Music Perception: An Interdisciplinary Journal*, vol. 36, no. 2, pp. 217–242, 2018.
- [10] J. Akkermans, R. Schapiro, D. Müllensiefen, K. Jakubowski, D. Shanahan, D. Baker, V. Busch, K. Lothwesen, P. Elvers, T. Fischinger *et al.*, "Decoding emotions in expressive music performances: A multi-lab replication and extension study," *Cognition and Emotion*, vol. 33, no. 6, pp. 1099–1118, 2019.
- [11] F. Pan, L. Zhang, Y. Ou, and X. Zhang, "The audio-visual integration effect on music emotion: Behavioral and physiological evidence," *PloS one*, vol. 14, no. 5, p. e0217040, 2019.
- [12] P. N. Juslin and P. Laukka, "Communication of emotions in vocal expression and music performance: Different channels, same code?" *Psychological bulletin*, vol. 129, no. 5, pp. 770–814, 2003.
- [13] P. E. Keller, "Ensemble performance: Interpersonal alignment of musical expression," *Expressiveness in music performance: Empirical approaches across styles and cultures*, vol. 1, pp. 260–282, 2014.
- [14] P. A. Holmes, "An exploration of musical communication through expressive use of timbre: The performer's perspective," *Psychology of Music*, vol. 40, no. 3, pp. 301–323, 2012.
- [15] A. C. Miu and F. R. Balteş, "Empathy manipulation impacts music-induced emotions: A psychophysiological study on opera," *PloS one*, vol. 7, no. 1, p. e30618, 2012.
- [16] Y. Hou, B. Song, Y. Hu, Y. Pan, and Y. Hu, "The averaged inter-brain coherence between the audience and a violinist predicts the popularity of violin performance," *Neuroimage*, vol. 211, p. 116655, 2020.
- [17] S. Vieillard, I. Peretz, N. Gosselin, S. Khalfa, L. Gagnon, and B. Bouchard, "Happy, sad, scary and peaceful musical excerpts for research on emotions," *Cognition & Emotion*, vol. 22, no. 4, pp. 720–752, 2008.
- [18] P. N. Juslin, "Cue utilization in communication of emotion in music performance: Relating performance to perception." *Journal of Experimental Psychology: Human perception and performance*, vol. 26, no. 6, pp. 1797–1812, 2000.
- [19] L. Xu, X. Wen, J. Shi, S. Li, Y. Xiao, Q. Wan, and X. Qian, "Effects of individual factors on perceived emotion and felt emotion of music: based on machine learning methods," *Psychology of Music*, vol. 49, no. 5, pp. 1069–1087, 2021.
- [20] B. A. Tabak, Z. Wallmark, L. H. Nghiem, T. Alvi, C. S. Sunahara, J. Lee, and J. Cao, "Initial evidence for a relation between behaviorally assessed empathic accuracy and affect sharing for people and music." *Emotion*, vol. 23, no. 2, pp. 437–449, 2023.
- [21] A. S. Cowen, X. Fang, D. Sauter, and D. Keltner, "What music makes us feel: At least 13 dimensions organize subjective experiences associated with music across different cultures," *Proceedings of the National*

- Academy of Sciences*, vol. 117, no. 4, pp. 1924–1934, 2020.
- [22] C. MacGregor, N. Ruth, and D. Müllensiefen, “Development and validation of the first adaptive test of emotion perception in music,” *Cognition and Emotion*, vol. 37, no. 2, pp. 284–302, 2023.
- [23] A. Gabrielsson and E. Lindström, “The influence of musical structure on emotional expression,” in *Music and emotion: Theory and research*, P. N. Juslin and J. A. Sloboda, Eds. Oxford University Press, 2001, pp. 223–248.
- [24] T. Eerola, A. Friberg, and R. Bresin, “Emotional expression in music: contribution, linearity, and additivity of primary musical cues,” *Frontiers in psychology*, vol. 4, p. 487, 2013.
- [25] C. Plut, P. Pasquier, J. Ens, and R. Tchemeube, “The isovat corpus: Parameterization of musical features for affective composition,” *Transactions of the International Society for Music Information Retrieval*, vol. 5, no. 1, pp. 173–189, 2022.
- [26] I. Salakka, A. Pitkäniemi, E. Pentikäinen, K. Mikkonen, P. Saari, P. Toiviainen, and T. Särkämö, “What makes music memorable? relationships between acoustic musical features and music-evoked emotions and memories in older adults,” *PloS one*, vol. 16, no. 5, p. e0251692, 2021.
- [27] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, “librosa: Audio and music signal analysis in python.” in *SciPy*, 2015, pp. 18–24.
- [28] E. J. Oh, “Shared empathic process in music and social contexts: exploring empathic accuracy and physiological responses across modalities and valence,” Master’s thesis, KAIST, 2024.
- [29] E. J. Oh and K. M. Lee, “Intermodal analysis of emotion inference: Examining shared processes in music and social contexts.” in *Society for Music Perception and Cognition (SMPC)*, 2024, p. 68.
- [30] J. Zaki, N. Bolger, and K. Ochsner, “It takes two: The interpersonal nature of empathic accuracy,” *Psychological science*, vol. 19, no. 4, pp. 399–404, 2008.
- [31] N. Dibben, E. Coutinho, J. A. Vilar, and G. Estévez-Pérez, “Do individual differences influence moment-by-moment reports of emotion perceived in music and speech prosody?” *Frontiers in behavioral neuroscience*, vol. 12, p. 184, 2018.
- [32] D. Bates, M. Maechler, B. Bolker, S. Walker, R. H. B. Christensen, H. Singmann, B. Dai, G. Grothendieck, P. Green, and M. B. Bolker, “Package ‘lme4,’” *convergence*, vol. 12, no. 1, p. 2, 2015.
- [33] A. Kuznetsova, P. B. Brockhoff, and R. H. B. Christensen, “lmerTest package: tests in linear mixed effects models,” *Journal of statistical software*, vol. 82, no. 13, 2017.
- [34] E. Ramdinmawii, A. Mohanta, and V. K. Mittal, “Emotion recognition from speech signal,” in *TENCON 2017 - 2017 IEEE Region 10 Conference*, 2017, pp. 1562–1567.

EXPLORING MUSICAL ROOTS: APPLYING AUDIO EMBEDDINGS TO EMPOWER INFLUENCE ATTRIBUTION FOR A GENERATIVE MUSIC MODEL

Julia Barnett
Northwestern University

Hugo Flores García
Northwestern University

Bryan Pardo
Northwestern University

ABSTRACT

Every artist has a creative process that draws inspiration from previous artists and their works. Today, “inspiration” has been automated by generative music models. The black box nature of these models obscures the identity of the works that influence their creative output. As a result, users may inadvertently appropriate or copy existing artists’ works. We establish a replicable methodology to systematically identify similar pieces of music audio in a manner that is useful for understanding training data attribution. We compare the effect of applying CLMR [1] and CLAP [2] embeddings to similarity measurement in a set of 5 million audio clips used to train VampNet [3], a recent open source generative music model. We validate this approach with a human listening study. We also explore the effect that modifications of an audio example (e.g., pitch shifting) have on similarity measurements. This work is foundational to incorporating automated influence attribution into generative modeling, which promises to let model creators and users move from ignorant appropriation to informed creation. Audio samples accompanying this paper are available at tinyurl.com/exploring-musical-roots.

1. INTRODUCTION

For creators and users of generative models to be informed and responsible, there needs to be a mechanism that provides information about works in the model’s training data that were highly influential upon generated outputs. This would enable both citation of existing work and offer the opportunity to learn about the influences of their creation. We assume a model-generated product that is a copy or near-copy of a work in the model’s training set indicates the model was influenced by that work. To develop methods to automatically detect the influences upon model-generated products it is, therefore, essential to develop good measures of similarity between works.

We define a measure of approximate memorization in deep generative audio models by establishing a thresh-

old for high similarity and memorization of training data against a large repertoire of 5,000,000+ song clips. We take inspiration from the “split-product” measure for image similarity from Somepalli et al. [4], which breaks the embedded feature vector of images into smaller chunks to compare inner products of corresponding localized features. In our work, every audio file is split into 3-second segments (a.k.a. clips), each of which is encoded as a feature vector (either a CLMR [1] or CLAP [2] embedding) produced by a machine learning model trained to encode audio for the purpose of measuring similarity. See Section 3.3.2 for details. We measure similarity between generated clips and training data clips to find similarity between subportions of songs (e.g., a single musical phrase), returning the songs with the most similar clips. We also evaluate the extent to which similarity measured in this way agrees with similarity assessments by human listeners (Section 4).

We apply our approach to VampNet [3], an open-source music audio generation model trained on 795k music songs. VampNet is representative of a widely-used class of generative models: language-model-style generation. This approach is used in AudioLM [5], JukeBox [6], Musi-cLM [7], SoundStorm [8], among others [9, 10]. While we utilize VampNet as a case study, our evaluation framework is both model and training data agnostic.

This paper makes the following **key contributions**. Primarily, it establishes **an easily replicable methodology and framework to perform training data attribution for a generative music model** (Section 3), which has been **validated in a human-listener study** (Section 4). Second, **we systematically explore the robustness of embedding-based similarity measures for music audio (CLMR and CLAP) to audio perturbations such as pitch shift, time stretch, and mixture with different types of noise** (Section 5.1). Generative models, even when creating near-copies of training data, are likely to add some form of variation to the outputs, making it essential to understand how robust this method is to such anticipated perturbations.

Our formal research questions are:

1. Can we measure similarity between generated music and music in the training data in a way that human listeners would agree with?
2. How do different perturbation types and amounts affect the ability of the evaluated similarity measure(s) to quantitatively identify similar pieces of music?



2. RELEVANT LITERATURE

2.1 Memorization in Non-Audio Generative Models

It is well established that large language models (LLMs) applied to text are capable of memorizing part of their training data [11–18]. LLMs like the 6 billion parameter GPT-J model can memorize at least 1% of training data [19]. If access to the training data is available, it is relatively straightforward to detect when language models copy strings of text verbatim due to the ability to check for exact sequences of tokens.

Memorized images created by generative models pose risks similar to memorized training data from text LLMs such as sensitive data leaks and copyright infringement. Detecting memorization and duplication by image models is fundamentally different from detecting duplication from a text-based language model; instead of memorizing and reproducing items verbatim from the training data, image models create images that are not identical to the training data, but are sufficiently similar to warrant being called content replication [4].

Carlini et al. [20] propose an approximation of a distance metric for memorization in the image space. A generated image whose nearest neighbor in the training data falls closer than a determined threshold (δ), when embedded in the appropriate manifold, is labeled as a memorized example even if not a verbatim copy. Somepalli et al. [4] demonstrate that diffusion models replicate images from training data with high fidelity, setting a lower bound for memorization of Stable Diffusion at 1.88% of generations [21]. We extend this methodology [4, 20] into the audio domain for our paper.

2.2 Audio Retrieval and Music Similarity

2.2.1 Music Similarity in Generative Audio Models

Popularized in the early 2000s, audio fingerprinting [22, 23] aims to detect exact copies of a given piece of audio. In 2006, Shazam popularized this method for the general public with a system utilizing query-by-example for everyday users [24]. Traditional audio fingerprinting (e.g., Shazam [25]) depends on low-level structural details that are not typically regenerated by generative models, so it is not a relevant approach for our methodology.

Most of the limited work examining similarity of audio made by generative models has been in the context of a different purpose, rather than the focus of an in-depth exploration. Examples include creating new strategies for text-to-music generation in order to create more novel songs [26] or brief ad-hoc memorization evaluations at time of release [7, 9]. Perhaps the closest work to our own is by Bralios et al. [27], who examined replication of audio utilizing text-to-audio latent diffusion models for general audio sounds, such as explosions or people cheering. They define replication of training data as “nearly-identical complex spectro-temporal patterns.” They did not perform any subjective evaluation by human listeners to validate their approach to measuring similarity. Our work instead focuses on music, uses a much larger dataset, and is intended

to be easily adoptable by any model creator.

2.2.2 Measuring Audio Similarity with Embeddings

The key to measuring similarity effectively is to have a representation that highlights the task-relevant features. Most popular right now in the age of generative modeling is measuring audio similarity with embeddings. Audio embeddings are continuous vector representations for excerpts of audio that are based on the internal representations of a neural model trained on a proxy task like generative pre-training [28], contrastive learning [1, 2], classification [29], autoencoding [30, 31], and other methods [32, 33].

To use an audio embedding model to measure the similarity of a collection of audio excerpts, we pass the audio signals through the embedding network, which gives us a multi-dimensional vector output for each audio signal: the “audio embedding”. To obtain a list of the most similar audio signals for a given query audio signal, we extract the embeddings for each audio signal using an embedding model of our choice. We then compute a cosine or L1 distance between our query audio signal and the signals in the database, returning a ranked list, where audio signals with higher similarity to the query audio are ranked higher.

The choice of audio embedding model can have a large impact on the results. There are a variety of embeddings capturing different features of audio, such as [1, 2, 28–30, 32, 33]. We focus on CLAP [2] and CLMR [1] embeddings for this work. Both are state-of-the-art (SOTA), produce human-validated similarity in our listening tests, are robust to perturbation, and are able to return relevant top songs.

3. DATA AND METHODOLOGY

3.1 Scope of Analysis

We want to create a system that identifies music both quantitatively similar and subjectively similar to humans. We do not focus on measuring similarity of any individual feature of music (e.g., timber, rhythm, lyrics), but rather use one of two embedding approaches (CLAP or CLMR) to encode audio and examine whether similarity in these embedding spaces aligns with human subjective evaluation (Sec. 4).

3.2 Data and Models Used

Though our approach is model agnostic, we validate our framework on VampNet [3], a generative model trained on 795k songs collected from the internet. VampNet takes a masked acoustic token modeling approach to music audio generation. In the first stage, a Descript Audio Codec (DAC) [31] learns to encode the audio data in a discrete vocabulary of “tokens”, of which it is then trained to model sequences. To create audio files, the token sequence is converted back into the input domain via the DAC decoder. VampNet adopts a masked generative modeling approach with a parallel iterative decoding procedure. Conditioning is done through example audio, either with a prefix (generating a continuation), postfix (generating an introduction) or as infill (masking the middle). We denote musical outputs of VampNet as “vamps.”

We chose VampNet because, at the time of writing, no other model made available both the training data and model weights. We trained another version of VampNet on a smaller training data and found no noticeable differences. While VampNet has a diverse set of music for the training set, our companion website also includes a small analysis of efficacy on various genres in the GTZAN [34] dataset.

3.3 Methodology

3.3.1 Similarity Metric

We define a measure of approximate memorization in generative audio models by establishing a threshold for high similarity and memorization of training data against a large collection of 5 million 3-second song clips drawn from the 795k songs in VampNet’s [3] training data. We take inspiration from the “split-product” measure for image similarity [4, 20], which breaks the images into smaller chunks to compare inner products of corresponding localized features. In our work, every audio file is split into 3-second clips, each of which is encoded as a feature vector. We measure the cosine similarity between generated clips and training data clips to find similarity between sub-portions of songs, returning the most similar songs.

3.3.2 Embeddings

We focus on two main embeddings: (1) contrastive learning of musical representations (CLMR) embeddings [1] and (2) contrastive language-audio pretraining (CLAP) embeddings [2]. We use both CLAP and CLMR embeddings because they can be applied to any dataset of raw music audio without the need for any transformation or fine-tuning, generalize well to out-of-domain datasets, and can be used as a baseline across different models and genres. Utilizing publicly available embeddings that generalize to any dataset is helpful in encouraging adoption.

We put all of the embeddings and their corresponding musical metadata in a vector database (Pinecone) that lets us quickly and efficiently search through millions of embeddings and return the top k similar songs by a chosen similarity metric (e.g., cosine similarity) in milliseconds.

3.3.3 Code and Tools Used

To recreate this study, use the following code and tools. To generate audio using VampNet: github.com/hugofloresgarcia/vampnet. To put audio in a format suitable for Pinecone and to add noise to clips (see Section 5.1) use: github.com/julbarnett/exploring-musical-roots.

4. LISTENING TEST: EXPERIMENTAL DESIGN

Presumably, output that is highly similar to a training audio clip was influenced by that clip. Of course, similarity is in the ear of the listener and many similarity measures do not align with human opinions. To build a replicable framework that will not require other audio researchers to conduct costly and cumbersome human listening tests, we conduct an experiment with human listeners to demonstrate the alignment of our quantitative technique with hu-

man listening. We utilize ReSEval, a framework that enables us to build subjective evaluation of audio tasks deployed on crowdworker platforms [35].

4.1 Dataset Preparation

To create the data for our study we take a random sample of 1,000 3-second clips from VampNet’s training data. For each of these 1,000 clips, we rank its top 10,000 closest clips in the training dataset by cosine similarity. For each embedding (CLAP and CLMR), we fit a Gaussian to the distribution of similarity scores of the top 10,000 clips (histograms in Table 1). The further above the mean a similarity score is, the more similar the clips are. We segment the data into 4 meaningful bins: the mean cosine similarity of the top 10,000 (CLAP: 0.815; CLMR: 0.693), $+1\sigma$ (CLAP: 0.885; CLMR: 0.784), $+2\sigma$ (CLAP: 0.955; CLMR: 0.875) and “random” (CLAP: 0.513; CLMR: 0.151). For the random bin, we take two random sets of 1,000 clips from the full 5 million clip dataset and measure pairwise cosine similarity; the mean similarity of this distribution gives the expected similarity score of random song pairs. We use these bin centers to create bins ± 0.02 for these similarity scores.

4.2 ABX Trials

Cartwright et al. [36, 37] overcame the difficulties of deploying time-consuming lab-based listener studies by utilizing pairwise comparison performed over the web, duplicating the findings of a lab-based test. We leverage these findings and employ a pairwise comparison study design, performing the study on Mechanical Turk (MTurk).

In our study, listeners are asked to perform ABX trials. The target audio clip (X) is presented, along with two other clips (A and B). The listener is asked to rate which clip (A or B) is more like the target X. The proportion of listeners that find A more similar than B is an estimate of the probability that people find A more similar to X than B.

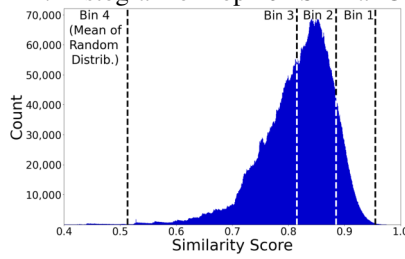
Given a clip X drawn from a random sample of 1,000 clips, one can then create a pair of examples A and B by selecting them randomly from different bins (see Section 4.1). This lets us create ABX trials with known differences in cosine similarity to X between the paired examples A and B. We can then collect statistics on the probability that users will find A more similar to X than B is to X. The greater the difference in cosine similarity, the more skewed we expect the listening results to be. If true, our objective measure’s similarity rankings align with human rankings.

We have 4 bins, resulting in 6 different pair-wise comparisons (bin 1 vs. bin 2, 1v3, 1v4, 2v3, 2v4, and 3v4). To have 150 evaluations per bin (900 evaluations total), we need 90 people to listen to 10 ABX comparisons each. We randomly choose 15 prompt “X” clips from the training data, with their respective 4 clips within the bins chosen as detailed above for the A and B comparison. An example set of clips for an ABX evaluation is at tinyurl.com/exploring-musical-roots.

4.3 Participant Recruitment

We utilized MTurk to recruit 150 participants each to evaluate similarity scores of CLAP and CLMR embeddings.

Human Evaluation Results: ABX Listening Test

		CLAP Embeddings			
		Bin 2	Bin 3	Bin 4	Total
		(0.885 ±0.02)	(0.815 ±0.02)	(0.513 ±0.02)	(All Trials)
CLAP: Histogram of Top 10k Similar Clips 	B				
	A				
	Bin 1	96.2%	98.0%	98.1%	97.4%
	(0.955 ±0.02)	(n = 156)	(n = 150)	(n = 162)	(n = 468)
Bin 2		73.3%	93.6%	83.7%	
(0.885 ±0.02)		(n = 135)	(n = 141)	(n = 276)	
Bin 3			81.5%	81.5%	
(0.815 ±0.02)			(n = 178)	(n = 178)	

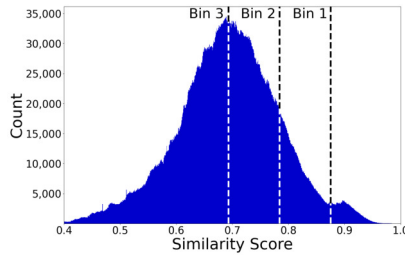
		CLMR Embeddings			
		Bin 2	Bin 3	Bin 4	Total
		(0.784 ±0.02)	(0.693 ±0.02)	(0.151 ±0.02)	(All Trials)
CLMR: Histogram of Top 10k Similar Clips 	B				
	A				
	Bin 1	90.7%	91.0%	98.5%	93.2%
	(0.875 ±0.02)	(n = 150)	(n = 156)	(n = 135)	(n = 441)
Bin 2		71.6%	93.6%	82.7%	
(0.784 ±0.02)		(n = 155)	(n = 157)	(n = 312)	
Bin 3			80.7%	80.7%	
(0.693 ±0.02)			(n = 140)	(n = 140)	

Table 1. Results from the listening experiment. Results show the percent of time listeners rated clip “A” (the clips with higher similarity scores to the prompt “X”) as more similar to the prompt clip “X” than clip “B” (those with lower similarity scores to prompt “X”). Histograms of the top 10k similar songs can be found to the left of the table. Bin regions are shown on these histograms. Bin 3 is centered on the mean of the top 10,000 most similar clips, Bin 2 = +1σ, Bin 1 = +2σ, and Bin 4 is the mean similarity score of a randomly selected clip from the entire training data (not just the top 10k).

We paid each evaluator \$1.50 to annotate 1 set of 10 ABX trials (estimated \$22.50/hour). We recruited US residents with an approval rating of at least 98 and 1,000 approved tasks. We filtered out bots by excluding evaluations that failed a pre-screening listening test. There were no requirements for music expertise beyond passing a listening test.

4.4 Results

Table 1 contains the results of our listening experiment. We found that human evaluations closely aligned with our quantitative metrics. For both CLAP and CLMR evaluations, listeners affirm by a wide margin that clips with higher similarity scores (lower bin numbers) sound more similar to the prompt clip than those with lower scores (higher bin numbers). Clips drawn from the most-similar bin (Bin 1) to the prompt track “X” were rated as more similar to the prompt clip than clips from any other bin 97.4% of the time for CLAP (93.2% for CLMR). For both embeddings, the vast majority of listeners ranked the clips with high similarity to the prompt track (“A”: Bins 1-3) as sounding more similar than the random song (“B”: Bin 4).

5. ANALYSIS OF OBJECTIVE MEASURES

5.1 Robustness to Perturbations

Our second research question focuses on the effect of different perturbations on our methodology’s ability to correctly return similar songs. Any generative music model will add some degree of variation to a training example during the generation process—the aim of these models is

not to replicate the training data exactly. This variation could take many forms (e.g., changing the pitch, speed). Therefore, in this section we evaluate the ability of our methodology to return target songs that have been modified by given perturbations. For varying amounts of each perturbation, we evaluate how frequently the target song (the unmodified clip) is returned as the most similar, within the top 5 similar songs, and within the top 10 most similar songs. The 7 types of perturbations we evaluate are:

- **Pitch shift** (in semitones; range: -12 to 12)
- **Time stretch** (in % of song; range: -20% to +20%)
- **White noise** overlaid on top of music (in dB; range: -30 to 30 dB in relation to original audio clip)
- **“Mash-up”** of two clips from training data (range: 5/95% to 95/5%; e.g., 60/40%)
- **“Mash-up”** of one clip from inside and one outside training data (range: 5/95% to 95/5%; e.g., 60/40%)
- **“Mash-up”** of a prompt clip and the generated vamp (range: 5/95% to 95/5%; e.g., 60/40%)

We selected these because we envision them as common alterations to music that would not render it unrecognizable by a human listener. We are not seeking to evaluate all types of adversarial noise since we are assuming users and creators are working cooperatively with these generative models to create something novel—not acting maliciously.

We evaluate all of the audio perturbations for both CLAP and CLMR embeddings to understand the robustness of our methodology while utilizing different embedding networks. For all perturbations except higher levels

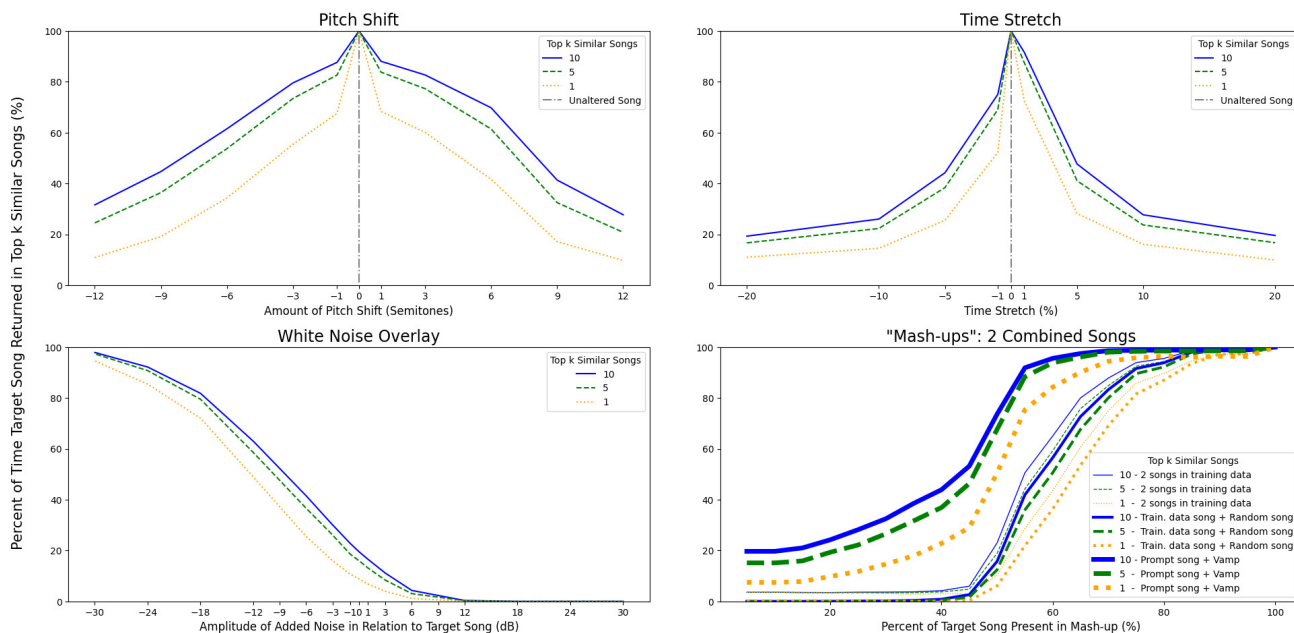


Figure 1. Plots of various amounts of noise perturbations to clips and the percent of the time they were returned in the top $k = 10$, $k = 5$, and $k = 1$ song using our methodology for CLMR embeddings. Displays pitch shift in semitones, time stretch as percent shortened/elongated, white noise overlay in decibels to target clip, and mash-ups of 2 songs in training data, 1 song in training data and one random, and a prompt song and its generated vamp.

of time stretch, CLMR embeddings are more robust than CLAP embeddings; all of the results using CLMR are presented in Figure 1. Example perturbations available at tinyurl.com/exploring-musical-roots.

Pitch shift is a common perturbation to audio that involves raising or lowering the original pitch of an audio clip without adjusting the length of the clip. Notably, human perception is extremely robust to pitch shift. Both embedding types were robust to small pitch shifts; for changes of ± 3 semitones the target song was returned the vast majority of the time. Both embedding types had a lower recall of the target song for larger pitch shifts.

Time stretching audio clips involves speeding up or slowing down audio while keeping the pitch constant. For this perturbation, we evaluate stretching the clip from 20% slower to 20% faster. Both embeddings consistently returned the target song for small amounts of time stretch, but were impacted by larger amounts ($> \pm 10\%$).

White noise overlay involves adding randomly generated white noise to audio clips. We evaluate the noise level in relation to the amplitude of the original clip in decibels, ranging from -30 to 30dB (-30dB being the quietest). Though we were only able to consistently return the target song at quiet levels of white noise overlay (≤ -18 dB) barely perceptible to the human ear, this perturbation has the largest impact on our method’s ability to identify the target track. Luckily, this is not an anticipated type of noise; generative models will add more “musical” variation to songs rather than white noise.

“Mash-ups” of two combined songs are defined here as splicing two clips together at different percentage levels (e.g., for 75/25% the first 2.25 seconds are the target song

and the last 0.75 seconds are some other song). We evaluate three types of “mash-ups”: combining (1) two clips from the training data, (2) one clip from the training data and one outside of the training data, and (3) a prompt track and its generated vamp from VampNet. For each mash-up, we seek to identify the percent of time the target (or prompt) track is returned in the top similar songs. CLMR embeddings only need 50-60% of the target song present in the mash-up to consistently return it in the top similar songs (CLAP need $\geq 80\%$). At each mash-up proportion the model returned the target song (prompt song) for mash-ups with vamps more consistently than for combining two different songs, indicating the vamp is more similar to the prompt song than two randomly selected songs are to each other. When the majority of the song analyzed is the vamp (i.e., $x\text{-axis} \leq 50\%$), it does not return the target (prompt) but rather other songs in the training data.

5.2 Systematic Evaluation of Generative Music Model

As a case study, we systematically evaluate VampNet [3] to demonstrate how to employ this technique to understand training data attribution on both individual songs and an entire model. To evaluate VampNet, we generate 10,000 vamps from 1,000 10-second prompt clips (10 different vamps per clip), and evaluate the most similar clips in the training data to the vamps. We embed each of the 10,000 vamps as a feature vector using both CLMR and CLAP embeddings and analyze the most similar 50 clips by cosine similarity (out of the five million+ clips from VampNet’s training data in our vector store). For each of the 10,000 vamps, the prompt that generated the vamp was rarely among the top similar clips returned by our methodology. Thus, we seek to understand the attribution of the

Systematic Evaluation of Generated Music (Vamps)			
Similarity Score		Vamp & Prompt	Vamp & #1 Similar Track
CLAP	Mean	0.393	0.795
	Median	0.402	0.815
	St. Dev.	0.151	0.084
CLMR	Mean	0.166	0.846
	Median	0.153	0.850
	St. Dev.	0.189	0.054

Table 2. Systematic evaluation of VampNet’s generations. Generated pieces of music (vamps) are less similar to the prompt song provided to the model at generation time than they are to other music from the training data.

rest of the training data on generations. For CLAP embeddings, the average cosine similarity between a prompt clip and generated vamp was 0.393, ($\sigma = 0.151$), whereas on average, the closest clip had a similarity score of 0.795 ($\sigma = 0.084$). CLMR had a similar disparity; full descriptive results are in Table 2. As noted in the above analysis on robustness to perturbations in Section 5.1, our methodology utilizing CLMR (rather than CLAP) embeddings is more robust to perturbations combining elements of new clips with clips present in the training data. Thus for the remainder of this section we will focus on CLMR embeddings for this case study using VampNet.

Leveraging insight from our listening study (Section 4), human evaluation affirms that within CLMR embeddings, music clips with a similarity score of ≥ 0.875 sound significantly more similar than clips with lower similarity scores. For this analysis, we utilize that same top bin as a benchmark and evaluate how often the most similar clips have similarity scores ≥ 0.875 . Findings are presented in Table 3. Over 30% of the vamps generated had at least one song with a similarity score ≥ 0.875 . Looking at scores in 0.02 increments above this benchmark similarity score, almost 20% of vamps had at least one song in the training data with a similarity score ≥ 0.895 , 9% ≥ 0.915 , 3% ≥ 0.935 , and almost 1% ≥ 0.955 . Evaluating more broadly among the top 10 songs, songs with these high similarity scores were concentrated among most similar couple clips, as opposed to having the entirety of the top 10 most similar clips have extremely high similarity scores. This indicates that at least 30% of the time, small sets of songs from the training data were highly influential on generated vamps.

6. DISCUSSION

These findings establish that the framework we propose is an effective means to systematically evaluate the training data attribution on any generative music model. This method is replicable and should be employed by model creators so they are able to have a greater understanding of their outputs. If exposed to end users, this framework also enables anyone to verify if they are copying music and learn about influences of their “novel” generations.

The authors first acknowledge the limitations of this ap-

Vamps with Highly Similar Songs in Training Data				
		Count of Songs in Top k		
		$k = 1$		$k = 10$
Similarity Score	k -clips ($n = 10,000$)	% Total		k -clips ($n = 100,000$)
		$k = 1$	$k = 10$	
≥ 0.955	89	0.89%	254	0.25%
≥ 0.935	317	3.17%	1,223	1.22%
≥ 0.915	924	9.24%	3,139	3.14%
≥ 0.895	1,929	19.29%	8,786	8.79%
≥ 0.875	3,201	32.01%	17,291	17.29%

Table 3. For 10,000 vamps, displays how many top k most similar training data songs were at or above given similarity scores for CLMR embeddings. The lowest similarity score in this table (0.875) corresponds to the highest benchmark (Bin 1) from the human listening test (Sec 4).

proach. First, the scope is intentionally limited to exclude lyrics. As generative music models continue to progress this can become an important area of memorization and copyright infringement, and we encourage future research to examine lyric memorization in tandem with our approach. Our scope also did not include any individualized feature levers for similarity (e.g., timbre or rhythm). We did this to both focus on a low-burden implementation for model creators who would follow this methodology as well as to identify encompassing interacting similarities without isolating any musical feature. However, these could be useful for both model creators and users.

Two potential harms of generative audio models are cultural appropriation and copyright infringement [38]. Our work aims to combat these issues both at the time of output generation and prior to model release. Our method can prevent cultural appropriation by giving users the opportunity to engage with the influences of the music, and prevent copyright infringement if the user realizes the generated piece of music is too similar to the identified influences.

7. CONCLUSION

We have proposed an easily-implementable framework for creators of generative music models to evaluate training data attribution. It can be used to prevent appropriation, copyright infringement, and otherwise uninformed creations, enabling model creators and users to understand the influences on their generated outputs by identifying similar songs in the training data. We evaluated a measure of cosine similarity for two embeddings and verified that they align with human perception with a subjective listening test. We also evaluated how robust our framework is to various forms of perturbations we anticipate models adding to training data during the transformation to “novel” output. We perform a case study on VampNet [3] in order to validate the efficacy of our framework. This work is a step towards transforming a generative model from a crutch replacing artistic knowledge to a tool creators and users alike can use to become better and more informed artists.

8. RESEARCH ETHICS AND SOCIAL IMPACT

The authors of this paper took the ethical considerations and social impact of this work seriously. A recent exhaustive study of the ethical implications of generative audio models [38] found that less than 10% of research on generative audio models published discussed any sort of potential negative impact of their work. We took that as inspiration to center our work around the ethical concerns and attempt to build a bridge between ethicists and generative audio engineers.

As mentioned in the discussion (Section 6), among the negative impacts uncovered for generative music models were the potential for cultural appropriation, copyright infringement, and loss of agency and authorship of the creators. This work aims to combat these issues at the time of generation, on a track by track level. By uncovering the roots of a given piece of generated music, we can empower the user of the model to understand where the music came from and learn about the influences.

A primary concern the authors have for this work is that future model creators will simply use this framework as a checkbox to complete their ethical evaluations. They may use this framework and assume since they did so, there are no other potential societal impacts or ethical harms to consider in regard to generative music models. This work only tackles a portion of the issues, and is only a first step in doing so. Though our method can highlight instances of copyright infringement and cultural appropriation, it by no means will catch everything. Though this can assist with educating users about the influences of their work, it will not solve the potential loss of agency and authorship users and musicians could feel when using these models. It does nothing to address creativity stifling, predominance of western bias, overuse of publicly available data, non-consensual use of training data, or job displacement and unemployment. It also requires energy consumption to generate the embeddings and perform searches, so it contributes to the issue of energy consumption of generative models rather than combating it.

In regard to the experiment utilizing human evaluators to subjectively analyze similar pieces of music, we ensured that our study was in line with institutional review board standards (our study was determined to be exempt). We had a thorough consent form for the crowdworkers and ensured they knew they could quit at anytime without any sort of penalty. We timed ourselves taking the survey and attempted to pay them a fair wage (estimated \$22.50 per hour, higher than any minimum wage in the United States). We even paid users who failed the listening pre-screening test for their time and thus were not able to take our survey, even though they did not contribute data to our study. However, we acknowledge that ethical crowdsourcing goes beyond fair pay [39, 40], and tested the listening test thoroughly prior to launch to be certain there would be no burden to crowdworkers beyond potential boredom. The most sensitive data we had access to were the Mechanical Turk IDs of users, but we held these on secure servers.

The authors determined that the positive impact of this

work outweighed these potential harms, especially since the primary motivation of this work is to address a few existing ethical issues in generative audio. However, it is essential to acknowledge these potential risks and where our method falls short.

9. ACKNOWLEDGMENTS

The authors would like to thank the reviewers for their helpful feedback on this work, as well as Max Morrison for his help with the use of Reproducible Subjective Evaluation (ReSEval) [35]. This research is partially supported by USA National Science Foundation award 2222369.

10. REFERENCES

- [1] J. Spijkervet and J. A. Burgoyne, “Contrastive learning of musical representations,” *arXiv preprint arXiv:2103.09410*, 2021.
- [2] Y. Wu, K. Chen, T. Zhang, Y. Hui, T. Berg-Kirkpatrick, and S. Dubnov, “Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation,” in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.
- [3] H. F. Garcia, P. Seetharaman, R. Kumar, and B. Pardo, “Vampnet: Music generation via masked acoustic token modeling,” *arXiv preprint arXiv:2307.04686*, 2023.
- [4] G. Somepalli, V. Singla, M. Goldblum, J. Geiping, and T. Goldstein, “Diffusion art or digital forgery? investigating data replication in diffusion models,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 6048–6058.
- [5] Z. Borsos, R. Marinier, D. Vincent, E. Kharitonov, O. Pietquin, M. Sharifi, D. Roblek, O. Teboul, D. Grangier, M. Tagliasacchi *et al.*, “Audiolm: a language modeling approach to audio generation,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.
- [6] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, and I. Sutskever, “Jukebox: A generative model for music,” *arXiv preprint arXiv:2005.00341*, 2020.
- [7] A. Agostinelli, T. I. Denk, Z. Borsos, J. Engel, M. Verzetti, A. Caillon, Q. Huang, A. Jansen, A. Roberts, M. Tagliasacchi *et al.*, “Musiclm: Generating music from text,” *arXiv preprint arXiv:2301.11325*, 2023.
- [8] Z. Borsos, M. Sharifi, D. Vincent, E. Kharitonov, N. Zeghidour, and M. Tagliasacchi, “Soundstorm: Efficient parallel audio generation,” *arXiv preprint arXiv:2305.09636*, 2023.

- [9] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Défossez, “Simple and controllable music generation,” *arXiv preprint arXiv:2306.05284*, 2023.
- [10] J. D. Parker, J. Spijkervet, K. Kosta, F. Yesiler, B. Kuznetsov, J.-C. Wang, M. Avent, J. Chen, and D. Le, “Stemgen: A music generation model that listens,” 2023.
- [11] N. Carlini, C. Liu, Ú. Erlingsson, J. Kos, and D. Song, “The secret sharer: Evaluating and testing unintended memorization in neural networks,” in *28th USENIX Security Symposium (USENIX Security 19)*, 2019, pp. 267–284.
- [12] V. Feldman and C. Zhang, “What neural networks memorize and why: Discovering the long tail via influence estimation,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 2881–2891, 2020.
- [13] N. Carlini, F. Tramer, E. Wallace, M. Jagielski, A. Herbert-Voss, K. Lee, A. Roberts, T. Brown, D. Song, U. Erlingsson *et al.*, “Extracting training data from large language models,” in *30th USENIX Security Symposium (USENIX Security 21)*, 2021, pp. 2633–2650.
- [14] H. Hu, Z. Salcic, L. Sun, G. Dobbie, P. S. Yu, and X. Zhang, “Membership inference attacks on machine learning: A survey,” *ACM Computing Surveys (CSUR)*, vol. 54, no. 11s, pp. 1–37, 2022.
- [15] Z. Peng, Z. Wang, and D. Deng, “Near-duplicate sequence search at scale for large language model memorization evaluation,” *Proceedings of the ACM on Management of Data*, vol. 1, no. 2, pp. 1–18, 2023.
- [16] A. de Wynter, X. Wang, A. Sokolov, Q. Gu, and S.-Q. Chen, “An evaluation on large language model outputs: Discourse and memorization,” *arXiv preprint arXiv:2304.08637*, 2023.
- [17] K. Tirumala, A. Markosyan, L. Zettlemoyer, and A. Aghajanyan, “Memorization without overfitting: Analyzing the training dynamics of large language models,” *Advances in Neural Information Processing Systems*, vol. 35, pp. 38 274–38 290, 2022.
- [18] S. Biderman, U. S. Prashanth, L. Sutawika, H. Schoelkopf, Q. Anthony, S. Purohit, and E. Raf, “Emergent and predictable memorization in large language models,” *arXiv preprint arXiv:2304.11158*, 2023.
- [19] N. Carlini, D. Ippolito, M. Jagielski, K. Lee, F. Tramer, and C. Zhang, “Quantifying memorization across neural language models,” *arXiv preprint arXiv:2202.07646*, 2022.
- [20] N. Carlini, J. Hayes, M. Nasr, M. Jagielski, V. Shwag, F. Tramer, B. Balle, D. Ippolito, and E. Wallace, “Extracting training data from diffusion models,” *arXiv preprint arXiv:2301.13188*, 2023.
- [21] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, “High-resolution image synthesis with latent diffusion models,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695.
- [22] P. Cano, E. Batlle, T. Kalker, and J. Haitsma, “A review of audio fingerprinting,” *Journal of VLSI signal processing systems for signal, image and video technology*, vol. 41, pp. 271–284, 2005.
- [23] J. Haitsma and T. Kalker, “A highly robust audio fingerprinting system,” in *Ismir*, vol. 2002, 2002, pp. 107–115.
- [24] A. Wang, “The shazam music recognition service,” *Communications of the ACM*, vol. 49, no. 8, pp. 44–48, 2006.
- [25] A. Wang *et al.*, “An industrial strength audio search algorithm,” in *Ismir*, vol. 2003. Washington, DC, 2003, pp. 7–13.
- [26] K. Chen, Y. Wu, H. Liu, M. Nezhurina, T. Berg-Kirkpatrick, and S. Dubnov, “Musicldm: Enhancing novelty in text-to-music generation using beat-synchronous mixup strategies,” *arXiv preprint arXiv:2308.01546*, 2023.
- [27] D. Bralios, G. Wichern, F. G. Germain, Z. Pan, S. Khurana, C. Hori, and J. L. Roux, “Generation or replication: Auscultating audio latent diffusion models,” *arXiv preprint arXiv:2310.10604*, 2023.
- [28] R. Castellon, C. Donahue, and P. Liang, “Codified audio language modeling learns useful representations for music information retrieval,” *arXiv preprint arXiv:2107.05677*, 2021.
- [29] B. Kim and B. Pardo, “Improving content-based audio retrieval by vocal imitation feedback,” in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 4100–4104.
- [30] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, “High fidelity neural audio compression,” *arXiv preprint arXiv:2210.13438*, 2022.
- [31] R. Kumar, P. Seetharaman, A. Luebs, I. Kumar, and K. Kumar, “High-fidelity audio compression with improved rvqgan,” *arXiv preprint arXiv:2306.06546*, 2023.
- [32] Y. Li, R. Yuan, G. Zhang, Y. Ma, C. Lin, X. Chen, A. Ragni, H. Yin, Z. Hu, H. He *et al.*, “Map-music2vec: A simple and effective baseline for self-supervised music audio representation learning,” *arXiv preprint arXiv:2212.02508*, 2022.
- [33] A. L. Cramer, H.-H. Wu, J. Salamon, and J. P. Bello, “Look, listen, and learn more: Design choices for deep

- audio embeddings,” in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 3852–3856.
- [34] B. L. Sturm, “The gtzan dataset: Its contents, its faults, their effects on evaluation, and its future use,” *arXiv preprint arXiv:1306.1461*, 2013.
- [35] M. Morrison, B. Tang, G. Tan, and B. Pardo, “Reproducible subjective evaluation,” in *ICLR Workshop on ML Evaluation Standards*, April 2022.
- [36] M. Cartwright, B. Pardo, and G. J. Mysore, “Crowdsourced pairwise-comparison for source separation evaluation,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 606–610.
- [37] M. Cartwright, B. Pardo, G. J. Mysore, and M. Hoffman, “Fast and easy crowdsourced perceptual audio evaluation,” in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 619–623.
- [38] J. Barnett, “The ethical implications of generative audio models: A systematic literature review,” in *Proceedings of the 2023 AAAI/ACM Conference on AI, Ethics, and Society*, 2023, pp. 146–161.
- [39] B. Shmueli, J. Fell, S. Ray, and L.-W. Ku, “Beyond fair pay: Ethical implications of nlp crowdsourcing,” *arXiv preprint arXiv:2104.10097*, 2021.
- [40] D. Schlagwein, D. Cecez-Kecmanovic, and B. Hanckel, “Ethical norms and issues in crowdsourcing practices: A habermasian analysis,” *Information Systems Journal*, vol. 29, no. 4, pp. 811–837, 2019.

Papers – Session III

GREEN MIR? INVESTIGATING COMPUTATIONAL COST OF RECENT MUSIC-AI RESEARCH IN ISMIR

Andre Holzapfel

Anna-Kaisa Kaila

Petra Jääskeläinen

Division of Media Technology and Interaction Design, KTH Royal Institute of Technology, Sweden

holzap@kth.se, akkaila@kth.se, ppja@kth.se

ABSTRACT

The environmental footprint of Generative AI and other Deep Learning (DL) technologies is increasing. To understand the scale of the problem and to identify solutions for avoiding excessive energy use in DL research at communities such as ISMIR, more knowledge is needed of the current energy cost of the undertaken research. In this paper, we provide a scoping inquiry of how the ISMIR research concerning automatic music generation (AMG) and computing-heavy music analysis currently discloses information related to environmental impact. We present a study based on two corpora that document 1) ISMIR papers published in the years 2017–2023 that introduce an AMG model, and 2) ISMIR papers from the years 2022–2023 that propose music analysis models and include heavy computations with GPUs. Our study demonstrates a lack of transparency in model training documentation. It provides the first estimates of energy consumption related to model training at ISMIR, as a baseline for making more systematic estimates about the energy footprint of the ISMIR conference in relation to other machine learning events. Furthermore, we map the geographical distribution of generative model contributions and discuss the corporate role in the funding and model choices in this body of work.

1. INTRODUCTION

Interest in AMG and DL-based analytical models is increasing dramatically at conferences such as ISMIR [1]. Case studies in domains other than music [2–4] have established that the environmental impact of AI technologies can be massive, particularly when it comes to energy consumption. International Energy Agency predicts that the accumulated electricity consumption of data centers, AI, and the cryptocurrency sector will double, reaching the level of whole electricity consumption of Japan by 2026 [5]. In the US, a recent proposal for legislation (Artificial Intelligence Environmental Impacts Act) suggests that AI companies would be urged to start reporting the

environmental impacts of their work [6]. With the increasing investment in AI and the general trend of high compute requirements for training state-of-the-art machine learning systems [7], we expect to see the accumulated energy footprint of the generative music industry and the surrounding research also growing. There is no reason why research around ISMIR would be isolated from these effects. For the community to gain an understanding of the scale of the problem and to identify solutions to avoid excessive energy use in AMG development, more knowledge is needed of the current energy cost of the research conducted. It is, hence, highly relevant and timely to investigate to what extent research at ISMIR acknowledges and documents the environmental impact of energy consumption.

Other research communities around music technology (e.g., NIME [8]) and machine learning technology (e.g., NeurIPS [9]) have shown increasing attention to various aspects of negative ethical impacts, among them environmental. Discussions of such topics continue, however, to be severely underrepresented in the generative music and audio research [10], and entirely absent from the ethics principles and guidelines for AI-music [11, p. 148]. In the context of the ISMIR community, Morreale [12] estimated that between 2011 and 2020, less than 0.5% of ISMIR submissions discussed issues related generally to ethics, of which sustainability could be seen as a subcategory. Our present scoping inquiry demonstrates this lack of concern and transparency in reporting the environmental impacts of AMG and other DL research, with a focus on ISMIR conferences. The title of our paper refers to Schwartz et al. [13], which proposes the concept of *Green AI* as “AI research that is more environmentally friendly and inclusive”. While the concept of Green MIR should be used carefully, as it can lead to practices of greenwashing research, we use this term to raise questions about the current practices and energy impact of MIR.

This study advances a critical discussion in the ISMIR community around the ethical impacts of model development work and the responsibility of MIR research from the underexplored perspective of environmental sustainability. It documents, firstly, the level of transparency in reporting the environmental impact in terms of energy consumption and the computational resource use in the model training process in ISMIR papers in the seven years 2017–2023. Secondly, based on the information documented in these ISMIR publications, we provide preliminary estimates of the energy demands related to training individual AMG



© A. Holzapfel, A. Kaila, and P. Jääskeläinen. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** A. Holzapfel, A. Kaila, and P. Jääskeläinen, “Green MIR? Investigating computational cost of recent music-AI research in ISMIR”, in *Proc. of the 25th Int. Society for Music Information Retrieval Conf.*, San Francisco, United States, 2024.

and other DL models, as well as an overall estimate of the total energy use and carbon footprint associated with DL training at an ISMIR conference. In this process, we also investigate if AMG models are related to higher energy consumption than other models at ISMIR. Our calculations establish a baseline to set the ISMIR community in relation to other machine learning communities and call for the issue of environmental impact to be addressed more systematically so that the conference can grow and evolve without compromising the environment. Thirdly, we map the geographical distribution of generative model contributions and discuss the role of corporate participation and the consequent political economies in this body of work.

Some limitations of the paper lie within (a) the many uncertainties in estimating the overall energy consumption of developing a DL model based on limited data related to the training process, (b) our focus on model training that leaves a focus on inference processes to future work, and, (c), the reduction of environmental sustainability to energy consumption. In the following two sections, we explain how these limitations emerge from the sparse amount of information and the complexity of the problem. We hope that this paper will motivate both individual authors and the ISMIR conference to take action toward more minute documentation of resource use in DL model development.

2. BACKGROUND

2.1 Environmental Impact of (Music-)AI

The soaring energy cost of AI technology is increasingly discussed in the academic literature [2, 3, 13–19]. Early work on the environmental impact of AI has introduced concepts, such as “green” and its opposite, “red” AI [13], problematized hidden environmental costs in AI [2], and provided methods for quantifying environmental impact [3]. We found only a few works that are related to music-AI (as a term to cover analytic and generative approaches based on – predominantly – deep learning). The first two concern the energy cost of AI models used in music information retrieval [20, 21], and the third focuses on the importance of studying sustainability in arts generally [4].

AI development has been increasingly steering towards Large Language Models (LLMs), which have a particularly high energy expenditure. The popularity of these models is attributed to their success in “generalizing” and performing better in tasks that have traditionally required human labor. But as a consequence, many research papers [22, 23] take a pre-trained foundation model and adapt it. This results in a situation in which the energy cost can be estimated only partially, *i.e.* for the part that extended the pre-trained LLM. Furthermore, it raises questions about how research can account for the environmental cost of using LLMs. Arguably, the researchers who use those LLMs are somewhat responsible for the popularity and increased use of LLMs – through creating demand for their use – which can further aggravate the use of computational resources and energy in LLM development.

Many research works that focus on the environmental

impact of AI take the assumption that energy (computational) cost is the core environmental problem of these technologies, and by reducing energy consumption, it is possible to work towards sustainability. However, this is a simplistic view because sustainability is a complex phenomenon that does not only concern electricity usage. For example, Jääskeläinen et al. [24, 25] discussed the complex networks of environmental harm resulting from resource consumption and capitalistic colonialism that prevail in the case of generative AI. Strengers [26] has generally outlined how behavior change is central to change toward sustainability, and providing metrics such as energy consumption data is insufficient to address change toward sustainability. While keeping this in mind, energy use is a valid starting point for discussing the environmental impact of AI. In this paper, when we refer to *environmental impact*, we explicitly refer to the energy cost and leave out factors such as the life cycle of the technology and water usage of data centers [27], among others.

Technological advances in recent decades have entered the music industry with the promise of reduced material and energy demands. For instance, it was expected that the introduction of mp3, the digitalization of music production, and eventually the platformization of its consumption would diminish the environmental footprint of the industry. As Devine [28] and Brennan [29] have demonstrated, the opposite has historically been the case: while the demand for plastic dropped in the era of the mp3 to a fraction compared the previous music consumption models (CD, cassette, vinyl, etc.), the greenhouse gas emissions of the industry, on the contrary, *increased*. This increase was explained in sustainability research by Hilty’s concept *rebound effects* [30], which describes indirect 2nd and 3rd order effects that result from adopting new technology.

Similar negative effects are emerging in the proliferation of AI in music. Calculations by Holzapfel [31] illustrate that creative applications of LLMs can amount to considerable levels of energy demand. Furthermore, results by Douwes [20, 21] and Ronchini and Serizel [32] indicate that the scale of energy consumption for audio generation and analysis tasks are not in a linear relation with the model performance, thus questioning the assumption that the growth in model complexity and resource demands are a prerequisite for better models. Even more importantly, Holzapfel [31] calls for the focus of inquiry to be expanded from the carbon footprint to the wider questions of political ecology, and to the perspectives of economic gain and power (see also [33]). In this view, we should not only measure the environmental impact but also form a more complete picture by looking at who is causing it, who is financing that work, and who benefits from it.

2.2 Timeliness of Addressing the Environmental Impact of Music-AI

Bringing energy concerns into research practices is still at an early stage in many communities [19], including MIR. In 2023, Morreale et al. [1] ran a systematic survey of the training datasets for AMG models presented at ISMIR

2013–2023. Their work illustrates a dramatic increase in the development of AMG models in the last decade, and especially since 2017. Taken together with the general lack of both breadth and depth of addressing environmental concerns in music and audio research contexts [10], this increase highlights the urgency of addressing the computational cost of the AMG models in ISMIR research.

Conferences such as NIME have already taken a proactive lead in promoting awareness of the environmental impacts of the research conducted around the conference [34], and by making resources available [35] for the research community to adopt more environmentally conscious research and development practices. NeurIPS [36] requires authors to disclose information on the training procedure as well as the amount and type of compute resources used in the development and research of AI models.¹ We argue that such practices of accessible documentation should be part of the submission requirements in ISMIR research publications as well. This is useful for reproducibility and allows examining the energy cost that ISMIR research contributes to when developing AMGs and other models. However, in order to start such a discussion, it is essential to examine the current practices of reporting environmental impact-related information on AMG development at ISMIR.

3. METHOD

This study covers two corpora (total $N = 113$) of papers: 1) ISMIR papers published in the years 2017–2023 that introduce an AMG model, and 2) a complementary corpus of ISMIR papers from the years 2022–2023 that present analysis models and discuss processes that included heavy computations with GPUs. This will provide a perspective on training resource documentation in recent ISMIR conferences beyond AMG.

The **first corpus** was obtained by selecting papers that were specified as introducing an AMG model in the table compiled by Morreale et al. [1]. We extended this initial list by adding all papers from ISMIR 2023 that presented such a model in that year. This resulted in an overall list of 88 papers that present AMG models between 2013 and 2023. An analysis of the older papers revealed that the majority of papers published before 2017 did not involve DL models trained on GPUs, but rather shallow models (e.g., [38, 39]) or no training at all (e.g., [40, 41]). Therefore, we decided to exclude the 8 papers in the list by [1] published before 2017, resulting in 80 papers in this first corpus.

For each of these 80 papers, we documented whether there was information about the training time, whether the number of parameters was specified (search “param*”²), and whether the computational resources used for training were documented (search “GPU*”, “CPU*”, “TPU*”).

¹ Interestingly, the first editions (2021, 2022) of this checklist included a recommendation to use a CO2 emissions tracker [37], but this aspect has been omitted from the latest version of the guidelines.

² The asterisk character (*) is used to find all spelling variations of a search term, e.g. parameter, parameters, parametric etc.

We further searched the papers for discussions on energy consumption and environmental impact of the models, using terms “environment*”, “sustainab*”, “ecolog*”, “carbon”, “energy” and “kWh”. Finally, as an effort to connect these aspects to the wider perspectives of political ecology, we documented whether the paper indicated company connections in the author affiliations, whether funding information was included in the acknowledgments or elsewhere (search “fund*”, “support*”), whether there were indications of full or partial corporate funding, as well as which countries were the author affiliations related to. The full information retrieved is available in a published data table [42]. Whereas our analysis mainly focuses on the documentation of energy consumption, the additional information included in our data collection was intended to facilitate further contextualization and future research investigations.

To account for the most recent work at ISMIR in our **second corpus**, we searched the proceedings documents of 2022 and 2023 for the keywords “GPU*” and “TPU*”. We did not consider papers that discuss CPU usage in order to focus on DL models, and we excluded all papers that are already part of the first corpus. This way, we obtained a corpus of 33 papers that present models for analysis rather than generation, with some consideration of computational resources (15 papers from 2023, 18 papers from 2022). From the papers we obtained, we collected further information relating to training time, computational resources, and company connections. We focused on energy-related aspects in the second corpus in order to facilitate a comparison with AMG models.

For both corpora, the energy used in the training of a model was estimated for papers that provided information about the type and number of GPUs/TPUs used, along with the training time. We found that websites or GitHub sources did not add information for the vast majority of papers and, therefore, focused on information provided in the published papers. The Thermal Design Power (TDP) of each processor type was obtained from the datasheets of the manufacturer, and the energy used for a single training run was computed as the product of a number of processors, computing time in hours, and TDP. Using the TDP as a basis for energy consumption is a rather conservative estimate, as it ignores the energy consumption of the remaining computer hardware [19]. To take these factors into account, the use of tools (e.g. [3, 43]) to measure actual energy consumption during model development and the publication of this overall consumption would be required. We refrain from attempting to estimate the carbon emissions related to the computed energy consumption of individual papers because a reliable estimate would require detailed information about the energy sources used in the specific computation environment [43]. In our analysis, we do not consider the energy consumption related to model inference, but we will discuss insights related to the energy demands of inference.

Authors 1 and 2 collaborated on collecting data from both corpora by dividing the conference years between

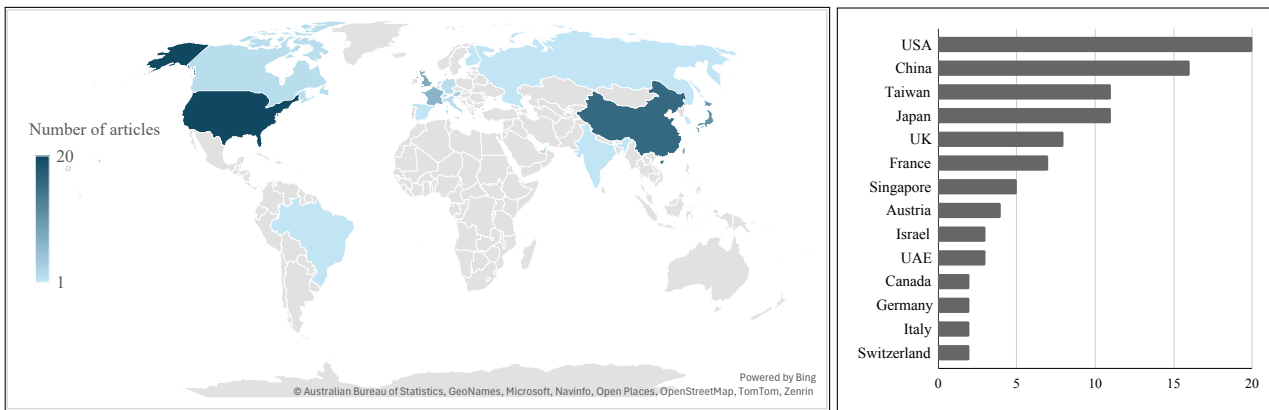


Figure 1. Geographical distribution of the authors in corpus 1 (N=80) at ISMIR 2017–2023, as indicated by the author affiliations. The block chart includes only countries from which at least two publications were found in the corpus.

them. The proceedings’ PDF files were searched for the above-listed terms, and the identified occurrences were analyzed manually without the use of scripts. Papers that presented unclear aspects in the data collection were flagged and discussed between both co-authors. Author 1 conducted the estimates of energy use for models in both corpora.

4. RESULTS

4.1 Results from Corpora 1 and 2

Our analysis revealed that 60 of the 80 papers (75%) in corpus 1 do not provide any information about the time and hardware required to train the model proposed in the paper. Of the remaining 20 papers, seven only provide information about the type and numbers of GPU but do not specify the time required for training. The remaining 13 papers provide full information about GPUs as well as training time. For corpus 2 (33 papers), we identified 13 papers that disclose full information about the computational hardware and training time, another 15 papers that provide partial compute information, and 5 papers that do not include such information. Overall, only 23 % of the papers in corpora 1 and 2 are fully transparent about the hardware and the model training.

When investigating potential change over time in corpus 1, we see that between the years 2017 and 2021, one or two papers annually disclose the full training and hardware data. In 2022, an exceptional six papers provided the full information (30% of the submissions analyzed for that year), whereas, in 2023, only three papers were partially transparent with the information about the GPU budgets. While this may indicate a general trend of increasing transparency in reporting the computational hardware and training cost of the AMG models, it would be misleading to claim this as the current norm in the ISMIR community. The lack of general reflection around the issues of environmental impact is furthermore evident from how the keywords “environment*”, “sustainab*”, “ecolog*”, “carbon”, “energ*”, and “kWh” were completely missing from the analyzed corpus (0 hits for proceedings of 2013–2023).

A few recent papers include reflections regarding increasing computational demands [44–46], but these reflections are motivated by the cost of computing and do not make a relation to environmental impact explicit.

It is also noteworthy how the increase in research engagement with AMG, as documented by Morreale at al. [1], coincides with corporate participation in these efforts. In the years 2020 until 2023, ca. 40 % of the papers in our corpus are co-authored by individuals with affiliations in private companies, compared to 27% in the years before that. This interestingly compares to the 27% of papers with industry-affiliated co-authors in our second corpus, *i.e.* papers that train models for non-generative purposes. Overall, these numbers suggest a certain focus of corporate interest on generative approaches.

Direct corporate funding of the research efforts is, however, rarely documented, with only four papers in the whole analyzed first corpus (N = 80) indicating either involvement of private funding or GPU support from NVIDIA. Overall, a slight majority of papers does not report any funding sources at all. The remainder refers mainly to public funding agencies, most likely as a response to the demands by the agencies for acknowledgment. This suggests an overall situation in which vested financial interests — by private and public stakeholders — are documented in a way that is not very transparent.

As shown in Figure 1, the majority of the AMG model development comes from researchers affiliated with institutions in the US (20 papers) or China (16), followed by Taiwan (11), Japan (11), and the UK (8). In total, these five countries account for over 60% of the ISMIR publications included in corpus 1. These numbers will gain significance in the context of carbon footprint estimates in the next section.

4.2 Energy Use Calculations

For the set of models from corpora 1 and 2 described above that reported the full details of the computational hardware and training time (N = 26), we conducted calculations on the estimated energy use based on the type and number of GPUs/TPUs, the reported training time, and the Thermal

Design Power (TDP) of each processor type, as provided by the manufacturers. The energy used for a single training run was consequently computed as the product of the number of processors, computing time in hours, and TDP. The results of this calculation for each model analyzed are shown in Table 1.

Based on our calculations, the mean/median amount required to train an ISMIR model (for either corpus 1 or 2) is about 224.8kWh (mean) and 18.46kWh (median). This amounts roughly to the energy demand of a single-person household for two months/three days in a Western country, such as Germany.³ As is evident from Table 1, there is no clear distinction between the energy use of generative or analytic models, which implies that the pursued MIR task may not be an important factor. Instead, the distribution of values is strongly focused around smaller values, and only four outlier models require an amount of energy that lies above the average of 225kWh (hence, the large difference between mean and median). Out of these, the three most energy-demanding models in terms of training come from large IT corporations. The amount of energy required to train the models provided in these papers sums to 5.11MWh, which is about 87% of the total energy demand related to all 26 papers with full resource disclosure. In total, taking into account the full range of energy requirements, the papers with industry-affiliated authors demand about 89% of the total resources related to all 26 full-disclosure papers. In contrast, industry-affiliated authors are found only in 40 out of the 113 papers (35%) in our two corpora.

As mentioned in Section 3, an estimate of the actual carbon footprint requires – among other aspects – detailed information about the data centers at which the computation takes place and their energy sources. Nevertheless, we will approach a preliminary estimate of the carbon footprint related to model training at the most recent ISMIR conference. We carefully checked all papers in ISMIR 2023 and determined the number of papers that train a machine learning model, resulting in 62 out of 104 papers (59.6%). We accommodate for the fact that a small amount of these papers train “shallow” machine learning models and use an estimate of 50% of ISMIR papers that train deep learning models in recent years. Assuming the median as the representative statistic for the average energy consumption for training a model, we arrive at an energy consumption of 18.46kWh * 52 papers = 959.92kWh.

Starting from this number, two further obstacles impede a reliable estimate of the carbon footprint: 1) In each paper, the model has not been trained only once, but the total development of the presented model will have required more energy. Strubell et al. [14] have documented how the process of fine-tuning a specific model exceeded the energy demand of one training run by 24 times, and that a whole R&D cycle is three orders more expensive than a single training run. Lacking more precise numbers, it seems, therefore, fair to assume that the actual energy con-

Article	Corpus	Energy cost
Hawthorne et al 2022 [47] ⁴	1	4 375 kWh
McCallum et al 2022 [44]	2	444 kWh
Toyama et al 2023 [48]	2	296 kWh
Sarkar et al 2022 [49]	2	240 kWh
Ma et al 2023 [50]	2	144 kWh
Alonso-Jiménez et al 2023 [51]	2	79 kWh
Perez et al 2023 [52]	2	36 kWh
Brunner 2018 [53]	1	33 kWh
Teng 2017 [54]	1	29 kWh
Di Giorgi et al 2022 [55]	2	24 kWh
Wu, Hsiao et al 2022 [56]	1	22 kWh
Zhao et al 2022b [57]	2	20 kWh
Donahue et al 2019 [58]	1	20 kWh
Donahue et al 2022 [59]	2	17 kWh
Yeh et al 2022 [60]	1	12 kWh
Wu, Chiu et al 2022 [61]	1	12 kWh
Singh et al 2022 [62]	2	10 kWh
Wei et al 2022 [46]	2	8 kWh
Wu & Yang 2020 [63]	1	6 kWh
Pasini & Schlüter 2022 [64] ⁵	1	6 kWh
Zhao et al 2022a [65]	1	4 kWh
Zhang et al 2022 [66]	1	3 kWh
Srivatsan & Berg-Kirkpatrick 2022 [67]	2	3 kWh
Mittal et al 2021 [68]	1	3 kWh
Foscarin et al 2023 [69]	2	0,3 kWh
Peracha 2020 [70]	1	0,2 kWh

Table 1. Energy cost of model training in corpora 1 (N=13) and 2 (N=13).

sumption related to a paper is at least that of fine-tuning an existing model. Hence, with a very conservative assumption of a factor of 20, we arrive at an estimate of $E_{est} = 19.20MWh$ for all model development related to a recent ISMIR conference.

The second obstacle is that the location of the data center at which computation took place is not documented. Therefore, we decided to use the countries of author affiliations as an indicator of where computation took place. In terms of carbon footprint, this has an impact as the USA and China are both on the high end of the carbon intensity spectrum [19]. We retrieved the average carbon intensity of the grids in 2022⁶ for each country depicted in Figure 1, I_c (in gCO₂eq/kWh) and computed the estimate for the total carbon footprint C_{total} of one conference as

$$C_{total} = (E_{est}/N_{total}) \cdot \sum_{c \in C} N_c \cdot I_c \quad (1)$$

with N_c being the number of times co-authors were from

⁴ For the four models in this paper, only the minimum and maximum training times were specified. We use the mean of these two values as an estimate.

⁵ Full compute info for one of the included models only.

⁶ <https://ember-climate.org/data-catalogue/yearly-electricity-data/>

³ 5.77kWh per day for a one-person household in 2021 in Germany according to www.destatis.de.

a specific country out of the set C of all countries as depicted in Figure 1, $N_{total} = 96$ is the total count of the histogram. This results in an estimate of $C_{total} = 7.593$ tons of carbon dioxide from training processes related to one recent ISMIR conference.

Putting this number into context, according to the estimates by [43], the training of GPT-3 has caused energy consumption of about 189 MWh. With the carbon intensity of the USA in 2017 (higher than in 2022) of 449.06 gCO₂eq/kWh, this has produced 85 tons of carbon dioxide, one order larger than our estimate for the whole of ISMIR.

5. DISCUSSION

While this paper focused on the training phase of music-AI models, more information is needed about the energy consumption along the full pipeline of model development, inference⁷, and deployment. To this end, authors of ISMIR papers should – at the very least – clearly document the resources (compute time; type and number of processors) needed for training and inference, and – ideally – include more minute documentation of actual energy use during the whole development cycle. We encourage a discussion to adopt standards similar to NeurIPS within the ISMIR submission process.

A commonly used framework that can guide the direction towards considering the environmental impact of ISMIR in a broader sense can be found in the concept of planetary boundaries [71]. There are nine planetary boundaries that can help us to understand and analyze how our actions might influence the environmental systems. These include, for example, biodiversity loss and species extinction, stratospheric ozone depletion, ocean acidification, land-system change/deforestation, freshwater use, and atmospheric aerosol load. Taking the example of freshwater use, these dimensions can be directly applied to ISMIR research to examine the environmental impact in relation to the planetary boundaries. Efforts can be directed toward questions such as what is the level of water use for hardware cooling in computational tasks at ISMIR, and whether the life cycles of the used hardware are contributing to environmental processes such as ocean acidification or species extinction. Unfortunately, six of nine planetary boundaries are currently transgressed [72], and that places us on track for increased climate change and breakage of the prevailing ecosystems.

While energy estimates provide a baseline for understanding the scale of the specific issue of energy consumption and for comparing individual model types to one another, they are not in and of themselves a sufficient solution to the problem of environmental sustainability in model development at ISMIR or elsewhere. In order to address the complexity of the issues in all dimensions of the planetary boundaries, context-specific inquiries into the impact and effect of the ISMIR research and technologies developed

and used by the community are needed. Furthermore, a broader cultural shift in thinking around AI development is necessary to bring environmental sustainability to the ISMIR research agenda. We argue that ISMIR can lead by a good example of more environmentally conscious model development, more mindful and minimalistic energy use, and reflective accounting for the environmental externalities and their political economies in current research and development practices.

We acknowledge that the calculations presented here are necessarily tentative by their nature. This is inherently a result of the lack of transparency in the ISMIR publications. While the information currently provided can provide us with indications of the scale of energy used in training the models, there are several details that may impact the exact values of these variables, which cannot be accounted for due to partial or lacking information. Such inaccuracies may skew the implied environmental impact, with undesirable consequences for social practices in the community. However, we argue that our estimate is very conservative on several points: First, the factor of 20 multiplied with the energy used for one training is below the estimates of [14], second, the use of TDP ignores all additional energy consumption by other hardware, and third, we use the median as a statistic. We would therefore like to point out that the likely underestimated energy costs could lull the research community into a false sense of security and encourage it to refrain from efforts that would be valuable for the environment. These estimates nevertheless provide an important basis upon which further inquiries into the complete environmental and ecological footprint of the conference can build.

Furthermore, we understand that the authors who contributed to our estimates were those who actively documented resource requirements. These papers may seem unfairly a focus of critique in our work, as many other authors who did not volunteer resource information at all were not cited in the paper. We believe it is instrumental to document the need for specifying the use of resources in the ISMIR community, and encourage further proactive efforts toward that goal.

6. CONCLUSION

In the era of acute climate crisis, the interest in resource-demanding music generation and analysis tasks shows signs of acceleration rather than slowing down. It is essential that research communities such as ISMIR apply critical self-reflection and acknowledge their role in promoting practices that may be excessively harmful to the environment. Increased transparency in documentation in ISMIR papers would serve better accounting for the current impacts of the research, steering the community norms and guidelines towards more sustainable practices, and providing a positive example for the wider industry. We encourage the ISMIR community to continue these critical discussions around the ethical impacts of MIR, including environmental sustainability and its political ecologies and beyond.

⁷ Two models in the second corpus discuss the use of GPU resources for inference, but the included information does not allow conclusions about the energy consumption during the experiments.

7. ACKNOWLEDGMENTS

This paper is an outcome of a project funded by the Marianne and Marcus Wallenberg Foundation (MMW 2020.0102).

8. ETHICS STATEMENT

This work is based on secondary data that is publicly accessible in online sources. We acknowledge that our paper presents estimates with many uncertainties. Therefore, the numbers may imply an environmental impact larger or smaller than the actual one, in both cases to the detriment of the community. To mitigate this, we put great effort into clarifying the uncertainties in our method. We also see a risk that a paper focusing on quantitative aspects of environmental impact may fail to motivate larger-scale behavior change, which in the context of global crisis may be seen as an ethical shortcoming.

9. REFERENCES

- [1] F. Morreale, M. Sharma, and I.-C. Wei, “Data collection in music generation training sets: A critical analysis,” in *International Society for Music Information Retrieval Conference (ISMIR)*, Milan, Italy, Nov. 2023.
- [2] A.-L. Ligozat, J. Lefèvre, A. Bugeau, and J. Combaz, “Unraveling the hidden environmental impacts of AI solutions for environment,” *Sustainability*, vol. 14, no. 9, 2022.
- [3] A. Lacoste, A. Luccioni, V. Schmidt, and T. Dandres, “Quantifying the Carbon Emissions of Machine Learning,” Nov. 2019. [Online]. Available: <https://arxiv.org/abs/1910.09700>
- [4] P. Jääskeläinen, D. Pargman, and A. Holzapfel, “On the environmental sustainability of Ai art(s),” in *8th Workshop on Computing Within Limits*, Online, Jun. 2022.
- [5] International Energy Agency, “Electricity 2024,” Jan. 2024. [Online]. Available: <https://www.iea.org/reports/electricity-2024>
- [6] E. Markey, “Press release: Markey, Heinrich, Eshoo, Beyer introduce legislation to investigate, measure environmental impacts of artificial intelligence,” 2024. [Online]. Available: <https://bit.ly/3PSwHzm>
- [7] J. Sevilla, L. Heim, A. Ho, T. Besiroglu, M. Hobbhahn, and P. Villalobos, “Compute Trends Across Three Eras of Machine Learning,” in *2022 International Joint Conference on Neural Networks (IJCNN)*, Padua, Italy, Jul. 2022, pp. 1–8.
- [8] R. Masu, A. Melbye, J. Sullivan, and A. Jensenius, “NIME and the environment: Toward a more sustainable NIME practice,” in *International Conference on New Interfaces for Musical Expression (NIME)*, Online and NYU Shanghai, Jun. 2021.
- [9] A. C. on Neural Information Processing Systems (NeurIPS), “NeurIPS code of ethics,” 2024. [Online]. Available: <https://neurips.cc/public/EthicsGuidelines>
- [10] J. Barnett, “The Ethical Implications of Generative Audio Models: A Systematic Literature Review,” in *Proceedings of the 2023 AAAI/ACM Conference on AI, Ethics, and Society (AIIES)*, Montréal, Canada, Aug. 2023, pp. 146–161.
- [11] S. Oğul, “In tune with ethics: Responsible artificial intelligence and music industry,” *Reflectif Journal of Social Sciences*, vol. 5, no. 1, pp. 139–149, 2024.
- [12] F. Morreale, “Where does the buck stop? Ethical and political issues with AI in music creation,” *Transactions of the International Society for Music Information Retrieval (ISMIR)*, vol. 4, no. 1, pp. 105–113, Jul. 2021.
- [13] R. Schwartz, J. Dodge, N. A. Smith, and O. Etzioni, “Green AI,” *Communications of the ACM*, vol. 63, pp. 54–63, Nov. 2020.
- [14] E. Strubell, A. Ganesh, and A. McCallum, “Energy and policy considerations for deep learning in NLP,” in *Annual Meeting of the Association for Computational Linguistics*, Florence, Italy, 2019, pp. 3645–3650.
- [15] P. Dhar, “The carbon impact of artificial intelligence,” in *Nature Machine Intelligence*, vol. 2, Aug. 2020, pp. 423–425.
- [16] P. Henderson, J. Hu, J. Romoff, E. Brunskill, D. Jurafsky, and J. Pineau, “Towards the systematic reporting of the energy and carbon footprints of machine learning,” *Journal of Machine Learning Research*, vol. 21, no. 248, pp. 1–43, 2020.
- [17] A. S. Luccioni, S. Viguier, and A.-L. Ligozat, “Estimating the carbon footprint of BLOOM, a 176B parameter language model,” *Journal of Machine Learning Research*, vol. 24, pp. 1–15, 2023.
- [18] C.-J. Wu, R. Raghavendra, U. Gupta, B. Acun, N. Ardalani, K. Maeng, G. Chang, F. A. Behram, J. Huang, C. Bai, M. Gschwind, A. Gupta, M. Ott, A. Melnikov, S. Candido, D. Brooks, G. Chauhan, B. Lee, H.-H. S. Lee, B. Akyildiz, M. Balandat, J. Spisak, R. Jain, M. Rabbat, and K. Hazelwood, “Sustainable AI: Environmental implications, challenges and opportunities,” 2022. [Online]. Available: <https://arxiv.org/abs/2111.00364>
- [19] A. S. Luccioni and A. Hernandez-Garcia, “Counting carbon: A survey of factors influencing the emissions of machine learning,” 2023. [Online]. Available: <http://arxiv.org/abs/2302.08476>
- [20] C. Douwes, P. Esling, and J.-P. Briot, “Energy consumption of deep generative audio models,” 2021. [Online]. Available: <https://arxiv.org/abs/2107.02621>

- [21] C. Douwes, G. Bindi, A. Caillon, P. Esling, and J.-P. Briot, “Is quality enough f integrating energy consumption in a large-scale evaluation of neural audio synthesis models,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Rhodes, Greece, Jun. 2023, pp. 1–5.
- [22] L. Zhuo, R. Yuan, J. Pan, Y. Ma, Y. Li, G. Zhang, S. Liu, R. Dannenberg, J. Fu, C. Lin *et al.*, “Lyricwhiz: Robust multilingual lyrics transcription by whispering to ChatGPT,” in *International Society for Music Information Retrieval Conference (ISMIR)*, Milan, Italy, 2023.
- [23] Y. Ding and A. Lerch, “Audio embeddings as teachers for music classification,” in *International Society for Music Information Retrieval Conference (ISMIR)*, Milan, Italy, 2023.
- [24] P. Jääskeläinen and A. Biorn-Hansen, “Critical questions for sustainability research in computational creativity,” in *ICCC International Conference of Computational Creativity*. ICCC, 2024.
- [25] P. Jääskeläinen, A. Holzapfel, and C. Åsberg, “Exploring more-than-human caring in Creative-Ai interactions,” in *Proceedings of the 18th Nordic Conference on Human-Computer Interaction (NordiCHI)*, Aarhus, Denmark, Oct. 2022.
- [26] Y. Strengers, “Smart energy in everyday life: are you designing for resource man?” *Interactions*, vol. 21, no. 4, pp. 24–31, 2014.
- [27] P. Li, J. Yang, M. A. Islam, and S. Ren, “Making AI less “thirsty”: Uncovering and addressing the secret water footprint of AI models,” 2023. [Online]. Available: <https://arxiv.org/abs/2304.03271>
- [28] K. Devine, *Decomposed: The political ecology of music*. Cambridge, Massachusetts: The MIT Press, 2019.
- [29] M. Brennan, “The environmental sustainability of the music industries,” in *Cultural Industries and the Environmental Crisis. New Approaches for Policy*, K. Oakley and M. Banks, Eds. Springer, 2020, pp. 37–49.
- [30] L. Hilty, “Why energy efficiency is not sufficient - some remarks on “Green by IT”.” Dessau, Germany: Shaker Verlag, 2012, pp. 13–20.
- [31] A. Holzapfel, “Introducing political ecology of Creative-Ai,” in *Critical Studies of Artificial Intelligence*, S. Lindgren, Ed. Glos: Edward Elgar Publishing, 2023, pp. 691–703.
- [32] F. Ronchini and R. Serizel, “Performance and energy balance: a comprehensive study of state-of-the-art sound event detection systems,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Seoul, Korea, 2024.
- [33] S. Keith, S. Collins, A. Renzo, and A. Mesker, “Slave to the ‘Rithm. The AI Turn in the Music Industries,” in *AI and the Future of Creative Work*. New York: Routledge, 2023, pp. 36–54.
- [34] The International Conference on New Interfaces for Musical Expression (NIME), “NIME conference environmental statement,” 2020. [Online]. Available: <https://www.nime.org/environment/>
- [35] J. Sullivan, R. Masu, and A. P. Melbye, “Info and resources for sustainable NIME research,” 2021. [Online]. Available: <https://eco.nime.org/>
- [36] Annual Conference on Neural Information Processing Systems (NeurIPS), “NeurIPS Paper Checklist Guidelines,” 2023. [Online]. Available: <https://neurips.cc/public/guides/PaperChecklist>
- [37] ———, “NeurIPS paper checklist guidelines,” 2021. [Online]. Available: <https://neurips.cc/Conferences/2021/PaperInformation/PaperChecklist>
- [38] I.-T. Liu, Y.-T. Lin, and J.-L. Wu, “Music cut and paste: A personalized musical medley generating system,” in *International Society for Music Information Retrieval Conference (ISMIR)*, Curitiba, Brazil, Nov. 2013.
- [39] S. Cherla, T. Weyde, and A. S. d’Avila Garcez, “Multiple viewpoint melodic prediction with fixed-context neural networks,” in *International Society for Music Information Retrieval Conference (ISMIR)*, 2014.
- [40] G. Percival, S. Fukayama, and M. Goto, “Song2quartet: A system for generating string quartet cover songs from polyphonic audio of popular music,” in *International Society for Music Information Retrieval Conference (ISMIR)*, Malaga, Spain, Oct. 2015.
- [41] H. Lim, S. Rhyu, and K. Lee, “Chord generation from symbolic melody using blstm networks,” in *International Society for Music Information Retrieval Conference (ISMIR)*, 2017.
- [42] A.-K. Kaila and A. Holzapfel, “Green MIR data table 2024,” 2024. [Online]. Available: <https://doi.org/10.5281/zenodo.12750436>
- [43] L. F. W. Anthony, B. Kanding, and R. Selvan, “Carbon-tracker: Tracking and predicting the carbon footprint of training deep learning models,” in *ICML Workshop on “Challenges in Deploying and monitoring Machine Learning Systems” at the Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2020.
- [44] M. McCallum, F. Korzeniowski, S. Oramas, F. Gouyon, and A. Ehmann, “Supervised and unsupervised learning of audio representations for music understanding,” in *International Society for Music Information Retrieval Conference (ISMIR)*, Bengaluru, India, Dec. 2022.

- [45] G. Plaja-Roglans, M. Miron, and X. Serra, “A diffusion-inspired training strategy for singing voice extraction in the waveform domain,” in *International Society for Music Information Retrieval Conference (ISMIR)*, Bengaluru, India, Dec. 2022.
- [46] W. Wei, P. C. Li, Y. Yu, and W. Li, “Hppnet: Modeling the harmonic structure and pitch invariance in piano transcription,” in *International Society for Music Information Retrieval Conference (ISMIR)*, Bengaluru, India, Dec. 2022.
- [47] C. Hawthorne, I. Simon, A. Roberts, N. Zeghidour, J. Gardner, E. Manilow, and J. Engel, “Multi-instrument music synthesis with spectrogram diffusion,” in *International Society for Music Information Retrieval Conference (ISMIR)*, Bengaluru, India, Dec. 2022.
- [48] K. Toyama, T. Akama, Y. Ikemiya, Y. Takida, W. Liao, and Y. Mitsufuji, “Automatic piano transcription with hierarchical frequency-time transformer,” in *International Society for Music Information Retrieval Conference (ISMIR)*, Milan, Italy, Nov. 2023.
- [49] S. Sarkar, E. Benetos, and M. Sandler, “EnsembleSet: a new high quality synthesised dataset for chamber ensemble separation,” in *International Society for Music Information Retrieval Conference (ISMIR)*. Bengaluru, India: ISMIR, Nov. 2022.
- [50] Y. Ma, R. Yuan, Y. Li, G. Zhang, C. Lin, X. Chen, A. Ragni, H. Yin, E. Benetos, N. Gyenge, R. Liu, G. Xia, R. B. Dannenberg, Y. Guo, and J. Fu, “On the Effectiveness of Speech Self-Supervised Learning for Music,” in *International Society for Music Information Retrieval Conference (ISMIR)*, Milan, Italy, Nov. 2023.
- [51] P. Alonso-Jiménez, X. Serra, and D. Bogdanov, “Efficient Supervised Training of Audio Transformers for Music Representation Learning,” in *International Society for Music Information Retrieval Conference (ISMIR)*, Milan, Italy, Nov. 2023.
- [52] M. Perez, H. Kirchhoff, and X. Serra, “Triad: Capturing harmonics with 3d convolutions,” in *International Society for Music Information Retrieval Conference (ISMIR)*, Milan, Italy, Nov. 2023.
- [53] G. Brunner, A. Konrad, Y. Wang, and R. Wattenhofer, “MIDI-VAE: Modeling dynamics and instrumentation of music with applications to style transfer,” in *International Society for Music Information Retrieval Conference (ISMIR)*, Paris, France, 2018.
- [54] Y. Teng, A. Zhao, and C. Goudeseune, “Generating nontrivial melodies for music as a service,” in *International Society for Music Information Retrieval Conference (ISMIR)*, Suzhou, China, Oct. 2017.
- [55] B. D. Giorgi, M. Levy, and R. Sharp, “Mel spectrogram inversion with stable pitch,” in *International Society for Music Information Retrieval Conference (ISMIR)*, Bengaluru, India, Dec. 2022.
- [56] D.-Y. Wu, W.-Y. Hsiao, F.-R. Yang, O. D. Friedman, W. Jackson, S. Bruzenak, Y.-W. Liu, and Y.-H. Yang, “DDSP-based singing vocoders: A new subtractive-based synthesizer and a comprehensive evaluation,” in *International Society for Music Information Retrieval Conference (ISMIR)*, Bengaluru, India, Dec. 2022.
- [57] J. Zhao, G. Xia, and Y. Wang, “Beat Transformer: Demixed Beat and Downbeat Tracking with Dilated Self-Attention,” in *International Society for Music Information Retrieval Conference (ISMIR)*, Bengaluru, India, Dec. 2022.
- [58] C. Donahue, H. H. Mao, Y. Li, G. Cottrell, and J. McAuley, “Lakhnes: Improving multi-instrumental music generation with cross-domain pre-training,” in *International Society for Music Information Retrieval Conference (ISMIR)*, Delft, Netherlands, Nov. 2019.
- [59] C. Donahue, J. Thickstun, and P. Liang, “Melody transcription via generative pre-training,” in *International Society for Music Information Retrieval Conference (ISMIR)*, Bengaluru, India, Dec. 2022.
- [60] Y.-T. Yeh, B.-Y. Chen, and Y.-H. Yang, “Exploiting pre-trained feature networks for generative adversarial networks in audio-domain loop generation,” in *International Society for Music Information Retrieval Conference (ISMIR)*, Bengaluru, India, Dec. 2022.
- [61] Y.-K. Wu, C.-Y. Chiu, and Y.-H. Yang, “Jukedrummer: Conditional beat-aware audio-domain drum accompaniment generation via transformer VQ-VAE,” in *International Society for Music Information Retrieval Conference (ISMIR)*, Bengaluru, India, Dec. 2022.
- [62] A. Singh, K. Demuynck, and V. Arora, “Attention-based audio embeddings for query-by-example,” in *International Society for Music Information Retrieval Conference (ISMIR)*, Bengaluru, India, Dec. 2022.
- [63] S.-L. Wu and Y.-H. Yang, “The jazz transformer on the front line: Exploring the shortcomings of AI-composed music through quantitative measures,” in *International Society for Music Information Retrieval Conference (ISMIR)*, Montréal, Canada, Oct. 2020.
- [64] M. Pasini and J. Schlüter, “Musika! Fast infinite waveform music generation,” in *International Society for Music Information Retrieval Conference (ISMIR)*, Bengaluru, India, Dec. 2022.
- [65] J. Zhao, G. G. Xia, and Y. Wang, “Domain adversarial training on conditional variational auto-encoder for controllable music generation,” in *International Society for Music Information Retrieval Conference (ISMIR)*, Bengaluru, India, Dec. 2022.

- [66] D. Zhang, J.-C. Wang, K. Kosta, J. B. L. Smith, and S. Zhou, “Modeling the rhythm from lyrics for melody generation of pop songs,” in *International Society for Music Information Retrieval Conference (ISMIR)*, Bengaluru, India, Dec. 2022.
- [67] N. Srivatsan and T. Berg-Kirkpatrick, “Checklist models for improved output fluency in piano fingering prediction,” in *International Society for Music Information Retrieval Conference (ISMIR)*, Bengaluru, India, Dec. 2022.
- [68] G. Mittal, J. Engel, C. Hawthorne, and I. Simon, “Symbolic music generation with diffusion models,” in *International Society for Music Information Retrieval Conference (ISMIR)*, Online, 2021.
- [69] F. Foscarin, D. Harasim, and G. Widmer, “Predicting music hierarchies with a graph-based neural decoder,” in *International Society for Music Information Retrieval Conference (ISMIR)*, Milan, Italy, 2023.
- [70] O. Peracha, “Improving polyphonic music models with feature-rich encoding,” in *International Society for Music Information Retrieval Conference (ISMIR)*, Delft, Netherlands, Nov. 2019.
- [71] J. Rockström, W. Steffen, K. Noone, F. S. C. Å. Persson, E. Lambin, and et al., “Planetary boundaries: exploring the safe operating space for humanity,” *Ecology and society*, vol. 14, no. 2, 2009.
- [72] K. Richardson, W. Steffen, W. Lucht, J. Bendtsen, S. E. Cornell, J. F. Donges, M. Drüke, I. Fetzer, G. Bala, W. von Bloh, G. Feulner, S. Fiedler, D. Gerten, T. Gleeson, M. Hofmann, W. Huiskamp, M. Kummu, C. Mohan, D. Nogués-Bravo, S. Petri, M. Porkka, S. Rahmstorf, S. Schaphoff, K. Thonicke, A. Tobian, V. Virkki, L. Wang-Erlandsson, L. Weber, and J. Rockström, “Earth beyond six of nine planetary boundaries,” *Science Advances*, vol. 9, no. 37, Sep. 2023.

Field Study on Children's Home Piano Practice: Developing a Comprehensive System for Enhanced Student-Teacher Engagement

Seikoh Fukuda¹

PTNA Research Institute
of Music
seikoh@piano.or.jp

Ami Motomura, Eri Sasao

To-on Kikaku Company
motomura@to-on.com,
e_sasao@to-on.com

Yuko Fukuda¹

Kyoritsu Women's
University²
yuko@piano.or.jp

Masaki Matsubara

University of Tsukuba
masaki@slis.
tsukuba.ac.jp

Masamichi Hosoda

NTT East Corporation
masamichi.hosoda.at
@east.ntt.co.jp

Masahiro Niitsuma

Keio University
mniitsuma@keio.jp

ABSTRACT

Regular weekly lessons and daily home practice are key for skill development. This paper focuses on identifying the challenges within such practice routines and developing a system to address these issues, thereby enhancing teacher support and elevating student performance in piano. Observations from real-world lessons and an analysis of practice videos spanning 177 days from 30 students reveal successful tactics, including the assignment of suitably challenging pieces and motivational rewards like stickers or stamps. Furthermore, the study underscores issues such as tension in parent-led practice and ineffective repetition. Insights from the field study suggest the potential of third-party feedback, practice segmentation, reporting practice records to teachers, and rewarding practice sessions. We developed a system incorporating these solutions and tested it with 80 children over 4 months. Results showed increased teacher engagement with students' home practice, improved student motivation and practice duration, and enhanced sight-reading skills, demonstrating the system's effectiveness in supporting piano education.

1. INTRODUCTION

Weekly lessons and daily home practice are vital for skill growth in young piano students [1-3]. However, teachers often rely solely on lesson performance to address issues in unseen home practice. Fostering resilience is essential in daily piano education. Research suggests praising not just outcomes, but also effort and perseverance [4]. Therefore, piano instructors should evaluate and commend not only performance outcomes but also efforts during home practice and the ability to overcome difficulties.

Identifying home practice challenges enables efficient skill improvement through targeted interventions and support systems. This paper aims to (1) identify home practice challenges and (2) develop a system to address them. This study aims to address issues and evaluate the system's effectiveness in improving practice outcomes.

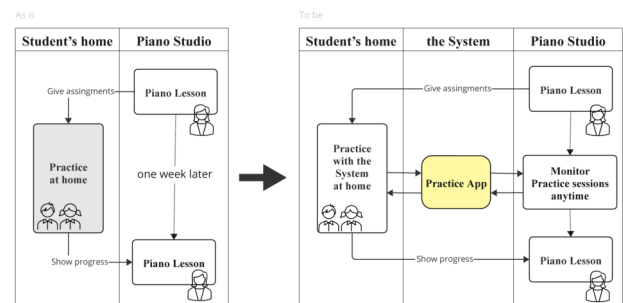


Figure 1. (Left) Current practice (Right) Enhanced System with App.

In 2023, with the cooperation of Piano Teachers' National Association of Japan (PTNA) [5], (1a) interviews were conducted with 8 piano teachers, (1b) observations were made of the piano lessons of their 12 students, (1c) survey results regarding home practice were collected from 81 piano teachers, (1d) one-week home practice records were obtained from 37 students (Average age: 7.02), and (1e) the analysis of 177 days of home practice videos from 30 of those students (Average age: 6.93) was performed. (2) Based on these findings, a support system (Fig. 1) was developed and tested over 4 months with 80 students (Average age: 7.11) and 46 teachers, aiming to enhance practice efficiency and outcomes. These students are a different population from the subjects in survey (1a-1e).

The evaluation of the system's effectiveness, based on its usage and surveys conducted before and after the trial, revealed the following:

- Segmenting tasks of target musical score, which is assigned as homework, increased students' practice time and improved their sight-reading skills.
- Reporting practice time and frequency to teachers increased teachers' awareness of home practice.
- Providing incentives for each piece practiced enhanced students' motivation and initiative to practice.



©S. Fukuda, Y. Fukuda, M. Hosoda, A. Motomura, E. Sasao, M. Matsubara, and M. Niitsuma. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** S. Fukuda, Y. Fukuda, M. Hosoda, A. Motomura, E. Sasao, M. Matsubara, and M. Niitsuma, "Field Study on Children's Home Piano Practice: Developing a Comprehensive System for Enhanced Student-

Teacher Engagement", in *Proc. of the 25th Int. Society for Music Information Retrieval Conf.*, San Francisco, United States, 2024.

¹ Seikoh Fukuda and Yuko Fukuda, Co-first author

² Adjunct Instructor at Kyoritsu Women's University

2. FIELD STUDY

2.1 Cultural Background

In Japan, piano lessons are the second most popular extra-curricular activity for elementary school students [6], and almost all students attending piano classes have a piano at home and practice daily. Throughout the 9 years of compulsory education from the first year of elementary school to the third year of middle school, music is consistently a compulsory subject, resulting in high levels of music literacy. Although few children aim to become piano professionals, it is presumed that many parents recognize the educational value of learning music and piano [7]. The educational value of music and piano learning is evident from the fact that many students at major U.S. universities like Harvard and MIT [8] study music as part of their liberal arts education and focus on developing non-cognitive skills [9]. The role of parents in their children's piano learning in Japan is multifaceted. Parents manage the daily practice schedule and maintain their children's motivation through feedback and encouragement. They also work closely with piano teachers, correcting mistakes in place of the teachers to ensure effective practice at home. In this way, active parental involvement significantly impacts the duration and progress of their children's piano learning.

2.2 Lesson Observations and Teacher Interviews

To explore how to maximize the effectiveness of home practice, we invited 8 experienced piano teachers (30s to 60s) to observe 12 lessons across 4 piano classes. These observations, coupled with interviews, highlighted 3 key factors essential for enhancing home practice:

- (1) Receiving objective feedback from a third party to gain a clearer perspective on one's own performance [10].
- (2) Assigning homework that is appropriately challenging, considering the student's age, experience, parental support, and skill level [11-12].
- (3) Rewarding completed assignments with stickers or stamps to motivate students [13].

These teachers, with their deep expertise, foster substantial musical skills, contributing to students' continued engagement with piano through high school and college.

2.3 Teacher and Student Questionnaires

A questionnaire was set up on the website of the Piano Teachers' National Association to clarify teachers' perceptions and students' actual practice conditions at home. Responses were collected from 81 teachers of various ages, genders, and skill levels. The student survey was conducted through teachers, with 37 students from schools reporting their practice status daily for 1 week using Google Forms.

Teacher Questionnaire: The top concern for teachers regarding students' home practice was "insufficient practice days," accounting for 83% of responses. This was followed by 59% of the teachers indicating that students practicing with incorrect sounds and rhythms was a concern.

Student Questionnaire: The home practice records were submitted via Google Form every day after piano practice.

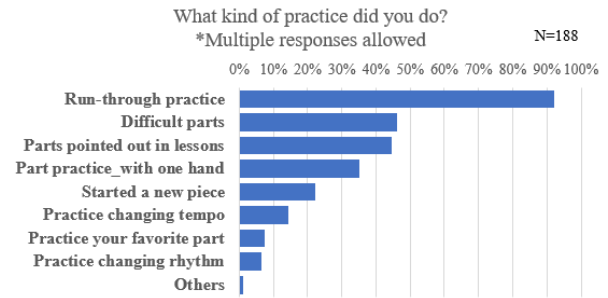


Figure 2. Survey Results on Practice Content for Elementary School Students.

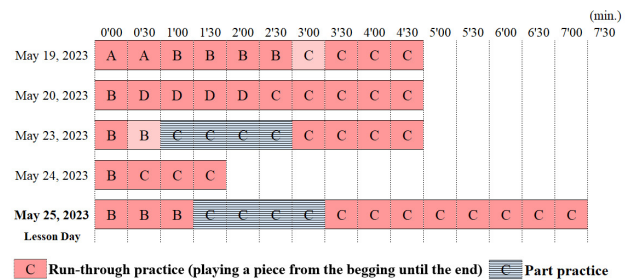


Figure 3. Timeline of home practice for first graders: most of the time was run-through practice.

Days and duration: Out of the total 259 days surveyed (37 students × 7 days), 250 days of responses were received. The number of practice days was 188, averaging 5.08 days per week per person, indicating that they practice on week-days. In addition, 69.1% of the respondents (130 out of 188 days) practiced for more than 15 minutes at a session.

Practice content: An analysis of responses (Fig. 2) to questions about actual practice content revealed that 92.0% (173 out of 188 days) of students reported performing "full run-throughs" of pieces from start to finish. However, only about half of the students practiced "partial sections" such as practicing difficult parts (46.3% or 87 days), practicing parts pointed out in lessons (44.7% or 84 days), or practicing with 1 hand (35.1% or 66 days).

The survey results from both teachers and students revealed a gap in their perceptions. While 83% of the 81 teachers surveyed expressed concerns about the insufficient number of practice days, the student survey results showed that students practiced an average of 5.08 days per week, with 69.1% spending more than 15 minutes per practice session. These results highlight a significant discrepancy between teachers' perceptions and the actual practice conditions of students.

2.4 Analysis of Home Practice Videos from Students

Thirty of the students in the study recorded their home practice for the same week and uploaded the video to Google Drive. As a result, a total of 177 days practice videos were collected. These videos were viewed and analyzed by 5 active teachers, with each teacher assigned a different set of videos to review. The analysis was conducted based on a format that included 5 items: timeline, piece, practice sections, practice methods, and free comments, allowing each teacher to record their observations.

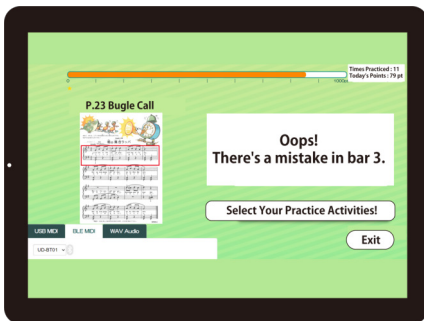


Figure 4. Screenshot of performance assessment.

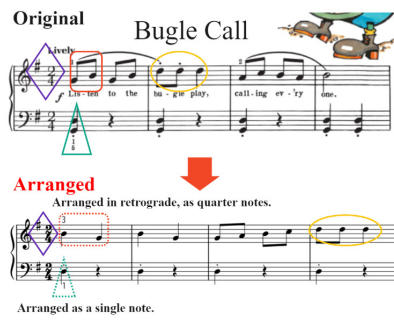


Figure 5. Example of Part Practice Method B

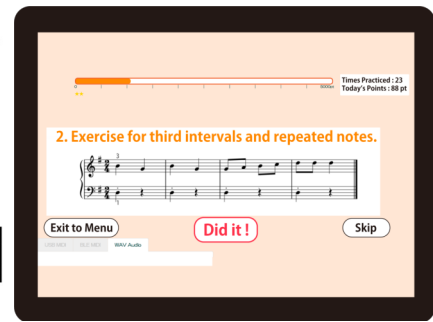


Figure 6. User Interface: students push “Did it !” button after each part practice was completed

For example, Fig. 3 shows a timeline of home practice of first-grade elementary school student who worked on 4 pieces labeled A to D over 5 days of the week. The videos showed the student independently engaging in practice, with a high proportion of full run-throughs in their practice routine. On 1 day, the student repeated a full run-through of the same section 3 times, making the same mistakes each time, but then moved on to the next piece without correcting them. On the day of the lesson (May 25, 2023), the practice time was longer than usual, and the mother's involvement was also observed.

In other videos, various methods of counting the number of plays were observed, such as using an iPad or notebook to keep track, or using educational toys like “Pop-It” to count. There was a tendency to end the practice session after a certain number of repetitions, regardless of whether they could play the sections correctly or not.

Moreover, from the perspective of parental involvement, a correlation was observed between the extent of parental involvement, the completion of assignments, and the students' initiative. Active parental involvement was seen to accelerate technical progress in students, although it tended to suppress their autonomy. While children's skills improved when parents pointed out mistakes in sound or gave prompts similar to those of teachers in lessons, this also led to situations where the child felt pressured and became overly tense. On the other hand, when parents were not overly involved and only supported when prompted by their child, the students tended to practice independently. Although practice often ended based on the number of times played, mistakes sometimes remained uncorrected over time. Furthermore, parents encouraging children to think about the next steps and motivating them through praise and encouragement helped support the children in approaching practice in a relaxed and thoughtful manner.

2.5 Identified Challenges from Field Study

The field study revealed the following challenges: (1) Students themselves find it difficult to objectively view mistakes in sound and rhythm. However, it is challenging for parents, who may not have a deep understanding of piano instruction, to provide appropriate support that is neither too interfering nor disinterested. (2) Merely completing a set number of full run-throughs makes it difficult to overcome sections that are not well-played. (3) There is a gap

between teachers' perceptions and the actual practice conditions of students. (4) Rewards are effective in improving motivation, but since they are only received during weekly lessons, they do not easily motivate home practice.

3. SYSTEM DESCRIPTION

Based on the identified challenges in the previous section, a system was designed to enhance the efficiency of students' home practice. The implemented features in this system are as follows:

- (A) Providing feedback on whether a performance is correct or incorrect by a third party other than parents
- (B) Encouraging targeted practice of difficult sections by segmenting practice pieces [3],[14]
- (C) Enabling teachers to review home practice records at any time
- (D) Motivating students by providing rewards every time they play their practice pieces, visualizing these rewards

3.1 Overview of the Practice App

To achieve the objectives (A) through (D), we implemented the system as follows. It is important to note that while objective (A) is only accessible to users of digital pianos with MIDI output, objectives (B) through (D) are available for both electronic and acoustic piano users.

(A) Design of Performance AI Assessment by System

The system is designed to allow a third party other than parents to provide feedback on the correctness of a performance.

After selecting the homework piece, the student chooses which section of the piece, previously divided into units of about 4 measures, they wish to practice. They start the performance by pressing the “Start” button. The student's performance is recorded in MIDI and converted into a Standard MIDI File (SMF). The recording is done without a metronome or click track to allow the student to play at their own tempo. The student's performance SMF is then compared with a pre-prepared exemplary performance SMF. As a preprocessing step before comparison, the note ON events in the SMF are sorted chronologically. Note ON events within 50 ms of each other are considered

simultaneous and are sorted by MIDI note number in ascending order. The preprocessed SMFs are compared using Dynamic Programming (DP) matching [15] to find corresponding notes between the student's performance and the exemplary performance, and any discrepancies are detected as mistakes. Since this is intended for beginners, a simple method like this is sufficient for now. However, using symbolic music alignment instead of DP matching is a subject for future consideration.

If the performance is flawless, the system responds with "Well played!" If there are mistakes, the system points out the first bar where a mistake occurred (Fig. 4).

(B) Design for Segmenting Practice Pieces

Video observation of home practice sessions revealed that it is difficult to compensate for mistakes and weaknesses by simply practicing through the piece. Therefore, we aimed to encourage segmented practice in order to improve overall mastery of the piece.

Students select their homework piece and, after receiving system feedback on any 4-bars block, they can choose to work on segmented practice pieces using 1 of following 2 methods:

Part Practice Method A: Simplifies the homework piece by concealing parts of the score every 4 measures. This method aims to focus on specific sections by maintaining the original sheet music's staff lines, bar spacing, and note-head sizes, while intentionally hiding parts to help students focus on particular areas.

Part Practice Method B: This method involves identifying key learning elements that are either crucial for learning or where many students stumble. Short 4-bars pieces, simpler than the original, are composed that include some of these learning elements (Fig. 5). The contents were composed by 6 music majors, including three active piano instructors. The following conditions apply to the composition process [16]:

- #1: Include at least one challenging learning element from the original phrase.
 - #2: Maintain the same time signature, position, and key as the original phrase.
 - #3: Include fewer learning elements than the original.
 - #4: Maintain or lower the level of learning elements. Lowering is defined as reverting to already learned related elements.
 - #5: If using elements other than melody and rhythm, employ the same starting note, melody, and rhythm as the practice phrase from the original.
 - #6: Use a melody that the students may have heard before.
- In both Part Practice Method A and B, students can either play the presented 4-bars practice piece or choose to skip it by pressing the skip button located at the bottom right of the sheet and move on to the next original practice piece.

(C&D) Design of Monitoring and Rewarding

Instead of teachers assessing students' home practice solely based on their performance during weekly lessons, the design allows teachers to continuously check daily and cumulative practice time since the start of using the Practice App, the number of times practice pieces are played, and the points earned.

The rewarding design: 1 point for just logging in, 1 to 5 points for pressing the "Did it!" button (Fig. 6), and 10 to

50 points awarded by teachers as a reward. The educational philosophy of this system is "from result-oriented to process-oriented." In a result-oriented approach, perfect performances evaluated by the AI performance assessment would likely earn higher points. However, in a process-oriented approach, value is found in the attempt itself, and regardless of the performance outcome, a consistent 5 points are awarded. Thus, these experimental results are evaluated without a strong AI performance assessment component, other than the simple pitch errors.

3.2 Overview of PoC (Proof of Concept)

Students participating in the PoC were recruited via the website of an organization for piano teachers. The PoC is not an independent experiment but is incorporated into actual students' regular lessons and practice. Participants were selected based on their responses to questions about teaching materials, instruments owned, and devices owned. Additionally, 30 tablets for the PoC were lent out, and it was anticipated that students would use devices (tablets, smartphones, computers) alongside their usual sheet music.

For the performance assessment feature, students who mainly use digital pianos at home were targeted, although some students with acoustic pianos were also accepted. The teaching materials used were "Bastien New Traditions: All In One Piano Course - Level 1A" and "Bastien Piano Basics [17]: Piano - Level 1," both of which have been translated into over 16 languages worldwide.

The PoC was conducted from October 2023 for 4 months. Piano students using the Practice App were introduced by their teachers, and the teachers' surveys were linked to individual students for analysis. To validate the Practice App, a pre-assessment questionnaire was conducted at the beginning and a post-assessment questionnaire after 4 months.

3.3 Results of System Usage

3.3.1 Period and number of participants

Students who participated in the PoC were referred by 46 teachers, and 80 students used the Practice App at least once. The age of the students mainly ranged from first to third grade of elementary school, with a few preschoolers and fourth to 6th graders included. The number of days the Practice App was used ranged from a minimum of 1 day to a maximum of 117 days, with an average usage of 39.2 sessions. The Total points, indicating the level of activity in using the Practice App, ranged from a minimum of 5 points to a maximum of 8,628 points, averaging 1,399 points. 36 teachers monitored their students' practice sessions at least once using the Practice App.

3.3.2 Comparison of Pre/Post PoC Questionnaire

The same questionnaires were administered to students before and after the PoC to validate the effectiveness of the Practice App. The questionnaires used a 5 level Likert scale to ask about students' attitudes towards piano practice and their parent-child relationships. The responses were based on the respondents' subjective perceptions of these aspects. 52 students responded to both the pre and post questionnaires. The students who earned more than 1,000

Questionnaire for HPG Students	Mean	
	before	after
Does your child enjoy daily practice? **	3.20	3.64
Is your child self-motivated in daily piano practice? **	2.76	3.52
How long do you practice each day? *	3.12	3.52
How do you feel about your relationship with your child during daily piano practice?	3.16	3.24

Table 1. Paired t-test of Pre- and Post-PoC Questionnaires for HPG Students (N=25, **p<0.01, *p<0.05)

points with the Practice App were categorized as the High-Practice Group (HPG) with 25 students, and those who earned less than 1,000 points were categorized as the Low-Practice Group (LPG) with 27 students. For the group of 25 HPG participants, a paired t-test was conducted on the pre- and post-assessment questionnaire results. As shown in Table 1, significant improvements were observed in Daily Practice Time, Voluntariness in Practice, and Enjoyment of Practice. However, no significant effect was observed in improving parent-child relationships.

3.3.3 Validation by questionnaire after PoC

Responses to questions included only in the post-PoC questionnaire were collected from 64 participants. These were divided into two groups: 30 in the High-Practice Group (HPG) and 34 in the Low-Practice Group (LPG). The average scores for HPG were listed in descending order in Table 2. Independent sample t-tests were conducted for each question.

In the HPG, half of the 14 question items averaged 4.0 points or higher. Furthermore, HPG received significantly higher scores than LPG in 9 out of the 14 questions. The item “Increased Voluntariness for Practice” in Table 2 corresponds to “Voluntariness in Practice,” which showed significant effects in the paired t-test described in previous section. Therefore, even though there were no significant differences found in the independent samples t-test for items like “Motivated by 'Did it!' Button,” “Supported by AI performance assessment,” and “Supported by Part Practice Method A,” the higher scores in “Increased Voluntariness for Practice” suggest that system was effective. Thus, it is estimated that the system influenced 13 out of the 14 items.

3.3.4 Results of the Teacher Questionnaire

After the PoC, feedback was obtained via a Likert scale questionnaire from 46 teachers, as shown in Table 3. An independent samples t-test was conducted between 26 teachers (HPG) who had at least 1 student scoring over 1,000 points and 20 teachers (LPG) who did not.

Out of 15 questionnaire items, 7 averaged 4.0 points or higher. Moreover, HPG received significantly higher responses in 10 items compared to LPG. The item "Have Students Use Part Practice Method A" scored particularly high for HPG at 4.69 points, with a significant difference from LPG. Conversely, "Have Students Use Part Practice Method B" was the only item among all 15 where both HPG and LPG teachers scored above 4.0 points. Significant responses were also seen in items relating to teacher

Questionnaire for Students	Mean	
	LPG	HPG
Did tracking practice motivate you? **	3.59	4.30
Did Method B support your practice? **	3.50	4.29
Did the "Did it!" button motivate you?	3.76	4.27
Did AI assessment support your practice?	3.67	4.27
Did Method B support your practice?	3.55	4.12
Did practice points motivate you? *	3.41	4.10
Did practicing become more enjoyable? **	3.15	4.00
Did your practice time and frequency increase? *	3.06	3.83
Did your piano skills improve? **	2.85	3.70
Did using Practice App motivate you? **	2.82	3.60
Did it reduce the burden on parents? *	2.85	3.60
Practice independently without parents? **	2.71	3.53
Did your motivation for practicing increase?	3.09	3.43
Did your teacher give you a passing mark earlier?	2.68	3.20

Table 2. Results and Mean Values from Independent Samples t-Test of Post-PoC Student Questionnaires Between HPG and LPG (N ranges from LPG: 9-34, HPG: 11-30, **p<0.01, *p<0.05)

Questionnaire for Teachers	Mean	
	LPG	HPG
Do you want students to use Method A? **	3.85	4.69
Did it spark home practice talks with students? **	3.30	4.42
Do you want students to use Method B?	4.15	4.38
Was Method B effective in improving sight-reading skills? *	3.95	4.38
Any insights from checking students' practice amount? **	3.20	4.27
Any positive changes in students? **	3.25	4.19
Did it help observe students' home practice? **	3.30	4.00
Did lesson efficiency improve? **	3.00	3.92
Did it lead to better lessons?	3.35	3.85
Did students' performance improve by the next lesson? **	2.85	3.85
Did students' sight-reading improve? *	3.20	3.73
Did it change how you assign homework?	3.15	3.65
Did AI assessment reduce pitch and rhythm mistakes?	3.05	3.54
Did points awarded by teachers motivate students? *	2.75	3.50
Did it increase the number of assigned pieces?	2.90	3.23

Table 3. Comparison of Mean Values Between Teachers with 1 or More Students in HPG and Those Without (N=LPG: 20, HPG: 26, **p<0.01, *p<0.05)

engagement with home practice, such as providing opportunities for discussions about home practice and observing the process.

The questions “Have Students Use Part Practice Method A,” “Have Students Use Part Practice Method B,” and “Part Practice Method B is effective for reading skills” reflect teachers' opinions on the functionality rather than the change in students due to implementation, which might explain the higher scores from LPG. As a result, while “Have Students Use Part Practice Method B” did not show a significant difference in scores between HPG and LPG, the high average score of 4.38 points for HPG indicates substantial positive expectations from the teachers.

3.4 Summary of Results

Results from section 3.3.2 indicated that there were significant differences in the Likert scale questionnaire scores

before and after the start of the PoC, demonstrating improvements in “Daily Practice Time,” “Voluntariness in Practice,” and “Enjoyment of Practice.”

From section 3.3.3, significant differences between the High-Practice Group and Low-Practice Group in the post-PoC questionnaire suggest that motivation for practice, practice time and frequency, and the sense of improvement increased while reducing parental burden. Items that scored an average of 4 points or higher in the HPG are considered to indicate the effectiveness of the Practice App. According to the results from section 3.3.4, the significant differences in scores between the HPG and LPG indicated an increase in teachers' awareness of home practice. Both Part Practice Method A and B being highly rated by both groups indicates that these methods are perceived as effective practices by teachers and hold high expectations.

4. DISCUSSION

4.1 Factors Enhancing Engagement

Motivational Effects: The visualization of the effort process has been shown to be effective in motivation in other studies as well [18], but this time, visualizing the efforts of child students at home practice with points confirmed significant motivational effects. For example, a second-grade elementary school monitor student practiced a song from the introductory tutorial book 3 times through, totaling about 2 minutes of practice before the PoC in October. However, two months after starting to use the Practice App, the student began practicing more than 30 minutes every day and was able to progress to “Burgmüller: 25 Progressive Etudes, Op. 100”[19]. This substantial change in motivation was attributed to daily point rewards by teachers, as revealed in interviews. The total points, including both self-reward points and teacher reward points, were always displayed. Students who noticed the addition of teacher points showed increased motivation. Moreover, segmenting practice pieces and increasing the frequency of pressing the “Did it!” button increased opportunities for earning points, enhancing students' autonomy and providing a game-like experience. This led to an increase in frequency and duration, thereby improving sight-reading skills.

Analysis of Part Practice Methods A and B: Field studies show that teachers have traditionally assigned students to practice with one hand or rhythm practice as homework [20], using methods such as writing instructions on the score or using sticky notes. However, in Part Practice Method A, for example, when practicing only with the right hand, the system hides the left-hand part, allowing focus solely on right-hand practice. The system displaying only the part being practiced helps students concentrate on the task without being distracted or overwhelmed by having to cognitively process the whole score first and subsequently disregard some parts. This focus on individual tasks was perceived as effective based on questionnaire results and post-interviews with teachers. Part Practice Method B is not just a specialized part-practice for the assigned homework piece but focuses on learning elements intended to be acquired in that piece, aimed at improving sight-reading skills overall. Interviews and questionnaires

with teachers suggest that compared to adult students aiming to master specific songs, there is a high expectation for child students to improve their sight-reading skills overall to play many pieces in the future.

Feedback and System Impact: In the free-response section of the post-use survey, both students and parents shared feedback such as, “It was helpful that the AI performance assessment could identify mistakes even when parents couldn't supervise the practice,” and “Knowing that the teacher was monitoring daily practice motivated the child (student).” Teachers also provided positive feedback, saying, “The system's suggestion for part practice helped students who tend to play through the entire piece from start to finish to adopt sectional practice,” and “The presence of the system as a third party seemed to reduce parents' frustration.” These responses aligned with the goals of our study, indicating a successful outcome.

4.2 Limitation

This study primarily aimed to conduct a PoC; hence, for the performance assessment, it did not involve using acoustic pianos with automatic musical acoustic alignment [21-24], but instead conducted validations using digital pianos, which offer higher recognition accuracy. Although the effectiveness of segmenting practice pieces was confirmed, the study did not perform detailed analyses such as comparing the impact of Part Practice Method A and B separately or comparing the effects of practice with and without segmentation. The interface was changed to English for the paper, but we use Japanese in practice. Additionally, feedback indicated the current teacher UI is difficult to use in multi-student classrooms. This suggests the need for UI improvements to reduce management costs for actual classroom deployment. Furthermore, while the study has statistically summarized outcomes, reports indicate that some students felt monitored during home practice, and it has not been possible to perform usability evaluations that consider such individual differences.

5. CONCLUSION

This study, through a large scale field study of piano teachers and students, revealed that the challenges in children's home piano practice include not recognizing errors in playing without parental support, repeating inefficient full run-throughs, teachers not understanding the practice process, and maintaining motivation. To address these problems, a system was developed incorporating performance assessment, presentation of segmented practice pieces, reports to teachers, and point allocation, and a PoC was conducted. The results confirmed that (1) the system identified mistakes, reducing parental burden, (2) increased practice time and improved sight-reading skills, (3) increased awareness among teachers about the practice process, and (4) enhanced student motivation and spontaneity. These outcomes suggest that the proposed system has the potential to enhance efficiency and effectiveness in children's piano learning. Challenges such as individual differences in UI and usability, as well as environmental settings, remain for actual deployment and are targeted for future work.

ACKNOWLEDGEMENTS

This work was supported by JSPS KAKENHI Grant number 22H01047. We would like to express our heartfelt thanks to the piano teachers and students who participated in this study by providing survey responses, interviews, home practice videos, and taking part in the PoC (Proof of Concept). Their contributions were essential in making this research possible.

SUPPLEMENTAL MATERIALS

All supplementary materials are available in "<https://prim.piano.or.jp/supplementary/ISMIR2024.html>"

REFERENCES

- [1] G. E. McPherson and J. W. Davidson, "Musical Practice: Mother and child interactions during the first year of learning an instrument," *Music Education Research*, vol. 4, pp. 141–156, 2002.
- [2] R. A. Duke, A. L. Simmons and C. D. Cash, "It's Not How Much; It's How: Characteristics of Practice Behavior and Retention of Performance Skills," *Journal of Research in Music Education*, Vol. 56, No. 4, pp. 310–321, 2009.
- [3] K. V. Hoover-Dempsey, A. C. Battiato, J. M. T. Walker, R. P. Reed, J. M. DeJong, and K. P. Jones, "Parental Involvement in Homework," *Educational Psychologist*, Vol. 36, No. 3, pp. 195–209, 2001.
- [4] C. M. Mueller and C. S. Dweck, "Praise for Intelligence Can Undermine Children's Motivation and Performance," *Journal of Personality and Social Psychology*, Vol. 75, pp. 33–52, 1998.
- [5] The Piano Teachers' National Association of Japan, "about PTNA," [online]. Available: <https://www.piano.or.jp/english/about/index.html>. Access date: 12 April 2024.
- [6] Ministry of Health, Labour and Welfare, "Longitudinal Survey of Births in the 21st Century (2010 births)," 2024. [online]. Available: <https://www.mhlw.go.jp/toukei/list/27-22.html>. Access date: 12 April 2024.
- [7] G. Schlaug, A. Norton, K. Overy and E. Winner, "Effects of Music Training on the Child's Brain and Cognitive Development," *Annals of the New York Academy of Sciences*, No. 1, pp. 219–230, 2005.
- [8] E. Sugano, "MIT Massachusetts Institute of Technology Music Class ~The world's best way to develop the power to create," Asa Publishing (in Japanese), 2020.
- [9] MIT Open Learning, "The Workforce Relevance of Liberal Arts Education," [online]. Available: <https://openlearning.mit.edu/news/workforce-relevance-liberal-arts-education>, Access date: 12 April 2024.
- [10] L. Hamond, E. Himonides and G. Welch, "The nature of feedback in higher education studio-based piano learning and teaching with the use of digital technology," *Journal of Music Technology & Education*, Vol. 13, No. 1, pp. 33–56, 2020.
- [11] S. Asahi, S. Tamura, Y. Sugiyama, and S. Hayamizu, "Toward a High Performance Piano Practice Support System for Beginners," in *Proc. of Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pp.73–79, 2018.
- [12] E. A. Locke and G. P. Latham, "Building a Practically Useful Theory of Goal Setting and Task Motivation," *American Psychologist*, Vol. 57, No. 9, pp. 705–717, 2002.
- [13] E. L. Deci, R. J. Vallerand, L. G. Pelletier and R. M. Ryan, "Motivation and Education: The Self-Determination Perspective," *Educational Psychologist*, Vol. 26, pp. 325–346, 1991.
- [14] G. E. McPherson, "From child to musician: Skill development during the beginning stages of learning an instrument," *Psychology of Music*, vol. 33, pp. 5–35, 2005.
- [15] R. B. Dannenberg, "An on-line algorithm for real-time accompaniment," *International Computer Music Conference (ICMC)*, pp. 193–198, 1984.
- [16] A. Volk, P. Kranenburg, J. Garbers, F. Wiering, R. C. Veltkamp and L. P. Grijp, "A Manual Annotation Method for Melodic Similarity and the Study of Melody Feature Sets," *International Society for Music Information Retrieval (ISMIR)*, pp. 101–106, 2008.
- [17] Z. Zhu, "Probe into the Training Stages of Basic Piano Course," *Journal of Gui Yang Teacher's College*, 2005.
- [18] A. Cheema and R. Bagchi, "The Effect of Goal Visualization on Goal Pursuit: Implications for Consumers and Managers," *Journal of Marketing*, Vol. 75, pp. 109–123, 2011.
- [19] PTNA Piano Encyclopedia, "Burgmüller, Johann Friedrich Franz:25 Etudes faciles et progressives, composées et doigtées expressément pour l'étendue des petites mains Op.100" [online]. Available: <https://enc.piano.or.jp/musics/680>. Access date: 30 July 2024.
- [20] J. Bastien, "How To Teach Piano Successfully, Third Edition." N.A. Kjos Music Co., 1988.
- [21] F. Kurth, M. Müller, C. Fremerey, Y. Chang and M. Clausen, "Automated Synchronization of Scanned Sheet Music with Audio Recordings," *International Society for Music Information Retrieval (ISMIR)*, pp. 261–266, 2007.
- [22] T. Nakamura, E. Nakamura and S. Sagayama, "Real-Time Audio-to-Score Alignment of Music Performances Containing Errors and Arbitrary Repeats and Skips," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, Vol. 24, Issue 2, pp. 329–339, 2015.

- [23] E. Nakamura, K. Yoshii, and H. Katayose, “Performance Error Detection and Post-Processing for Fast and Accurate Symbolic Music Alignment.” *International Society for Music Information Retrieval (ISMIR)*, pp. 347–353, 2017.
- [24] P. Silvan, “Online Symbolic Music Alignment with Offline Reinforcement Learning.” *International Society for Music Information Retrieval (ISMIR)*, pp. 5–13, 2023.

INNER METRIC ANALYSIS AS A MEASURE OF RHYTHMIC SYNCOPATION

Brian Bemman

Durham University

brian.m.bemman@durham.ac.uk

Justin Christensen

University of Sheffield

j.christensen@sheffield.ac.uk

ABSTRACT

Inner Metric Analysis (IMA) is a method for symbolic music analysis that identifies strong and weak metrical positions according to coinciding periodicities within note onsets. These periodicities are visualized with bar graphs known as *metric weight* and *spectral weight profiles*. Analyzing these profiles for the presence of syncopation has thus far required manual inspection. In this paper, we propose a simple measure using chi-squared distance for quantifying the level of syncopation found in IMA weight profiles by considering each as a distribution to be compared against (1) a uniform distribution ‘nominal’ weight profile, and (2) a non-uniform distribution based on beat strength. We apply this measure to the task of predicting perceptual ratings of syncopation using the Song (2014) dataset of 111 single-bar rhythmic patterns and compare its performance to seven existing models of syncopation/complexity. Our results indicate that the proposed measure based on (1) achieves a moderately high Spearman rank correlation ($r_s = 0.80$) to all ratings and is the only single measure that reportedly works across all categories. For so-called polyrhythms in 4/4, the measure based on (2) surpasses all other models and further outperforms five models for monorhythms in 6/8 and three models for monorhythms in 4/4.

1. INTRODUCTION

Much research has gone into understanding the perception of temporal patterns [1–3] and many more recent studies within this scope have focused on the perceived levels of syncopation and complexity in these patterns [4–11]. Subsequently, a number of different computational methods have been proposed for modeling these, including models that are based on a metric hierarchy using tree-based structures [7, 12–14] and those that are not [15–19]. Many of these models have been tested on various perceptual tasks, such as syn-

copation prediction, and their respective performances have been compared [6, 9, 20–23]. However, none of the comparisons carried out to date have considered Inner Metric Analysis [24].

Inner Metric Analysis (IMA) is a method of symbolic music analysis for identifying strong and weak metrical positions in a piece based on coinciding periodicities found in its note onsets [24]. Over the years, IMA has been applied to the tasks of automatic meter detection [25] and dance music classification [26], but it has largely been used in more traditional music analysis contexts [24, 27]. An important feature of IMA is its ability to provide a representation of the *inner* metric structure of a piece rather than a representation tied to its *outer* metric structure—the meter as indicated by the time signature in a score. This feature allows IMA to identify, for example, instances where the notated music conflicts with the implied or perceived meter. For this reason, it has been used to aid in the identification of syncopation [24], which has typically been defined as a temporary displacement of the regular metrical accent [28]. However, until now the use of IMA to identify syncopation in a musical passage has required manual analysis by a music theorist or other domain expert.

In this paper, we propose using chi-squared distance as a first step towards computing a quantifiable measure of syncopation from weight profiles produced by IMA. We apply this method to the task of predicting perceptual ratings of syncopation in the Song (2014) [22] dataset containing 111 one-bar rhythmic patterns in two different meters and rhythm types (i.e., monorhythms in 4/4, monorhythms in 6/8, and so-called polyrhythms in 4/4). In section 2, we explain how IMA produces a metrical analysis of a musical passage and detail the rhythmic patterns in the Song (2014) dataset. In section 3, we introduce our proposed measure based on chi-squared distance for comparing the weight profiles produced by IMA to a uniform distribution or ‘nominal’ weight profile as well as to a non-uniform distribution based on beat strength [29]. We evaluate this measure in section 4 by testing it on the aforementioned dataset and compare its performance to the reported performances of seven existing models of syncopation/complexity. We summarize our findings in section 5 and suggest possible directions for future work.



© B. Bemman, and J. Christensen. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** B. Bemman, and J. Christensen, “Inner Metric Analysis as a Measure of Rhythmic Syncopation”, in *Proc. of the 25th Int. Society for Music Information Retrieval Conf.*, San Francisco, United States, 2024.



Figure 1. Opening bars of the “Twinkle, Twinkle Little Star” melody and single local meter (A) with its pulses (black circles) generated by Inner Metric Analysis (IMA). Note that stars denote onsets (On).

2. RELATED WORK

2.1 Inner Metric Analysis (IMA)

IMA computes, from the note onsets of a piece, an exhaustive listing of *local meters*—each of which must be a sub-sequence of onsets or *pulses* that are (1) at least 3 in number, (2) separated by a fixed inter-onset interval, called the *period*, (3) not able to be extended further (forwards or backwards in time) within the sequence of all onsets of the piece, and (4) not contained within the pulses belonging to any other local meter. Figure 1 shows the opening two bars of “Twinkle, Twinkle Little Star” with its single local meter. Note that the single local meter (A) contains at least 3—in this case, 7, evenly-spaced pulses (black circles), each aligned with a corresponding onset in the music above. The numbers at the bottom indicate the positions within an underlying “grid” called *time points*, equivalent to *tatums*, upon which the onsets are fitted. Because all adjacent onsets have an equal, constant spacing, represented in the score as quarter-note durations, no time point exists between them. In such passages, there will be only a single local meter as any other possible set of pulses would necessarily be contained within this local meter. For more complex rhythms, this will not be the case.

Following the enumeration of all local meters in a piece, IMA computes a *metric weight* for each onset based on the number of pulses in local meters that coincide with this onset and the lengths of those local meters to which these pulses belong. Formally, let On be the set of all onset time points in a piece and m be a local meter that contains an onset, o , and where k_m denotes the length of m minus 1. $M(l)$ denotes the set of all local meters of length at least l , where in straight-forward implementations of IMA, l is 2 (equal to the minimum number of pulses, 3, minus 1). The metric weight of o is defined as the sum of the values, k_m , of the local meters, m , that contain o . The metric weight of an onset $o \in On$ is thus given by

$$W_{l,p}(o) = \sum_{\{m \in M(l): o \in m\}} (k_m)^p, \quad (1)$$

where p is a weighting parameter typically set to $p = 2$ that is used to control the relative influence of the length of local meters on the metric weight [24]. For example, the metric weight assigned to each of the 7 onsets of the melody shown in Figure 1, using the

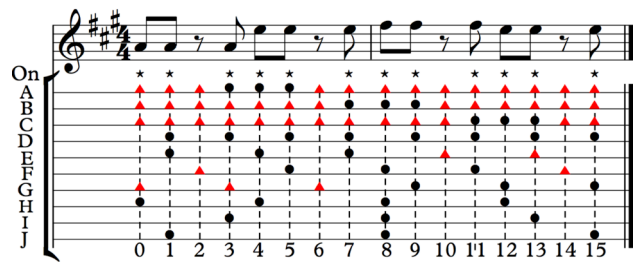


Figure 2. Opening bars of a syncopated variation of the “Twinkle, Twinkle Little Star” melody with all ten (A–J) local meters and their extensions (red triangles) generated by Inner Metric Analysis (IMA).

typical weighting parameter of $p = 2$, would be $(7 - 1)^2 = 36$, as each onset has only a single pulse (i.e., no coinciding pulses) belonging to one local meter of length $7 - 1 = 6$.

In addition to the metric weight, IMA also computes a *spectral weight* that considers the *extension* of each local meter to certain time points on the grid that align with either onsets or silence (i.e., rests) in a piece. Formally, an extension, $ext(m)$, of a local meter, m , is defined as the set of time points, $\{s + id, \forall i\}$, where s denotes the time point of the first onset in m , d is the period, and i is an integer time point in the underlying grid. Figure 2 shows a syncopated variation of the melody shown in Figure 1 with all ten of its local meters (A–J) and extensions (red triangles). Note that, in contrast to Figure 1, there are multiple local meters (ten) where no one local meter is contained within the pulses belonging to any other local meter. Take, for example, local meter (E), which shares two of its pulses (time points 1 and 7) with local meter (D), but contains a third (time point 4) that is not shared with (D). The purpose of extensions in IMA is to allow for pulses to contribute to parts of passages where they are not even present. The case for projecting pulses further in time in this way through extensions, for example, is made stronger when one considers the possibility for some latent or persisting pulse in the listener’s perception. The spectral weight is computed in a similar manner to the metric weight (shown in Equation 1), except that it assigns a weight to each time point (rather than only to each onset) based on the pulses and now extensions which coincide with this time point. For a time point, t , the spectral weight is given by

$$SW_{l,p}(t) = \sum_{\{m \in M(l): t \in ext(m)\}} (k_m)^p. \quad (2)$$

Whereas the metric weight of, for example, the first onset (at time point 0) shown in Figure 2, would consider only local meter (H) due to it having the only coinciding pulse, the spectral weight would consider the additional contributions of local meters (A, B, C, G), due to their coinciding extensions.

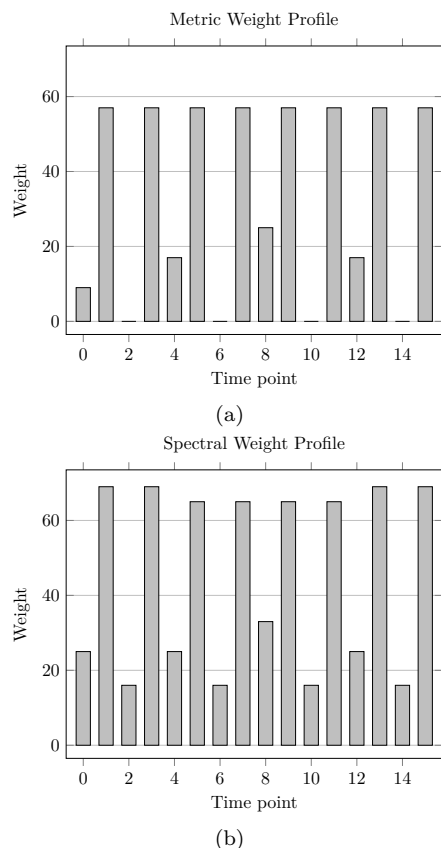


Figure 3. The metric and spectral weight profiles of the opening two bars of the syncopated “Twinkle, Twinkle Little Star” melody from Figure 2 with metric weights shown in (a) and spectral weights shown in (b), as computed by Inner Metric Analysis (IMA).

Whether an analysis of a piece by IMA uses metric weights or spectral weights, it is typical for the weights to be plotted in the form of a bar graph called a *profile*. With a trained eye, musical features of a piece such as a possible meter (whether notated or not) and syncopation often emerge through visual inspection of the profile. Figure 3 shows the *metric weight profile* and *spectral weight profile* for the opening two bars of the syncopated “Twinkle, Twinkle Little Star” melody shown in Figure 2. Given that the actual meter of the syncopated melody in Figure 2 is known, we can see in its corresponding weight profiles, shown in Figure 3, that all lower weights at time points 0, 2, 4, 6, 8, 10, 12, 14 appear at on-beat locations while all higher weights at time points 1, 3, 5, 7, 9, 11, 13, 15 appear at offbeat locations, suggesting a strong possibility for the presence of syncopation.

2.2 The Song (2014) Syncopation Dataset

Datasets containing rhythmic (or temporal) patterns for studying human perception remain relatively few in number and small in size [1–4, 7, 30, 31]. A number of these early datasets were originally constructed as a means for evaluating perceptual complexity and have since been co-opted for the study of syncopation [5,

6]. Even fewer datasets exist for the explicit study of syncopation [7, 22, 32, 33], however, the Song (2014) [22] dataset is arguably one of the largest.

The Song (2014) [22] dataset contains 111 single-bar rhythmic patterns (and their mean listener perceptual ratings from 0 to 4) in two possible meters, 4/4 and 6/8, and of two different rhythm types, mono and poly.¹ There are 27 monorhythm patterns in 4/4 (15 on quarter-note grid; 12 on eighth-note grid), 36 monorhythm patterns in 6/8 (eighth-note grid), and 48 so-called polyrhythm patterns in 4/4 (quarter-note triplet grid)—each of which were preceded for listeners by an audible two-bar metronome in their respective meter. Patterns in each category range from having a single onset (e.g., $\langle 0, 0, 0, 1 \rangle$ monorhythm in 4/4 with an onset on the fourth beat) up to a number of onsets equal to the number of time points in the underlying grid (e.g., $\langle 1, 1, 1, 1, 1, 1 \rangle$ monorhythm in 6/8 of all eighth notes).

3. PROPOSED MEASURE OF SYNCOPATION USING IMA

The central premise motivating our proposed measure is the consideration of weight profiles produced by IMA as distributions through which comparisons with other distributions using chi-squared distance [34] can provide insight into the underlying rhythmic structure that is relevant to predicting syncopation. We consider two possible distributions that we will compare against the weight profiles produced by IMA for a given rhythmic pattern: (1) a uniform distribution based on what we call a ‘nominal’ weight profile that operates conservatively and in the spirit of IMA without knowledge of the underlying meter, and (2) a non-uniform distribution based on beat strength [29] that operates with explicit knowledge of the underlying meter and is nearly analogous to a nominal weight profile but for metrical (hierarchy) structure. A nominal weight profile is the uniform distribution of weights that results from an IMA analysis of any sequence consisting entirely of equally-spaced onsets irrespective of the hierarchical metrical level at which these are expressed. For example, a pattern in 4/4 consisting of all quarter notes, half-notes, or eighth notes will each result in a nominal weight profile. Our motivation for considering nominal weight profiles is based on a simplifying assumption that a less syncopated rhythmic pattern or passage of music will have more equal weighting across its weight profile than a more syncopated pattern or passage. This was observed, for example, in Figures 1 and 2, with the less syncopated melody containing a single local meter resulting in a metric weight profile containing at each onset a constant weight and its syncopated version containing multiple local meters resulting in weight pro-

¹ The complete Song (2014) [22] dataset and perceptual ratings can be found in the following repository: <https://code.soundsoftware.ac.uk/projects/syncopation-dataset>.

files (shown in Figure 3) containing a variable weight that fluctuates over time. Our motivation in (2) for considering non-uniform distributions based on beat strength comes from the fact that clearly not all rhythmic patterns consist of all equally-spaced onsets, and, much like previous models of syncopation, such as Weighted Note-to-Beat Distance (WNBD) [17] or the Longuet-Higgins and Lee (LHL) model [12], providing additional information beyond what is explicitly available in the onsets (e.g., beat locations or rests), can provide relevant (or indeed necessary) information for modelling or predicting syncopation.

The proposed measure adopts two different constructions for handling normalization across patterns and distributions, both of which we will use in our evaluation. The first of these constructions considers a given metric or spectral weight profile produced by IMA as a normalized distribution, P' , scaled to unit length and a second, un-normalized uniform distribution, Q , representing a nominal weight profile having some constant value, Q_i for all i . The measure-normalized (weighted) chi-squared distance, χ_{D1} , between two distributions P' and Q of length n (time points) is given by

$$\text{IMA}_{M,S}\chi_{D1} = \frac{1}{n} \sum_{i=0}^{n-1} \left(\frac{(P'_i - Q_i)^2}{(P'_i + Q_i)} \right)^a, \quad (3)$$

where a is a weighting parameter (discussed in section 4) and $\frac{1}{n}$ serves to normalize the distance by measure length. By calculating the distance of an observed weight profile from a nominal weight profile, we obtain a measure of the overall aperiodicity or irregularity of the rhythmic content (relative to the constant, Q_i , in the uniform distribution), where the higher the overall value, the greater the amount of perceived syncopation there is predicted to be. In principle, while the constant Q_i could be any rational value, for the purposes of this paper, we will utilize a constant value between $[0, 1]$ corresponding to the maximum upper and minimum lower ranges of the P' distribution. In addition to the a weighting parameter, an optimal constant value for Q_i will be learned in section 4.

Whereas the Q uniform distribution in Equation 3 was left un-normalized to allow for various constant values to be learned, which would otherwise disappear with normalization, other distributions, such as our non-uniform distribution based on beat strength, will require normalization for fair comparisons with P' . Thus, an alternative weighted construction, χ_{D2} , of Equation 3 appears below for the same normalized distribution, P' , and another normalized distribution, S' , also of length n and scaled to unit length:

$$\text{IMA}_{M,S}\chi_{D2} = \sum_{i=0}^{n-1} \left(\frac{(P'_i - S'_i)^2}{(P'_i + S'_i)} \right)^a, \quad (4)$$

where a is the same weighting parameter as in

the earlier construction. Note that because both distributions have been scaled to unit length, normalizing by measure length, n , as was done in Equation 3, is no longer necessary. In our use of Equation 4, we consider four different distributions for S' , corresponding to the beat strengths produced by music21 [29] (using the `beatStrength` method) for each of the four different types of meter/rhythm types found in the Song (2014) [22] dataset. The following four (un-normalized) beat strength distributions are those used with this construction: (1) 4/4 meter with quarter-note grid $\langle 1.0, 0.25, 0.5, 0.25 \rangle$, (2) 4/4 meter with eighth-note grid $\langle 1.0, 0.125, 0.25, 0.125, 0.5, 0.125, 0.25, 0.125 \rangle$, (3) 6/8 meter with eighth-note grid $\langle 1.0, 0.25, 0.25, 0.5, 0.25, 0.25 \rangle$, and (4) 4/4 meter with quarter-note triplet grid $\langle 1.0, 0.0625, 0.0625, 0.25, 0.0625, 0.0625, 0.5, 0.0625, 0.0625, 0.25, 0.0625, 0.0625 \rangle$.

4. EVALUATION

We have evaluated our IMA-based measure on the Song (2014) [22] dataset of 111 one-bar rhythmic patterns and their perceptual ratings of syncopation for three reasons: (1) there is a relatively large number of stimuli in comparison to other available datasets, (2) the stimuli were constructed specifically for the purpose of studying syncopation and not, for example, complexity or groove, and (3) there has been significant work already done on evaluating other computational models of syncopation with this dataset. The reader is referred to [22] for an in-depth discussion of the performances of existing models using this dataset.

4.1 Procedure

We have adopted an optimization approach using leave-one-out cross-validation in which we performed a grid search over the pair of parameters, Q_i , and a from Equation 3 for 100^2 value-pairs within the range $[0, 1]$ in increments of 0.01. For each distinct weighting parameter pair, we carried out the procedure below for all rhythmic patterns in the training set:

1. Repeat the time-span note sequence of the given Song (2014) [22] one-bar rhythmic pattern twelve times. As IMA requires at least three pulses to form a local meter, it is generally less effective with short rhythmic patterns having few onsets. For this reason, it has been suggested in [26] to repeat short patterns in this way when using IMA.
2. Convert this extended time-span note sequence from (1) to an ordered set of note onset indices suitable for analysis by IMA e.g., $\langle 0, 1, 0, 1 \rangle$ to $\langle 1, 3 \rangle$ (using 0-based indexing).
3. Compute IMA metric and spectral weight profiles for this extended twelve-bar rhythmic pat-

tern from (2) using an IMA weighting parameter $p = 2$ and minimum local meter length, $l = 2$.

4. ‘Fold’ the metric and spectral weight profiles in (3) into single bars and sum those weights at each time point having equivalent within-bar locations. Then scale each weight profile to unit length such that they each sum to 1.
5. Compute a measure of syncopation from each normalized single-bar metric and spectral weight profile from (4) using Equation 3 and the given weighting parameter pair, Q_i and a .

Following the procedure above for all training rhythmic patterns and a given weighting parameter pair, the respective sets of syncopation scores computed for each of the metric and spectral weight profiles are min-max normalized. The Spearman rank correlation coefficient, r_s , is then computed for each of these sets of syncopation scores and the min-max normalized mean perceptual ratings, in the same way that was done for each of the computational models evaluated in [22, pp. 92–94] so that fair comparisons could be made. The procedure for using Equation 4 and the non-uniform distributions based on beat strength is identical to the steps outlined above, except the grid search was performed across all 100 values between $[0, 1]$ for a only, and the set of beat strengths chosen for any given pattern was that matching in number of time points, n . The weight parameter (Equation 4) or weight parameter pair (Equation 3) that produced the highest mean rank correlation achieved across all k -folds was retained and the final results below are reported using the best parameters across the entire dataset. All syncopation-dependent procedures were implemented in Julia (v. 1.10.0) and all statistical calculations were made with R (v. 4.3.2).

4.2 Results and Discussion

We compare the performance of the proposed measure to the reported performances of seven models of syncopation/complexity previously evaluated in [22] and [21, 23] on the same dataset. These models are Longuet-Higgins and Lee (LHL) [12], Off-Beatness (TOB) [16], Metric Complexity (TMC) [14], Weighted Note-to-Beat Distance (WNBD) [17], Cognitive Complexity (PRS) [13], Off-beat model (KTH) [15], and Sioros et al. (SG) [7]. Table 1 shows the results of our IMA-based measure of syncopation for both the metric and spectral weight profiles using the two proposed distributions across the dataset in comparison to these other models.

In Table 1, the best weighting parameters found for Equation 3 (χ_{D1}) were $a = 1.0$ for both metric and spectral weight and $Q_i = 0.74$ for metric weight and $Q_i = 0.83$ for spectral weight. The best weighting parameters for Equation 4 (χ_{D2}) were $a = 0.82$

³ There may be disagreement as to whether polyrhythms in the Song (2014) [22] dataset are what they claim and whether some existing models are in fact incapable of analyzing these

	Model/Measure	Rhythm Type & Meter			Total
		Mono $\frac{4}{4}$	Mono $\frac{6}{8}$	Poly $\frac{4}{4}$	
1.	IMA $_M\chi_{D1}$	0.53*	0.67*	0.46*	0.80*
2.	IMA $_S\chi_{D1}$	0.51*	0.67*	0.39*	0.79*
3.	IMA $_M\chi_{D2}$	0.86*	0.74*	0.73*	0.66*
4.	IMA $_S\chi_{D2}$	0.83*	0.74*	0.70*	0.61*
5.	LHL	0.86*	0.68*	-	
6.	TMC	0.92*	0.67*	-	
7.	PRS	0.95*	0.76*	-	
8.	SG	0.88*	0.73*	-	
9.	TOB	0.36	0.17	NA	
10.	WNBD	0.52*	0.47*	0.41*	
11.	KTH	0.79*	-	-0.23	

Table 1. Spearman correlation rank coefficients (r_s) of 9 different models/measures of syncopation for 111 mono- and poly-rhythmic patterns in two meters and their perceptual ratings. For the proposed measures based on IMA (1–4), IMA $_M$ and IMA $_S$ denote use of metric and spectral weight, respectively. Note that results for models 5–11 are the values reported in [22, pp. 92–94]. An asterisk denotes where $p < 0.01$, a hyphen indicates where a given measure is reported as being incapable of providing a result³, and empty cells mark no reported results.

for metric weight and $a = 0.35$ for spectral weight. It is clear from these results that while Equation 3 worked best across the entirety of the dataset (e.g., IMA $_M\chi_{D1}$: $r_s = 0.80^*$; $p < 0.01$ —an improvement over no a weighting parameter and Q_i set to an arbitrary 0.5, $r_s = 0.73^*$; $p < 0.01$), it resulted in relatively poor performance within the individual categories. Perhaps not surprisingly, however, providing additional information about the underlying meter, in the form of beat strengths as done in Equation 4, significantly improved the performance in these individual categories but to the detriment of overall performance (e.g., IMA $_M\chi_{D2}$: $r_s = 0.66^*$; $p < 0.01$ —same as without a). In all cases except monorhythms in 6/8, metric weight performed better than spectral weight. In particular, IMA $_M\chi_{D2}$, outperformed all three of the existing models (TOB, WNBD, KTH) reportedly capable of providing a result for the so-called polyrhythms in 4/4 ($r_s = 0.73^*$; $p < 0.01$); five of the existing models (LHL, TMC, SG, TOB, WNBD) capable of providing a result for monorhythms in 6/8 ($r_s = 0.74^*$; $p < 0.01$); and only three (TOB, WNBD, KTH) of all seven models for monorhythms in 4/4 ($r_s = 0.86^*$; $p < 0.01$; tying with LHL). It should be noted that in [22, p. 139] a so-called weighted-multiple combined (WMC) model using optimized versions of the best combinations of these previous models was able to achieve a rank correlation across the entire dataset of $r_s = 0.89^*$ ($p < 0.01$). While the proposed IMA measure falls short in this regard

without reinterpreting their meter (e.g., 4/4 to 12/8). The reader is referred to [22] for detail on these possible concerns.

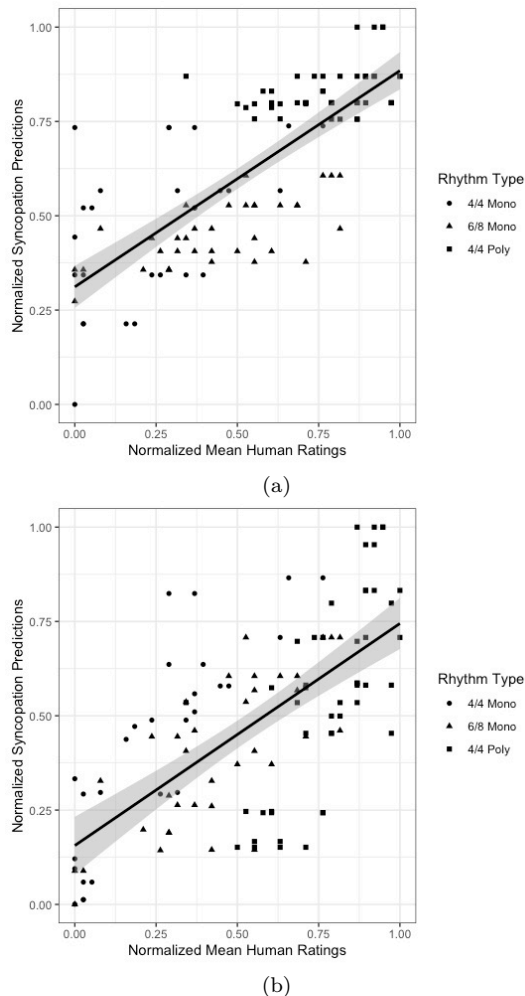


Figure 4. Normalized syncopation predictions produced by $\text{IMA}_M\chi_{D1}$ ($r_s = 0.80^*$; $p < 0.01$) in (a) and $\text{IMA}_M\chi_{D2}$ ($r_s = 0.66^*$; $p < 0.01$) in (b) against the normalized human ratings for the 111 rhythmic patterns in the Song (2014) [22] dataset. Note that regression lines are plotted with shaded areas indicating a 95% confidence interval.

($r_s = 0.80^*$; $p < 0.01$), it remains noteworthy that its performance is close to approaching the performance of a significantly more complex method consisting of many different models. For completeness, Figure 4 shows regression plots of the predicted syncopation scores across the entire dataset for both $\text{IMA}_M\chi_{D1}$ and $\text{IMA}_M\chi_{D2}$ against the human ratings.

The reason for the difference in performances for both constructions across the dataset versus within the individual categories is not immediately clear, however, the use of rank correlation combined with the distributed locations and smaller sizes of the respective sets of rhythm and meter type examples within the dataset is likely a contributing factor. Despite the better overall performance of Equation 3 over Equation 4, one problem with our first construction using this dataset concerns the density of pattern onsets, which has been shown to interact with their perceived displacement from a metrical hierar-

chy in regards to syncopation [33]. Many of the patterns are highly sparse, and Equation 3 is unable to differentiate, for example, between two distinct patterns each having a single onset, such as $\langle 1, 0, 0, 0 \rangle$ and $\langle 0, 1, 0, 0 \rangle$, or the same number of equally-spaced onsets shifted in time, such as $\langle 1, 0, 1, 0 \rangle$ and $\langle 0, 1, 0, 1 \rangle$. This would help to explain its relatively low performance in the individual categories. Equation 4, on the other hand, does not encounter these same difficulties, and its improved performance in the individual categories suggests an informative structural correspondence between the metrical strengths as identified by IMA weight profiles and the beat strengths they were compared against. In an actual piece of music, however, one might expect to find relatively less sparse and more complex examples, so more ecologically valid comparisons may provide deeper insights into whether syncopated patterns have generally less equal weighting in their profiles as un-syncopated patterns, as is assumed by Equation 3. Finally, while the choice of chi-squared distance is motivated by the desire to obtain the best possible results across the entirety of the dataset using the simplest method, multiple other distance measures were tested (e.g., Euclidean and Minkowski) with the relatively more complex Jensen-Shannon divergence [35] performing marginally better across the dataset ($r_s = 0.81^*$; $p < 0.01$) but marginally worse within the individual categories.

5. CONCLUSION

In this paper, we proposed a first step towards using Inner Metric Analysis (IMA) to provide a quantifiable measure of syncopation based on chi-squared distance and comparisons to two different types of distributions. We evaluated our method using a dataset of rhythmic patterns constructed specifically for the task of studying syncopation and compared its performance to the performances of seven existing computational models. Our results indicate that the proposed measure based on comparisons with a uniform distribution achieves a moderately high Spearman rank correlation ($r_s = 0.80$) to all perceptual ratings and is the only single measure that reportedly works across all meters and rhythm types (mono, poly, 4/4 and 6/8). For so-called polyrhythms in 4/4, the measure based on comparisons with a distribution of beat strengths surpasses all other models and further outperforms five models for monorhythms in 6/8 and three models for monorhythms in 4/4. Finally, considering the entirety of a rhythmic sequence as done here rather than summing isolated instances of syncopation as in, for example, the LHL [12] model, appears to have higher validity [36]. In future work, it would be useful to consider other datasets, particularly ones which contain more ecologically valid examples, as well as with other distributions, possibly coming from statistical corpora studies or perceptual profiles, that could be automatically selected for in comparisons.

6. ACKNOWLEDGMENTS

We would like to thank Tuomas Eerola, Mark Gotham, and the anonymous reviewers for their helpful suggestions during the later stages of this paper as well as David Meredith for useful feedback provided during its early stages.

7. REFERENCES

- [1] P. J. Essens and D. J. Povel, “Metrical and non-metrical representations of temporal patterns,” *Perception & Psychophysics*, vol. 37, no. 1, pp. 1–7, 1985.
- [2] D.-J. Povel and P. J. Essens, “Perception of temporal patterns,” *Music Perception: An Interdisciplinary Journal*, vol. 2, no. 4, pp. 411–440, 1985.
- [3] P. J. Essens, “Structuring temporal sequences: Comparison of models and factors of complexity,” *Perception & Psychophysics*, vol. 57, no. 2, pp. 519–532, 1995.
- [4] I. Shmulevich and D. J. Povel, “Measures of temporal pattern complexity,” *Journal of New Music Research*, vol. 29, no. 1, pp. 61–69, 2000.
- [5] L. M. Smith and H. Honing, “Evaluating and extending computational models of rhythmic syncopation in music,” in *Proceedings of the International Computer Music Conference*, 2006, pp. 688–691.
- [6] F. Gómez, E. Thul, and G. T. Toussaint, “An experimental comparison of formal measures of rhythmic syncopation,” in *Proceedings of the International Computer Music Conference*, 2007, pp. 101–104.
- [7] G. Sioros, M. Miron, M. Davies, F. Gouyon, and G. Madison, “Syncopation creates the sensation of groove in synthesized music examples,” *Frontiers in Psychology*, vol. 5, p. 1036, 2014.
- [8] P. Vuust, M. J. Dietz, M. Witek, and M. Kringelbach, “Now you hear it: a predictive coding model for understanding rhythmic incongruity,” *Annals of the New York Academy of Sciences*, vol. 1423, pp. 19–29, 2018.
- [9] F. Hoesl and O. Senn, “Modelling perceived syncopation in popular music drum patterns: A preliminary study,” *Music & Science*, vol. 1, p. 2059204318791464, 2018.
- [10] D. Temperley, “Modeling common-practice rhythm,” *Music Perception: An Interdisciplinary Journal*, vol. 27, no. 5, pp. 355–376, 2010.
- [11] M. Witek, E. F. Clarke, M. L. Kringelbach, and P. Vuust, “Effects of polyphonic context, instrumentation and metric location on syncopation in music,” *Music Perception: An Interdisciplinary Journal*, vol. 32, pp. 201–217, 2014.
- [12] H. Longuet-Higgins and C. Lee, “The rhythmic interpretation of monophonic music,” *Music Perception: An Interdisciplinary Journal*, vol. 4, no. 1, pp. 424–441, 1984.
- [13] J. Pressing, “Cognitive complexity and the structure of musical patterns,” in *Proceedings of the 4th Conference of the Australian Cognitive Science Society*, 2002.
- [14] G. T. Toussaint, “A mathematical analysis of african, brazilian, and cuban clave rhythms,” in *Proceedings of BRIDGES: Mathematical Connections in Art, Music and Science*, 2002, pp. 157–168.
- [15] M. Keith, *From Polychords to Pólya: Adventures in Music Combinatorics*. Princeton: Vinculum Press, 1991.
- [16] G. T. Toussaint, “Mathematical features for recognizing preference in sub-saharan african traditional rhythm timelines,” in *3rd International Conference on Advances in Pattern Recognition*, 2005, pp. 18–27.
- [17] F. Gómez, A. Melvin, D. Rappaport, and G. T. Toussaint, “Mathematical measures of syncopation,” in *Renaissance Banff: Mathematics, Music, Art, Culture*, 2005, pp. 73–84.
- [18] M. E. P. D. George Sioros and C. Guedes, “A generative model for the characterization of musical rhythms,” *Journal of New Music Research*, vol. 47, no. 2, pp. 114–128, 2018.
- [19] M. Rohrmeier, “Towards a formalization of musical rhythm,” *Proceedings of the 21st International Society for Music Information Retrieval*, pp. 621–629, 2020.
- [20] E. Thul and G. T. Toussaint, “Rhythm complexity measures: A comparison of mathematical models of human perception and performance,” in *Proceedings of the 9th International Society for Music Information Retrieval Conference (ISMIR)*, 2008, pp. 663–668.
- [21] C. Song, A. J. R. Simpson, C. A. Harte, M. T. Pearce, and M. B. Sandler, “Syncopation and the score,” *PLoS ONE*, vol. 8, no. 9, pp. 1–7, 2013.
- [22] C. Song, “Syncopation: Unifying music theory and perception,” PH.D. diss., Queen Mary, University of London, 2014.
- [23] C. Song, M. T. Pearce, and C. A. Harte, “Synpy: A python toolkit for syncopation modelling,” in *Proceedings of the 12th International Conference on Sound and Music Computing (SMC15)*, 2015, pp. 295–300.

- [24] A. Volk, "The study of syncopation using Inner Metric Analysis: Linking theoretical and experimental analysis of metre in music," *Journal of New Music Research*, vol. 37, no. 4, pp. 259–273, 2008.
- [25] W. B. De Haas and A. Volk, "Meter detection in symbolic music using inner metric analysis," in *Proceedings of the 17th International Society for Music Information Retrieval Conference*, 2016, pp. 441–447.
- [26] E. Chew, A. Volk, and C. Y. Lee, "Dance music classification using inner metric analysis," in *The Next Wave in Computing, Optimization, and Decision Technologies*, B. Golden, S. Raghavan, and E. Wasil, Eds. Boston, MA: Springer US, 2005, pp. 355–370.
- [27] A. Volk, "Applying inner metric analysis to 20th century compositions," in *Proceedings of the 1st Conference of the Society for Mathematics and Computation in Music, MCM 2007, Berlin, Germany, May 18–20, 2007*, 2007, pp. 204–210.
- [28] A. Latham, *The Oxford companion to music*. Oxford University Press, 2011.
- [29] Michael Scott Asato Cuthbert, "Music 21: A toolkit for computer-aided musicology." [Online]. Available: <https://web.mit.edu/music21/>
- [30] W. T. Fitch and A. J. Rosenfeld, "Perception and production of syncopated rhythms," *Music Perception: An Interdisciplinary Journal*, vol. 25, no. 1, pp. 43–58, 2007.
- [31] P. E. Keller and E. Schubert, "Cognitive and affective judgements of syncopated musical themes," *Advances in Cognitive Psychology*, vol. 7, pp. 142–156, 2011.
- [32] G. Sioros, A. Holzapfel, and C. Guedes, "On measuring syncopation to drive an interactive music system," in *Proceedings of the 13th International Society for Music Information Retrieval Conference (ISMIR)*, 2012, pp. 283–288.
- [33] N. R. Fram and J. Berger, "Syncopation as probabilistic expectation: Conceptual, computational, and experimental evidence," *Cognitive Science*, vol. 47, no. 12, p. e13390, 2023.
- [34] W. G. Cochran, "The χ^2 test of goodness of fit," *The Annals of mathematical statistics*, pp. 315–345, 1952.
- [35] J. Lin, "Divergence measures based on the shannon entropy," *IEEE Transactions on Information theory*, vol. 37, no. 1, pp. 145–151, 1991.
- [36] G. Sioros, G. Madison, D. Cocharro, A. Danielsen, and F. Gouyon, "Syncopation and Groove in Polyphonic Music: Patterns Matter," *Music Perception*, vol. 39, no. 5, pp. 503–531, 06 2022.

HAISP: A DATASET OF HUMAN–AI SONGWRITING PROCESSES FROM THE AI SONG CONTEST

Lidia Morris^{1*} Rebecca Leger^{2*} Michele Newman¹
John Ashley Burgoyne³ Ryan Groves⁴ Natasha Mangal⁵ Jin Ha Lee¹

¹ University of Washington, USA ² Fraunhofer IIS, Germany

³ University of Amsterdam, Netherlands ⁴ Infinite Album, Switzerland ⁵ CISAC, France

ljmorris@uw.edu, rebecca.leger@iis.fraunhofer.de, mmn13@uw.edu,

j.a.burgoyne@uva.nl, ryan@infinitealbum.io, natasha.mangal@cisac.org, jinhalee@uw.edu

* denotes shared first authorship

ABSTRACT

The advent of accessible artificial intelligence (AI) tools and systems has begun a new era for creative expression, challenging us to gain a better understanding of human-AI collaboration and creativity. In this paper, we introduce Human–AI Songwriting Processes Dataset (HAISP), consisting of 34 coded submissions from the 2023 AI Song Contest. This dataset offers a resource for exploring the complex dynamics of AI-supported songwriting processes, facilitating investigations into the possibilities and challenges posed by AI in creative endeavors. Overall, HAISP is anticipated to contribute to advancing understanding of human-AI co-creation from the users’ perspective. We suggest potential use cases for the dataset, including examining AI tools used in songwriting and exploring users’ ethical considerations and creative approaches. This could help inform academic research and practical applications in music composition and related fields.

1. INTRODUCTION

Open and easy to access artificial intelligence (AI) technologies have created new opportunities for creativity, challenging conventional notions of authorship, expression, and human-AI creativity [1]. Within this landscape, the AI Song Contest (AISC) has emerged as a unique platform where teams of musicians, data scientists, researchers and AI enthusiasts can leverage AI tools to compose original songs, providing a prolific ground for studying the interplay between human creativity and machine intelligence [2].

In this paper we present the Human-AI Songwriting

Processes Dataset (HAISP), a curated dataset extracted from the written process documentation of participants in the AI Song Contest. This dataset provides a useful resource for exploring various aspects of AI-supported songwriting processes. It consists of 34 submissions from the 2023 AISC teams, cleaned, organized, and cross-annotated by four annotators using our data dictionary. The HAISP dataset includes information on the AI systems utilized, creative and technical inspirations, methodologies for working with AI, teams’ assessments of the songs, and reflections on ethical considerations in AI-generated content. This dataset provides researchers with a unique perspective into the complex relationship between human creativity and AI assistance in songwriting. By analyzing how songwriting processes are affected by the use of AI tools, scholars can gain insights into how AI systems may augment, complement, or challenge creative endeavors. The dataset also supports investigations into the ethical aspects of AI-generated music, including considerations like diversity in training data, intellectual property rights, and accessibility in music creation. It can serve as a valuable resource for scholars, practitioners, and enthusiasts alike, fostering deeper understanding, critical inquiry, and informed discourse in the burgeoning field of human-AI collaboration and creativity. Overall, it provides various insights and opportunities for further research, contributing to our understanding of the interaction between technology and creativity in the digital age.

2. BACKGROUND

2.1 Human–AI Music Creation

Using computational methods for music creation that would be classified as AI today, began in the 1950s, with early examples including Iannis Xenakis using Markov Chains for composition [3] or David Cope’s *Experiments in Musical Intelligence* in the 1980s [4]. For more detail on the history of AI music we refer to *The Oxford Handbook of Algorithmic Music* by Roger Dean and Alex McLean [5]. Although the use of neural networks for mu-



© L. Morris, R. Leger, M. Newman, J. A. Burgoyne, R. Groves, N. Mangal and J. H. Lee. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** L. Morris, R. Leger, M. Newman, J. A. Burgoyne, R. Groves, N. Mangal and J. H. Lee, “HAISP: A Dataset of Human–AI Songwriting Processes from the AI Song Contest”, in *Proc. of the 25th Int. Society for Music Information Retrieval Conf.*, San Francisco, United States, 2024.

sic modeling was mentioned as early as 1989 [6], the recent progress in the field of deep learning led to an increase of powerful and easy-to-use AI tools for music creation [7], from accessible applications for a large audience [8], to tools intended for professional musicians (e.g. [9]).

2.2 AI and Music Information Retrieval

In the area of music composition, researchers have developed many methods [10] and machine learning-powered interfaces that enable interactive exploration of musical variations by mapping user inputs to musical structures. Other tools have recently emerged to assist in various aspects of the music creation process, including infilling missing parts of compositions [11–14], creating new instruments [15, 16], counterpoint improvisation [17], and generating and recommending chord progressions [18–20], harmonies [21–23], and even accompaniment [24, 25].

Many AI models have also been created to aid in music information retrieval (MIR) research. With the explosive growth of digital music archives and streaming platforms, the need for effective MIR systems has become increasingly pronounced, driving research efforts to develop more sophisticated methods for understanding and processing music data through AI, such as the utilization of deep learning, neural networks, and large language models [26–29]. Existing datasets in the realm of AI in MIR primarily focus on training data, such as audio features [30, 31], music [32, 33], and metadata [34, 35].

2.3 AI Song Contest

The AI Song Contest is an annual international music competition wherein teams from diverse musical traditions and disciplines collaborate to compose songs using AI methods. It was launched in 2020 by the Dutch public broadcaster VPRO [2], and has been organized independently afterwards every year. There have been 132 teams so far, with 35 from the 2023 edition included in the dataset. We plan to extend the dataset with the other year’s team entries in the future. To participate in the AI Song Contest 2023, teams had to submit their song, a team image, cover art, and an online form in which they described their team, their creative vision, their motivation to participate, the steps of composition, their impression of the human-AI co-creation process, their workflow, all AI tools and databases used and their ethical and cultural considerations. The form has been developed by the AI Song Contest organizing team, slightly modified for each new edition. After successful submission, the songs and process documents are sent to a jury. The top ten entries of the jury voting are open for a public online vote. Whichever team gets the most points from the jury voting combined with the public voting wins the AI Song Contest.

3. OBJECTIVE AND SCOPE

There is a growing recognition of the need for complementary data that provides insights into the qualitative aspects of human-AI collaboration in music creation – the

human side of training and using these AI tools and systems. Recently, researchers have called for a cultural and ethical turn in MIR [36]. Rezwana and Maher [37] and Lee et al. [38] emphasize the importance of understanding not only perspectives but also expectations and ethical concerns of users of AI tools.

To our knowledge, there is no dataset exploring the qualitative processes and reflections of those who have used AI tools for music creation. Recognizing this gap, we were motivated to curate the written submissions of AISC participants into a unique and publicly available dataset that would allow researchers to go behind the scenes and explore the AI-supported songwriting and creation processes of the teams, beyond the final song submissions. By complementing existing quantitative datasets in AI in MIR, the HAISP dataset contributes to a more holistic understanding of the role of AI in creative endeavors and facilitates deeper insights into the collaborative dynamics between human composers and AI systems.

However, there are limitations to the data in this dataset. Firstly, the dataset relies solely on the words of the 34 participants in their written (subjective) process documentation. Compared to other datasets in MIR publications, this qualitative dataset is of limited size which might not lead to conclusions that are broadly applicable. Additionally, the dataset may not capture the full spectrum of AI tools and methodologies employed by participants, as teams may choose not to disclose certain details or may use proprietary technologies. Furthermore, the dataset represents a snapshot of a specific event and a specific outcome, the song, which may not fully generalize to other contexts of AI-supported music creation such as AI tools for jazz improvisation (e.g. see *GenJam* [39]).

Nonetheless, the amount of detail and depth provides an extensive and rich insight into the creative experience of the teams. By compiling and analyzing the teams’ process documentation, we seek to illuminate the landscape of human-AI creative collaboration. Through the creation of this dataset, we aim to facilitate deeper insights into the collaborative dynamics between human composers and AI systems, thereby fostering a richer understanding of the potential, limitations, and societal implications of AI in music production.

4. DATASET DESCRIPTION

The HAISP Dataset is accessible as a .csv and .xlsx on OSF under a Creative Commons Attribution-NonCommercial 4.0 International (CC BY-NC) license, which allows for broad access and utilization for research purposes.¹

4.1 Data Collection

For submission to the contest each team had to fill in the AI Song Contest 2023 Submission Form via Google forms. The form consists of entry fields to upload the song and

¹ <https://creativecommons.org/licenses/by-nc/4.0/>

Category	Subcategory	Definition
Team Data	Team_ID	The given label of the team for the dataset.
	Number of Team Members	The number of team members in a team.
	Type of Affiliation	The given work or personal affiliation of the team members.
	Country	The given country(s) of origin of the team.
	Found Out About Contest	The way that the team discovered or were informed of the AI Song Contest.
	Motivation to Participate in the Contest	The reason the team gave for joining the competition.
Song Data	Song Length	The length of the submitted song given in minutes and seconds.
	Song Description	The short description of the song written for the contest website.
	Final Ranking	Denotes the final ranking of the team's song in the competition.
	Song Title	The title of the song submitted to the AI Song Contest.
	Song Concept	The overarching idea or theme of the song.
Process	Creation Process	The given order in which the separate pieces of the song, or potentially the whole song, was created, as described by the team.
Song Elements Use of AI	Melody	Whether an AI system was used to generate the melody.
	Harmony	Whether an AI system was used to generate the harmony.
	Bassline	Whether an AI system was used to generate the bassline.
	Drums	Whether an AI system was used to generate drum patterns or rhythms.
	Formal Structure	Whether an AI system was used to generate the formal structure.
	Lyrics	Whether an AI system was used to generate the lyrics of the song.
	Voice Synthesis	Whether an AI system was used to generate the singing voice of the song.
Song Process Use of AI	Idea Generation	Whether an AI system was used to generate the idea for the song.
	Composing/Arranging	Whether an AI system was used to organize the elements of the song.
	Evaluation	Whether an AI system was used to evaluate the output or final song.
	Mixing & Mastering	Whether an AI system was used to do the mixing and mastering of the song
	Performance	Whether an AI system is or would be used for live performance.
AI Tools Used	Model Used	The AI models as used and indicated by the teams in the song creation process.
	Database(s) Used	The databases as used and indicated by the teams in the training and song creation process.
Ethical Considerations	Diversity, Ethical, and Cultural Considerations	Ethical and cultural considerations stated by the teams regarding their process and use of AI.
Human Evaluation of AI Co-Creation	Evaluation of Output	The words that teams used to assess the output of the AI system(s).
	Evaluation of Process	The words that teams used to assess the process of working with the AI system(s).
	Ownership	Teams statements regarding ownership of the system output and/or final song.
	Motivation to Use AI	The reasons that teams mentioned why they used AI in the process.
Other	Other	Additional information that does not fit in another category.

Table 1. Categories of HAISP Dataset. The HAISP dataset consists of data collected in 31 categories in total.

visual material (team image and song cover), and a free-text and multiple choice questionnaire. The questions that the teams filled in covered everything in these categories:

- team (bio for the website, location, level of expertise, motivation to participate, how they heard about the AISC);
- song (title, length, link to music video/soundcloud/blogpost, concept/idea, lyrics, live performance);
- human–AI process (short description for the website, models and databases used, steps of the process, creative vision, capabilities/limitations of AI tools in the creative process, workflow, collaboration with team members and AI, input data, ownership and conflict with intellectual property law);
- and diversity, ethical and cultural considerations

All teams had to further give consent for their answers description to be published in a scientific paper.

The answers were collected automatically in a Google Sheet. In total there were 40 submissions collected with one being a corrective submission replacing an existing entry and four submissions being incomplete. After unanswered inquiry these submissions were excluded, leading to 35 participating teams in the 2023 edition. Their complete questionnaires were then handed over to the research group excluding any personal data. One process documentation (teamID: 2023_14) was submitted in Spanish which was excluded from the dataset for linguistic consistency.

4.2 Data Statistics

The HAISP dataset consists of the data from 34 teams of the 2023 edition of the AISC. Looking at the team data, there is a total of 104 team members involved, with an average of three members per team, and 14 countries represented. Of the countries represented, eight teams were based in the United States, six were based in the United Kingdom, and four were based in Guatemala, Sweden, and Germany. Other countries represented included South Korea, Spain, North Macedonia, and more.

Type of affiliation – the given work or personal affiliation of the team members – was determined by coders based on the data, with teams being assigned multiple affiliations based on team members. A majority of the team members (58.8%) were members of academic field (20 mentions), meaning they worked primarily within academic institutions; this included universities, archives, or museums. A partially overlapping 44.1% of the team members were artists or worked in the creative industries, 17.6% worked as researchers outside academia, and 8.8% were classified as independent, (i.e., working in fields unrelated to the study/research/creation of AI or the creation of music but rather participating out of their own curiosity, hobby, or interest).

Teams had primarily heard about the contest from academia (29%), web search, or participation in prior editions of the AISC. The motivation for why teams used AI

in their process and why they decided to take part in the competition can be described as exploratory, while seven teams were also participating in order to display the use of their own or institutionally-created software.

Regarding the AI tools, there were 74 different tools used by teams in the 2023 edition. On average, there were 2.17 tools used per team. Half of the teams used a form of GPT by OpenAI (e.g., ChatGPT) and 38.24% of the teams used tools by Google Magenta (e.g., Magenta Studio, Tensorflow, DDSP). Other tools that were frequently used were TransformerXL [40] (14.1%), AIVA (11.4%) or MusicGen by Meta (3 teams). There were 57 models that have been only used once.

Looking at the use of AI in the compositional process, illustrated in Figure 1, it shows that 21 teams used only AI or co-created with AI for arranging, 18 teams used AI for idea generation and 10 (would) use AI in a performance. Nine teams mentioned the use of AI for mixing/mastering while only four teams used AI for evaluation of the output. Co-creation means this part was created by the human working with AI, human that only human was involved, AI that only AI was involved. Failed attempts to use AI in a specific step of the process were mentioned only in one case. Interestingly, the number of using “AI only” for these steps are low, with most uses (8) for “idea generation”. Only one team, 2023_28, mentioned a failed attempt to use AI in a specific step of the process: mixing and mastering.

Looking at the use of AI to create the elements of the songs – melody, harmony, bassline, drums, formal structure, lyrics, voice – depicted in Figure 2, 29 teams used AI, either in co-creation or solely for melody, while 19 used it for harmony and lyrics. Interestingly, while for melody, harmony, bassline, and drums the co-creation outweighs the AI-only approach, this is the opposite for formal structure, lyrics and voice synthesis. Team 2023_20 was the only team reporting a failed use of AI for creating harmony and voice synthesis with AI.

The teams’ own writings on their song concepts, their described creation process, their evaluation of diversity and ethical issues within said process, their thoughts on ownership of the song, and their overall evaluation of their human-AI co-creation process and the outcome of said process can be found in the dataset and can be used for further analysis.

4.3 Methodology and Validation

After reviewing the initial survey responses, we proceeded to create the data dictionary through a mixed-methods approach. The categories were created inductively by five researchers, iterating four times to reach final consensus. For each iteration of the data dictionary, two coders tested it on two sample entries to ensure that the categories were properly defined and applicable for the data.

We generated the dataset via consensus coding [41]. One researcher coded a selection of the data entries, collecting them into the dataset. A second coder then reviewed the initial coding, validating the coding by either

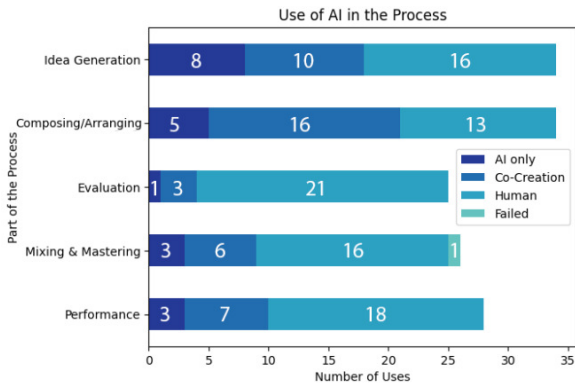


Figure 1. Overview of the Use of AI for the Compositional Process in HAISP: Idea Generation, Arrangement, Evaluation, Mixing & Mastering, Performance.

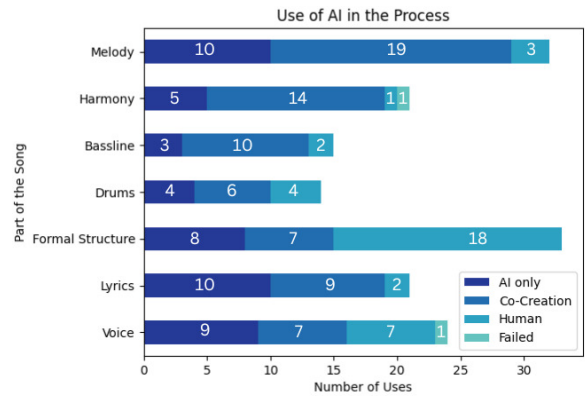


Figure 2. Overview of the Use of AI for the Elements of the Song in HAISP: Melody, Harmony, Bassline, Drums, Formal Structure, Lyrics and Voice.

citing a +1 for agreement with the coding choices or -1 for disagreement with the coding within the dataset, adding what they felt the coder was missing within their codes from the data. In the case of disagreement a third researcher helped decide on the final code as a tie-breaker.

4.4 Ethical Considerations

The study was approved by the university Institutional Review Board. When submitting to the contest, one question in the submission form asked directly and transparently for the team’s consent for their song and documentation to be used for research purposes. Teams gave consent by ticking a box in the form. While we erased all personal data from the dataset, song names are included due to their public availability on the AI Song Contest website.

5. USE CASE AND APPLICATIONS

Our dataset can be used to understand a variety of MIR tasks and work related to human-AI creative interactions, ranging from goals focused on research to practical creation of music. This section presents four possible use cases.

5.1 Use Case 1: More Insights on the AI Tools used in the Songwriting Process

HAISP lists not only all AI tools that teams used and indicated, but also contains the teams’ description of the process of utilizing said tools, with data showing which parts of the creative process (idea generation, composition, mixing/mastering, performance) AI was used on its own, in co-creation with the team, or even when co-creation with the AI system failed. Additionally, it reports the teams’ level of expertise and location. This provides not only valuable insights for tool developers and inspiration or guidance on using AI for other artists but also helps to answer research questions such as “How does the utilization of AI tools in music creation vary across different professional affiliations and stages of the creative process?”

5.2 Use Case 2: Understanding Attitudes and Impact of AI on the Creative Process

Within our dataset, there is reference to how creators use AI within their process, as well as their perceptions around the capabilities and limitations of AI. Additionally, creators share their perceptions of the final creation resulting from their collaboration with AI, providing valuable insights into the impact of AI on artistic expression and creative outcomes.

In their work *Beyond Diverse Datasets* [36], Huang et al. pose questions such as “What is valuable to those communities [that MIR investigates] and what is valuable to the community contributing to MIR?” and “How do musical communities wish for their practices to interact with emerging technologies (if at all), and what do they consider as potential misuses of their traditions?”. Our dataset offers a unique opportunity to start exploring these questions, drawing from the diverse perspectives of creators from various cultural, educational, and experiential backgrounds. By analyzing the reflections and experiences of creators documented in the dataset, researchers can gain insights into how different musical communities perceive and interact with AI technologies in their creative practices.

5.3 Use Case 3: Gaining Insight into Users’ Understanding of Ethics Around H-AI Co-Creation

Ethical considerations and questions about the validity of data were found frequently in the responses of AISC participants. Working with AI tools during the creation process can trigger questions around control [42], ownership of the final output [43], and freedom of personal expression [44]. Additionally, many participants spoke on the issue of AI systems and tools used during their process potentially being trained on datasets that violate the intellectual property rights of the original artists.

By examining how creators navigate ethical considerations in their process, researchers can uncover how AI tools are adopted and used within different musical traditions. This deeper understanding can inform discussions around the ethical implications of AI in music creation and

contribute to the development of responsible approaches to AI-driven creativity. One research question that this data can provide insight on is “What are the ethical considerations and challenges faced by creators when utilizing AI tools in their creative process, and how do these considerations impact their creative workflows?”

5.4 Use Case 4: Execution of Creative Possibilities

Our dataset can be used to understand the ways in which AI tools can expand the range of creative possibilities in songwriting and music creation. By leveraging this dataset, researchers can analyze how AI models have allowed users to generate innovative methods to address creative challenges, potentially expanding beyond the field of music creation into other creative fields.

One research question that can be explored with this dataset is “How do AI tools, particularly those leveraging big data and machine learning techniques, expand the creative possibilities in songwriting and music creation, and what novel approaches to executing creative problems do they enable?”. Specifically, one can explore the utilization of big data and machine learning techniques to address challenges such as data processing, limited musical ability, and idea generation. For example, AI-powered algorithms can analyze vast amounts of musical data to identify patterns and trends, providing inspiration for melody creation, chord progressions, and rhythmic structures. Additionally, machine learning techniques can assist in data processing tasks focused on mechanisms of creation, enabling creators to focus more on the higher level of creation process in music composition [45].

Extending the scope of research beyond music creation into other creative fields, researchers can use HAISP to examine how AI algorithms are applied to address creative challenges and gain insight into the broader implications of AI for fostering creativity and innovation.

6. COMPARISON WITH OTHER DATASETS

There are extensive datasets of AI-based music tools that focus on the methods of the respective AI systems [7] or reviews of AI-based music tools that focus on the metadata of the publication [1]. Another way to approach the topic of AI-supported creative processes is analyzing interfaces of AI-supported tools for creative endeavors [37]. In the MIR community, datasets are common, especially quantitative datasets and training datasets. Apart from [2], who analyzed the creation process of the first AISC teams, there have been no datasets consisting of qualitative data of the human-AI co-creation process released in MIR venues. The HAISP dataset presented in this paper, is a qualitative dataset of a substantial amount of curated user data, including a subjective description and evaluation of the process, practices, and ethical issues around the creative process with AI. Due to the international character of the AISC – with teams from over 20 countries in the 2023 edition alone – these descriptions come from individuals with diverse cultural backgrounds and musical traditions, which

is shown in their reflections and experiences.

The limited amount of data makes HAISP unsuitable for making general and widely applicable statements about human-AI interaction in songwriting. Rather, we see HAISP as a dataset that can be used to extend existing research in various academic disciplines, as it gives very detailed and rich insights that are well-suited to complement quantitative research insights. Therefore, we made HAISP publicly available and encourage researchers from different fields to work with the data, bringing in their respective perspectives and methods in order to foster an interdisciplinary dialogue on human-AI co-creation.

7. CONCLUSION

In conclusion, the Human-AI Songwriting Processes dataset stands as a potentially significant resource for the exploration of human-AI collaboration and creativity in music composition. Curated from submissions to the AI Song Contest, this dataset offers a view of the dynamics underlying AI-supported songwriting processes. It provides valuable insights into how creators from diverse backgrounds integrate AI tools into their creative workflows, reflecting on the capabilities, limitations, and ethical considerations inherent in human-AI collaboration. Furthermore, the adoption of the HAISP dataset has the potential to advance interdisciplinary inquiry, inspire further research, and contribute to ongoing discourse surrounding human-AI collaboration and creativity. By fostering critical inquiry and facilitating informed discourse, this dataset contributes to our understanding of technology’s role in creativity and innovation in the digital age. Future work on the Human-AI Songwriting Processes dataset will involve expanding the dataset to include information from participants in the AI Song Contest prior to 2023, enriching the dataset with a broader range of submissions and perspectives. Additionally, there are plans to conduct further analysis on the dataset, including analysis to explore the descriptive terminology used for AI tools and systems. In closing, the HAISP dataset holds promise for advancing our understanding of human-AI collaboration in music composition. Through its insights and reflections, it encourages continued exploration of the dynamic relationship between human creativity and machine intelligence.

8. ACKNOWLEDGMENTS

We thank all organizers and former co-organizers of the AI Song Contest, Anna Huang, Rujing Stacey Huang and Vincent Koops for their contribution in organizing this event. We also thank Aarushi Buddhavarapu for their contribution to the coding on this project.

9. REFERENCES

- [1] M. Civit, J. Civit-Masot, F. Cuadrado, and M. J. Escalona, “A Systematic Review of Artificial Intelligence-based Music Generation: Scope, Ap-

- plications, and Future Trends,” *Expert Systems with Applications*, vol. 209, pp. 118–190, 2022.
- [2] C.-Z. A. Huang, H. V. Koops, E. Newton-Rex, M. Dinulescu, and C. J. Cai, “AI Song Contest: Human-AI Co-Creation in Songwriting,” in *21st International Society for Music Information Retrieval Conference*, 2020.
- [3] I. Xenakis, *Formalized Music: Thought and Mathematics in Composition: Thoughts and Mathematics in Composition*. Pendragon Press, 1992.
- [4] E. Frid, C. Gomes, and Z. Jin, “Music Creation by Example,” in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 2020, pp. 1–13.
- [5] R. T. Dean and A. McLean, *The Oxford Handbook of Algorithmic Music*. Oxford University Press, 2018.
- [6] J. J. Bharucha and P. M. Todd, “Modeling the Perception of Tonal Structure with Neural Nets,” vol. 13, no. 4, pp. 44–53, 1989, publisher: The MIT Press.
- [7] J.-P. Briot, G. Hadjeres, and F.-D. Pachet, *Deep Learning Techniques for Music Generation*, ser. Computational Synthesis and Creative Systems. Cham: Springer International Publishing, 2020.
- [8] C.-Z. A. Huang, C. Hawthorne, A. Roberts, M. Dinulescu, J. Wexler, L. Hong, and J. Howcroft, “The Bach Doodle: Approachable Music Composition with Machine Learning at Scale,” in *20th International Society for Music Information Retrieval Conference*, 2019.
- [9] T. Bazin and G. Hadjeres, “NONOTO: A Model-agnostic Web Interface for Interactive Music Composition by Inpainting,” in *International Conference on Computational Creativity 2019*, 2019, pp. 89–91.
- [10] O. Lopez-Rincon, O. Starostenko, and G. A.-S. Martín, “Algorithmic music composition based on artificial intelligence: A survey,” in *2018 International Conference on Electronics, Communications and Computers (CONIELECOMP)*, 2018, pp. 187–193.
- [11] G. Hadjeres, F. Pachet, and F. Nielsen, “DeepBach: a Steerable Model for Bach Chorales Generation,” in *International Conference on Machine Learning*. PMLR, 2017, pp. 1362–1371.
- [12] A. Shaw, “A Multitask Music Model with Bert, transformer-xl and seq2seq,” 2019. [Online]. Available: <https://towardsdatascience.com/a-multitask-music-model-with-bert-transformer-xl-and-seq2seq-3d80bd2ea08e>
- [13] D. Ippolito, A. Huang, C. Hawthorne, and D. Eck, “Infilling Piano Performances,” in *NIPS Workshop on Machine Learning for Creativity and Design*, vol. 2, no. 5, 2018.
- [14] C.-J. Chang, C.-Y. Lee, and Y.-H. Yang, “Variable-Length Music Score Infilling via XLNet and Musically Specialized Positional Encoding,” in *Proceedings of the 22nd International Society for Music Information Retrieval Conference*, 2021.
- [15] C. Donahue, I. Simon, and S. Dieleman, “Piano Genie,” in *Proceedings of the 24th International Conference on Intelligent User Interfaces*, 2019, pp. 160–164.
- [16] K. Tahiroglu, M. Kastemaa, O. Koli *et al.*, “AI-terity: Non-Rigid Musical Instrument with Artificial Intelligence Applied to Real-Time Audio Synthesis,” in *International Conference on New Interfaces for Musical Expression*, 2020, pp. 337–342.
- [17] C. Benetatos and Z. Duan, “BachDuet: A Deep Learning System for Human-Machine Counterpoint Improvisation,” *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2019.
- [18] S. Fukayama, K. Yoshii, and M. Goto, “Chord-Sequence-Factory: A Chord Arrangement System Modifying Factorized Chord Sequence Probabilities,” in *14th International Society for Music Information Retrieval Conference*, 2013, pp. 457–462.
- [19] C.-Z. A. Huang, D. Duvenaud, and K. Z. Gajos, “ChordRipple: Recommending Chords to Help Novice Composers Go Beyond the Ordinary,” in *Proceedings of the 21st International Conference on Intelligent User Interfaces*, 2016, pp. 241–250.
- [20] M. Navarro-Caceres, M. Caetano, G. Bernardes, and L. N. de Castro, “Chordais: An Assistive System for the Generation of Chord Progressions with an Artificial Immune System,” *Swarm and Evolutionary Computation*, vol. 50, p. 100543, 2019.
- [21] C. M. Wilk and S. Sagayama, “Automatic Music Completion Based on Joint Optimization of Harmony Progression and Voicing,” *Journal of Information Processing*, vol. 27, pp. 693–700, 2019.
- [22] H. V. Koops, J. P. Magalhaes, and W. B. De Haas, “A functional approach to automatic melody harmonisation,” in *Proceedings of the first ACM Workshop on Functional Art, Music, Modeling, and Design*, 2013, pp. 47–58.
- [23] T.-P. Chen and L. Su, “Attend to Chords: Improving Harmonic Analysis of Symbolic Music Using Transformer-Based Models,” *Transactions of the International Society for Music Information Retrieval*, vol. 4, no. 1, pp. 1–13, 2021.
- [24] I. Simon, D. Morris, and S. Basu, “MySong: Automatic Accompaniment for Vocal Melodies,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2008, pp. 725–734.

- [25] C.-H. Chuan and E. Chew, “A hybrid system for automatic generation of style-specific accompaniment,” in *Proceedings of the 4th International Joint Workshop on Computational Creativity*, 2007, pp. 57–64.
- [26] S. Oramas, F. Barbieri, O. Nieto Caballero, and X. Serra, “Multimodal Deep Learning for Music Genre Classification,” *Transactions of the International Society for Music Information Retrieval*, pp. 4–21, 2018.
- [27] D. Schneider, N. Korfhage, M. Mühling, P. Lüttig, and B. Freisleben, “Automatic Transcription of Organ Tablature Music Notation with Deep Neural Networks,” *Transactions of the International Society for Music Information Retrieval*, vol. 4, no. 1, pp. 14–28, 2021.
- [28] H. Cuesta and E. Gómez Gutiérrez, “Voice Assignment in Vocal Quartets Using Deep Learning Models Based on Pitch Salience,” *Transactions of the International Society for Music Information Retrieval*, 2022.
- [29] P. Torras, A. Baró, L. Kang, and A. Fornés, “On the Integration of Language Models into Sequence to Sequence Architectures for Handwritten Music Recognition,” in *Proceedings of the 22nd International Society for Music Information Retrieval Conference*, 2021, pp. 690–696.
- [30] B. Meléndez-Catalán, E. Molina, and E. Gómez, “Open Broadcast Media Audio from TV: A Dataset of TV Broadcast Audio with Relative Music Loudness Annotations,” *Transactions of the International Society for Music Information Retrieval*, vol. 2, no. 1, pp. 43–51, 2019.
- [31] S. Schwär, M. Krause, M. Fast, S. Rosenzweig, F. Scherbaum, and M. Müller, “A Dataset of Larynx Microphone Recordings for Singing Voice Reconstruction,” *Transactions of the International Society for Music Information Retrieval*, 2024.
- [32] S. Rosenzweig, F. Scherbaum, D. Shugliashvili, V. Arifi-Müller, and M. Müller, “Erkomaishvili Dataset: A Curated Corpus of Traditional Georgian Vocal Music for Computational Musicology,” *Transactions of the International Society for Music Information Retrieval*, vol. 3, no. 1, pp. 31–41, 2020.
- [33] Q. Kong, B. Li, J. Chen, and Y. Wang, “GiantMIDI-piano: A large-scale MIDI dataset for classical piano music,” *Transactions of the International Society for Music Information Retrieval*, vol. 5, no. 1, pp. 87–98, 2022.
- [34] F. Zalkow, S. Balke, V. Arifi-Müller, and M. Müller, “MTD: A Multimodal Dataset of Musical Themes for MIR Research,” *Transactions of the International Society for Music Information Retrieval*, vol. 3, no. 1, pp. 180–192, 2020.
- [35] G. Meseguer-Brocal, A. Cohen-Hadria, and G. Peeters, “Creating DALI, a Large Dataset of Synchronized Audio, Lyrics, and Notes,” *Transactions of the International Society for Music Information Retrieval*, vol. 3, no. 1, pp. 55–67, 2020.
- [36] R. S. Huang, A. Holzapfel, B. L. T. Sturm, and A.-K. Kaila, “Beyond Diverse Datasets: Responsible MIR, Interdisciplinarity, and the Fractured Worlds of Music,” *Transactions of the International Society for Music Information Retrieval*, vol. 6, no. 1, pp. 43–59, 2023.
- [37] J. Rezwana and M. L. Maher, “User Perspectives on Ethical Challenges in Human-AI Co-Creativity: A Design Fiction Study,” in *C&C '23: Creativity and Cognition*, 2023, pp. 62–74.
- [38] K. Lee, G. Hitt, E. Terada, and J. H. Lee, “Ethics of Singing Voice Synthesis: Perceptions of Users and Developers,” in *Proceedings of the 23rd International Society for Music Information Retrieval Conference*, 2022.
- [39] J. A. Biles, “GenJam: A Genetic Algorithm for Generating Jazz Solos,” in *Proceedings of the 1994 International Computer Music Conference*, 1994, p. 8.
- [40] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. V. Le, and R. Salakhutdinov, “Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context,” in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.
- [41] C. E. Hill, S. Knox, B. J. Thompson, E. N. Williams, S. A. Hess, and N. Ladany, “Consensual qualitative research: An update,” *Journal of Counseling Psychology*, vol. 52, no. 2, pp. 196–205, 2005.
- [42] N. Davis, C.-P. Hsiao, K. Y. Singh, and B. Magerko, “Co-Creative Drawing Agent with Object Recognition,” in *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, vol. 12, no. 1, 2016, pp. 9–15.
- [43] J. Rezwana and M. L. Maher, “User Perspectives of the Ethical Dilemmas of Ownership, Accountability, Leadership in Human-AI Co-Creation,” in *Joint Proceedings of the ACM IUI Workshops*, 2023.
- [44] D. Buschek, L. Mecke, F. Lehmann, and H. Dang, “Nine Potential Pitfalls when Designing Human-AI Co-Creative Systems,” in *Joint Proceedings of the ACM IUI 2021 Workshops*, 2021.
- [45] M. Newman, L. Morris, and J. H. Lee, “Human-AI Music Creation: Understanding the Perceptions and Experiences of Music Creators for Ethical and Productive Collaboration,” in *Proceedings of the 24th International Society for Music Information Retrieval Conference*, 2023, pp. 80–88.

CUE POINT ESTIMATION USING OBJECT DETECTION

Giulia Argüello

Luca A. Lanzendörfer
ETH Zurich

Roger Wattenhofer

{agiulia, lanzendoerfer, wattenhofer}@ethz.ch

ABSTRACT

Cue points indicate possible temporal boundaries in a transition between two pieces of music in DJ mixing and constitute a crucial element in autonomous DJ systems as well as for live mixing. In this work, we present a novel method for automatic cue point estimation, interpreted as a computer vision object detection task. Our proposed system is based on a pre-trained object detection transformer which we fine-tune on our novel cue point dataset. Our provided dataset contains 21k manually annotated cue points from human experts as well as metronome information for nearly 5k individual tracks, making this dataset 35x larger than the previously available cue point dataset. Unlike previous methods, our approach does not require low-level musical information analysis, while demonstrating increased precision in retrieving cue point positions. Moreover, our proposed method demonstrates high adherence to phrasing, a type of high-level music structure commonly emphasized in electronic dance music. The code, model checkpoints, and dataset are made publicly available.¹

1. INTRODUCTION

The skills required by a “Disc Jockey” (DJ) are diverse. To record and play live DJ mixes, DJs need to prepare and know their tracks well. An integral part of the track preparation phase is the placement of cue points. Coined by scratch DJs who placed stickers on vinyl records to indicate important sections, the functionality of cue points remains unchanged in the digital setting. A cue point may serve as an annotation for musical highlights, suitable mixing boundaries, or the general track structure which consists of musical phrases. Furthermore, digital cue points allow DJs to quickly loop a track segment or skip back and forward during a live performance, altering the track structure on the spot. Unfortunately, placing cue points and track preparation is often a cumbersome and time-consuming process. Similarly to other music information retrieval (MIR) tasks, such as onset detection or beat tracking, cue point placement is not straightforward, despite

¹ <https://github.com/ETH-DISCO/cue-detr>

the prominent structural regularity in electronic dance music (EDM) [1]. For instance, the presence of a prelude shifts the track structure, creating irregularity, and similarly, tracks with arbitrary number of additional bars or tempo variations create a significant challenge which needs to be addressed. We therefore ask the question whether cue point estimation can be automated with a learned approach, imitating human cue point placements by training a model on a manually annotated dataset.

This work addresses the placement of cue points, one of the first tasks during the preparation phase of a DJ mix. With this goal in mind we present CUE-DETR, a fine-tuned DETR image object detection model trained for cue point estimation on EDM tracks. We show CUE-DETR outperforms previous approaches without requiring detailed and meticulously curated rule sets, which leverage underlying low-level audio information.

Our contributions can be summarized as follows:

- We propose CUE-DETR, an object detection model capable of predicting cue points in EDM tracks. Compared to previous methods, our model achieves higher precision and shows significantly closer alignment with manually placed cue points.
- We make our EDM-CUE dataset publicly available, which is 35x larger than the previously available cue point dataset [2]. EDM-CUE contains the metadata for 4,710 EDM tracks, which includes tempo, beat, downbeat, and 21k manually placed cue point annotations provided by human experts.
- To increase evaluation objectivity, we introduce additional phrase aligned points to evaluate prediction accuracy. Moreover, we open-source the code and model checkpoints to further the research of DJ-related MIR tasks.

2. RELATED WORK

Recent years have seen emerging interests in building automated DJ systems where most approaches try to recreate a fully automated DJ pipeline [3–9]. Such systems aim to create seamless transitions between two tracks, each focusing on a different subset of challenges in the DJ’s task pipeline. Cue points are predominantly addressed in the context of finding suitable mix positions in automatic mixing systems [3, 6, 7, 10]. Music structure analysis forms the basis for most cue detection algorithms, as DJ mixes tend to adhere to the underlying high-level track structures [11].



High-novelty regions found through self-similarity [12], for instance, allow the determination of suitable mix sections based on the high-level music structure [6, 13, 14]. Furthering the structural knowledge of a track, crowd-sourced scrubbing data from streaming services uncovers additional structural context, as listeners tend to skip forward to the most prominent section of a track [10]. Applying learning-based concepts for the direct search of musical highlights [7] reveals useful information about the musical structure in a similar manner.

Generally, the accuracy of algorithmically chosen cue points varies depending on the granularity and completeness of the rule set implemented in conjunction with the structural analysis [13]. Adding further rules into the set, for instance, introduces a trade-off between the number of correctly estimated cue point positions and the correctness of each estimated cue point [14]. The main focal point of the open-source DJ system Automix [14] is a rule-based cue point estimation algorithm, including a validation dataset containing 145 tracks [2]. Automix implements four empirically chosen rules describing possible locations of “switch points,” a subset of cue points, on top of structural analysis. Furthermore, the implementation of Automix depends on underlying MIR tasks, such as beat tracking.

DJ mix reverse-engineering [15, 16] is a related task to cue point estimation, as it addresses the lack of available and ready-to-use datasets [17]. Such “unmixing” methods extract latent mixing information from recorded DJ mixes, whose retrieval typically relies on manual annotations, such as mix-in and mix-out points or volume gain curves. The use of pure DJ mix reverse-engineering for cue point estimation is limited as no novel cue points can be retrieved from existing DJ mixes.

In the context of lower-level MIR tasks, convolutional neural networks (CNNs) have been studied, for example, in onset detection [18] or beat tracking [19]. Furthermore, CNNs have proven helpful in musical structural analysis and boundary estimation [20]. Using an attention mechanism in conjunction with a CNN can help alleviating the challenges posed by the sequential nature of music. Nevertheless, adding an attention mechanism does not solve the main concern posed by the large amounts of data required for training. Another possible solution is to instead use a large pre-trained model and to then fine-tune the model on task-specific datasets. The Audio Spectrogram Transformer [21], for instance, demonstrates the possibility to transfer a pre-trained ViT model [22] from the image domain to the audio domain. Transformer architectures are often designed to apply the attention mechanism together with a pre-trained CNN backbone, leveraging the feature space previously learned by the CNN [23, 24].

3. METHODOLOGY

3.1 Dataset

We created EDM-CUE, a dataset containing music meta-data from four private collections of professional DJs.

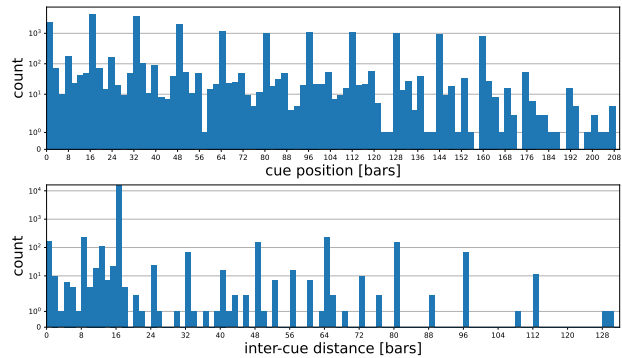


Figure 1. Top: Distribution of cue point positions in EDM-CUE. Bottom: Distribution of distances between two subsequent cue points in EDM-CUE. The inter-cue distances indicate that 16 bars is the most represented phrasing length in our dataset.

Each of the four DJs uses the library management tool rekordbox² from which we collect the track name, artist name, tempo, beat grid, and cue points for each contained track. Cue points are given by their absolute position in seconds. The beat grid represents a visual metronome, which can be calculated from its stored values: the tempo and grid offset return the beat positions. Applying the time signature in combination with the initial beat number reveals the downbeat. Since we aggregate tracks from four individual collections, all duplicate tracks need to be merged. We summarize the tempo and grid offset to their respective mean values for all duplicate track entries. In order to merge duplicate cue points, we group all cue points based on their distance to neighboring points. Cue points within a distance of a quarter beat of one another form a group. The merged cue point value corresponds to the group center position. All dataset tracks are based on a 4/4 time signature and show constant tempo over time, outlier tracks were excluded during collection. We then pair the information of each track with the track ID found on Deezer³ to provide an additional reference.

Our dataset contains 4,710 EDM tracks consisting of around 380 hours of music. The tempo-range lies between 95 and 190 bpm, and track duration ranges from 1 minute 37 seconds to 10 minutes with an average of 4 minutes and 50 seconds. In total, the dataset contains 21,461 cue point annotations with an average count of 4.6 cue points per track. All tracks used to train the model are compressed to 128 kbps MP3 at 44.1 kHz.

3.2 Phrasing

Although cue points frequently align with high-level structural boundaries and tend to strongly coincide with phrase boundaries [11], the placement of cue points is a subjective task with no clear definition; therefore, annotations collected from DJs may not contain all plausible cue points. We first examine the distribution of our training data for

²<https://rekordbox.com>

³<https://www.deezer.com>

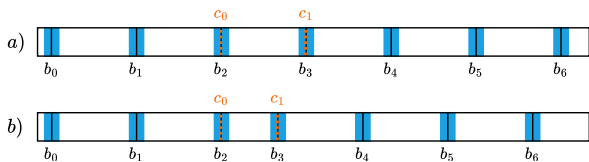


Figure 2. Calculation of phrase boundaries b_i using cue points c_i . Phrase boundaries, highlighted in blue, serve as additional points to evaluate prediction accuracy. Example a) represents a track with regular phrasing whereas b) shows a track with an irregular phrase between cue points c_0 and c_1 . The computed phrase boundaries b_i include the cue points.

cue point positions quantized to bars. Our training cue points exhibit a periodicity with high occurrences of cue points on multiples of 8 and 16 bars, as illustrated in Figure 1. When also taking the inter-cue spacing between neighboring cue points into account, we observe that a majority of our training tracks adhere to phrase lengths of 16 bars, followed by 8 bars. Due to the strong regularity, we will refer to sections with phrase lengths other than 8 or 16 bars as “irregular.” Furthermore, analyzing the cue points in EDM-CUE we find DJs often place cue points at the start of such irregular sections.

Since regular and clearly defined phrasing is common in EDM [1], we generalize our collected ground-truth data by estimating phrase boundaries B . Phrase boundaries serve as an approximation of the track structure which we use to further validate model accuracy. Using track duration t , phrase length l , and an ordered, ground-truth cue point set C , we find B . The non-empty set C must include cue points c_i which mark the start point of irregular phrase boundaries. Traversing the section preceding the first cue point $c_0 = b_0$ in increments of l yields the first entries of B . When the iteration reaches a negative value, the remaining track section from c_0 is traversed in the opposite direction until $b_i \geq t$. A new boundary b_i is added to B if the iteration step did not skip or reach any c_i . Otherwise, the next cue c_i is added to B as b_i . The two simplified examples in Figure 2 show resulting boundaries.

3.3 Model

Our proposed cue estimation system is based on DETR [23], a pre-trained object detection transformer. For each track in the dataset, we generate Mel spectrograms using 128 Mel bands at a sampling rate of 22,050 Hz. Our window length measures 2,048 samples, and the hop length is 512 samples.

The input of the model consists of 128×355 pixel spectrogram segments to fit the expected input image format for DETR while also maximizing the duration of the depicted audio to approximately 11 seconds per image. In the following, we refer to a complete track spectrogram as S . The training spectrogram segments S_T and inference spectrogram segments S_I denote the input images of the model. The model returns positional encodings for the pre-

dicted bounding boxes alongside the accompanying confidence scores and class labels represented by logits. The data pipeline is illustrated in Figure 3.

3.4 Preprocessing

We differentiate between preprocessing for training and inference, as the model is required to process complete spectrograms during inference, whereas for training, the model only requires image segments depicting cue points. A training image segment S_T is cut from S around a cue point p found in S . Using a random integer offset $o \in [0, 355)$, image S_T is defined as the segment with left side $p - o$ and right side $p - o + 355$. If image S_T partly lies outside of spectrogram S , the additional space in S_T is zero-padded. The inclusion of image offset o acts as a simple data augmentation strategy. For the training annotations, each cue point in an image S_T is encapsulated by a bounding box. The aforementioned box occupies the entire height of S_T and is centered around the cue point. In the event that the box extends beyond the image, it is cropped to align with the image borders. Due to this cropping strategy, all training tracks are split into training and validation sets and are indexed by their respective cue annotations.

To make predictions over the span of a full track, during inference, the complete spectrogram needs to be shown to the model. We employ a sliding window cropping strategy on spectrogram S with an overlap of 0.75 in order to generate inference image segments S_I . Similarly to training, the left side of spectrogram S is zero-padded with an arbitrary offset $o \in [89, 266]$ prior to cropping. Applying the zero-padding approaches the uniform distribution of cue point positions seen in the training data, thus increasing the chance to detect cue points at the very start of a spectrogram. As the final step, the resulting image sequence is normalized.

3.5 Postprocessing

We implement additional postprocessing for inference only since additional processing of the basic DETR output is not necessary during training. The model outputs contain the logits and positional encodings mapping to the predicted bounding box coordinates over images S_I . Applying a softmax function to the logits yields the class labels and confidence scores for each prediction, cue points are retrieved from the respective positional encodings. The positional box representation is converted to pixel coordinates in corner format to find the center point on the x -axis. The resulting point is mapped back to the absolute coordinates of track spectrogram S using the left edge of image segment S_I . Once all conversion results for spectrogram S have been accumulated, the confidence scores are sorted by their associated position, resulting in peaks where the confidence is highest. We implement a peak selection strategy using radius r ; final cue point candidates are selected in descending order based on their predicted confidence score. Candidates within radius r of a previously selected candidate are ignored. We use a confidence

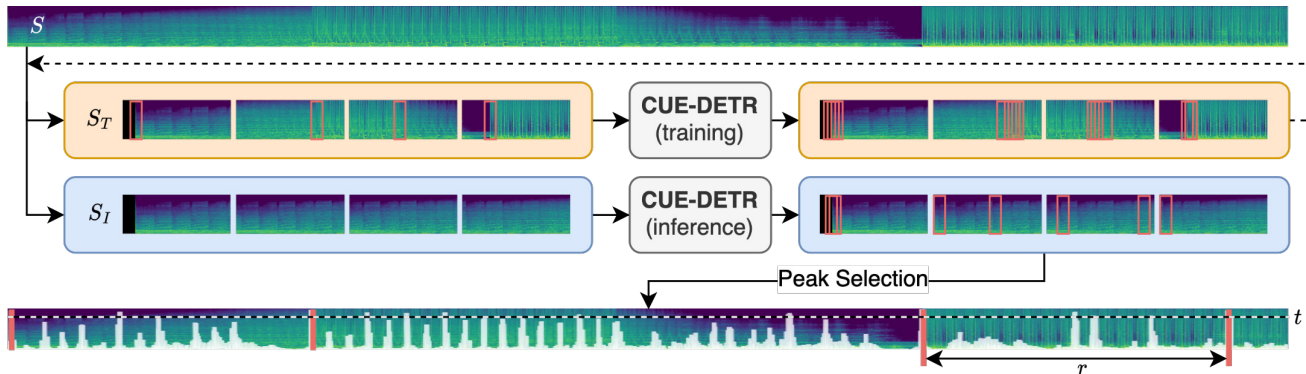


Figure 3. Pipeline of the proposed CUE-DETR architecture. During training, an input Mel spectrogram S is segmented into training images S_T . Each S_T consists of a spectrogram segment containing a cue point which is represented as a bounding box. Inference images S_I move across S using a sliding window. The predicted bounding boxes are converted to their center x -coordinate. The highest scoring positions are selected greedily among all candidates with minimum confidence $t = 0.9$. A selected position excludes all other candidates within a radius r . The bottom spectrogram shows the predicted positions as peaks based on the confidence value.

score threshold of 0.9 as the lower bound for selected candidates. Figure 4 shows three example spectrograms with sorted confidence scores. The highest peaks in the curve representation of confidence scores coincide with ground-truth cue points or phrase boundaries, however with noticeable additional high scoring positions. The additional high peaks are predominantly present at 4 bar intervals. As discussed in Section 3.2, we found ground-truth cue points align best with 16-bar phrases. We found that enforcing a minimum spacing r of 16 or 8 bars between consecutively predicted cue points improves the outcome of the final predictions with respect to precision.

4. EVALUATION

The final evaluation is conducted on 101 tracks which were excluded from the training and validation split. This test set contains 607 ground-truth cue point annotations.

4.1 Experiment Setup

We initialize CUE-DETR with pre-trained weights from DETR.⁴ The backbone is initialized with the ResNet-50 weights, and we set the backbone learning rate to 10^{-6} . For the transformer, we choose a learning rate of 10^{-5} , and set the weight decay to 10^{-4} . The bounding box width w is set to 21 pixels and the postprocessing radius r is fixed at 16 and 8 bars, referenced as r_{16} and r_8 , respectively. We train the model using AdamW [25] and schedule a learning rate reduction by factor 10 when the validation loss does not improve for 10 epochs. The final model is trained for 50 epochs on one NVIDIA TITAN Xp GPU with a batch size of 192.

While we experimented with training CUE-DETR using randomly initialized transformer weights, we found using pre-trained weights provided significantly better results. Even though the pre-trained transformer weights were trained on COCO 2017 [23, 26], a distinctly different

data distribution compared to Mel spectrograms, we corroborate previous findings of visual feature space transfer learning [21, 27].

We compare our model with two other methods, namely “Mixed In Key 10” (MIK), a commercial DJ software,⁵ and Automix [14], an open-source research project. We analyze all tracks directly without manual interference in MIK, as the program simultaneously estimates the beat grid to which it snaps generated cue points. From Automix, we used the cue point generation method directly.

4.2 Evaluation Metrics

We investigate the predicted cue points with respect to the manually annotated cue points and phrase alignment separately. In the following, we address the manually annotated cue point ground-truth set by *cues-only* and use the phrase length, measured in bars, to reference phrase alignment. Similarly to Automix, we assess the predictions using a tolerance window around the ground-truth cue points to estimate the hit rate of the predictions. We evaluate the models on two different tolerance windows T_1 and $T_{1/2}$ which measure one beat and one half-beat, respectively. On average, one half-beat in our test data measures approximately 172 milliseconds, which is comparable to the standard 150 milliseconds tolerance in beat tracking [28]. The values for precision, recall, the F_1 -score and Average Precision (AP) scores are retrieved from the hit rate. Lastly, we measure the cosine similarity between the sets of the predicted and actual cue point positions.

4.3 Ablations

As cue points have no clearly defined object boundaries, we further investigate the influence of the spectrogram context around a cue point included in a bounding box. We report the impact of the bounding box width w for the quality of predictions in Table 1 using AP. We report AP for cues-only as AP_C and report AP for phrase alignment

⁴ <https://huggingface.co/facebook/detr-resnet-50>

⁵ <https://mixedinkey.com>

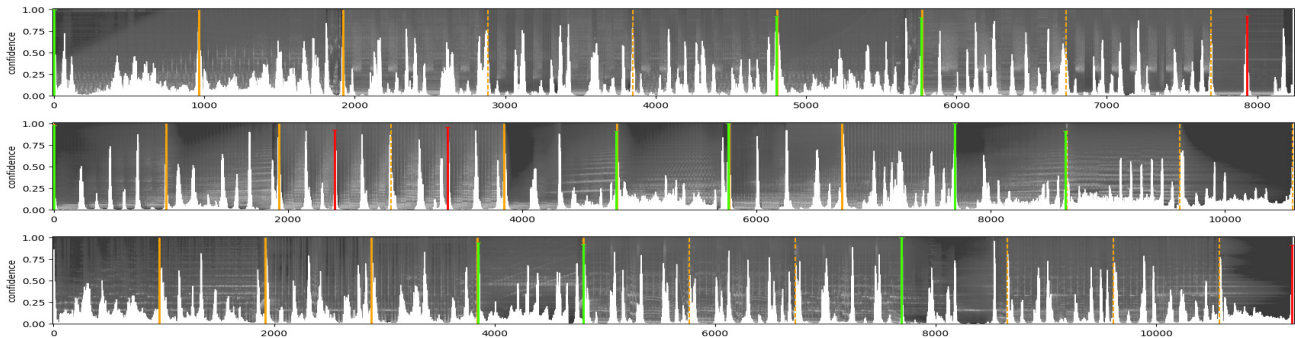


Figure 4. Predicted and ground-truth cue point positions shown over three Mel spectrograms of different random tracks from the evaluation split of EDM-CUE. The confidence score for each position is illustrated as the white curve. Magenta lines indicate correct model predictions, red lines indicate wrong model predictions. For reference, solid orange lines represent ground-truth positions and dashed orange lines illustrate 16-bar phrase boundaries.

Table 1. Ablation of the bounding box width w used during training of CUE-DETR. The Average Precision (AP) scores are reported as AP_C for cues-only ground-truth, AP_{16} and AP_8 indicate phrase alignment. The best results per scenario are bold and larger values are better.

	w	T_1 (one beat)			$T_{1/2}$ (half beat)		
		AP_C	AP_{16}	AP_8	AP_C	AP_{16}	AP_8
r_{16}	7	0.41	0.51	0.52	0.34	0.37	0.37
	15	0.41	0.57	0.60	0.36	0.42	0.42
	21	0.41	0.57	0.60	0.38	0.47	0.48
r_8	7	0.32	0.42	0.45	0.23	0.26	0.27
	15	0.32	0.49	0.53	0.25	0.33	0.34
	21	0.32	0.50	0.54	0.28	0.38	0.41

as AP_{16} and AP_8 . We trained three models with identical initialization parameters except for w which we set to $w_7 = 7$, $w_{15} = 15$, and $w_{21} = 21$ pixels, respectively.

Looking at the results for T_1 , the box width shows no impact on AP_C . The larger peak radius r_{16} increases AP_C for all models. Furthermore, AP increases from AP_C to AP_{16} for all models, most notably by 0.18 from 0.32 to 0.5 for w_{21} with r_8 . From AP_{16} to AP_8 we report an additional increase in AP. Using a larger w improves AP for the phrase alignment cases. The overall best AP score measures 0.6 for w_{15} and w_{21} with radius r_{16} . This radius produces identical results for w_{15} and w_{21} . The results for $T_{1/2}$ exhibit similar patterns, with the exception of w_{21} reporting improved AP over w_{15} on all accounts. Overall, the model with w_7 performs the least favorable, followed by w_{15} , which in turn is outperformed by w_{21} .

4.4 Results

The evaluation of the mean precision, recall, and the F_1 -score is summarized in Table 2. For all methods, the precision increases from the cues-only to the 16-bars and 8-bars ground-truth sets. Our r_{16} -model achieves the highest precision in all cases. The precision increases most notably for tolerance T_1 from the cues-only to phrase alignment

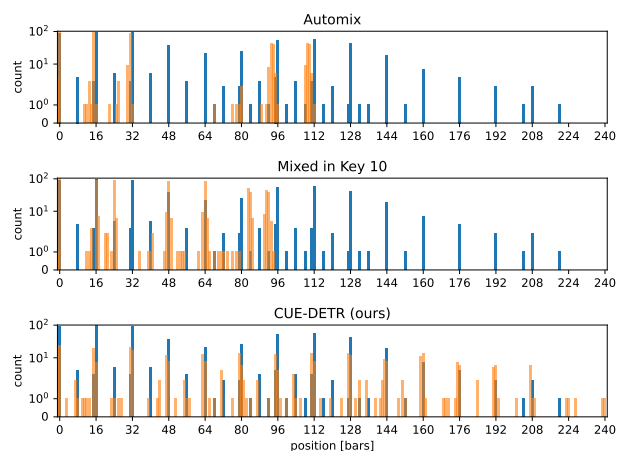


Figure 5. Distribution of ground-truth cue point positions in blue and predicted cue point positions in orange quantized to bars. The cosine similarity between the predicted cue point positions and ground-truth is 0.425 (Automix), 0.371 (MIK), and 0.851 (CUE-DETR).

ground-truth sets. More precisely, our r_8 -model shows an increase in precision by 0.31 from cues-only to 8-bar phrasing. The change from 16 to 8-bars is not as prevalent. Automix shows an improvement in precision from 0.14 to 0.24 and 0.3 over the three ground-truth sets. MIK shows little improvement over the different scenarios and produces more stable precision values. Using the tighter tolerance $T_{1/2}$, all precision values fall in proportion to each other. For recall, the difference of values between the two tolerances is similar to what is observed for precision. With the added phrasing boundaries, all methods show a reduction in recall, opposite to precision. The most significant drop in recall is observed from 16 to 8-bars. Our r_8 -model reports the highest recall on all accounts. The changes in the F_1 -score are less pronounced for all methods as the values remain nearly stable for cues-only and 16-bar phrase alignment. The best reported F_1 -score is associated with our r_8 -model over 16-bar phrasing at 0.46.

For further insight, we look at the distribution of the predicted results in Figure 5. Automix favors cue posi-

Table 2. Comparison of precision, recall, and the F_1 -score of Automix, Mixed In Key (MIK), and our method. Higher values correspond to better results. The upper rows show the evaluation using tolerance T_1 and the lower rows using $T_{1/2}$. From left to right, the results are given for the manually placed cue point only, the computed 16-bar phrasing and the computed 8-bar phrasing. We observe that CUE-DETR outperforms previous methods on precision, recall, and F_1 -score.

		<i>cues-only</i>			<i>16-bars</i>			<i>8-bars</i>		
		Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall	F_1
T_1	Automix	0.14	0.12	0.13	0.24	0.11	0.15	0.30	0.07	0.11
	MIK 10	0.20	0.25	0.22	0.21	0.13	0.16	0.25	0.08	0.12
	CUE-DETR (r_{16})	0.38	0.35	0.36	0.62	0.27	0.38	0.69	0.16	0.26
	CUE-DETR (r_8)	0.32	0.49	0.39	0.53	0.41	0.46	0.63	0.26	0.36
$T_{1/2}$	Automix	0.11	0.10	0.10	0.20	0.09	0.13	0.24	0.06	0.10
	MIK 10	0.14	0.19	0.16	0.15	0.09	0.12	0.18	0.06	0.09
	CUE-DETR (r_{16})	0.27	0.25	0.26	0.43	0.19	0.27	0.48	0.11	0.18
	CUE-DETR (r_8)	0.22	0.34	0.27	0.37	0.28	0.32	0.43	0.17	0.25

tions around the first three phrases with high alignment to ground-truth. The second cluster is predicted at the start of phrases 6 to 8 with an increased tendency for early predictions. MIK on the other hand exhibits more evenly distributed cue placements over the first 7 phrases. However, an increased number of predictions lie in between phrases where no ground-truth points lie. We observe that both Automix and MIK tend not to predict possible cue points in the second half of tracks. CUE-DETR predicts cue points with the highest adherence to ground-truth. Despite a few additional predictions similar to MIK, positions with the highest accumulation of cue points are covered by our predictions in a similar pattern. The cosine similarity of our quantized predictions reports the highest score of 0.851. In comparison, Automix scores 0.425 whereas MIK reaches 0.371.

4.5 Discussion

CUE-DETR shows strong adherence to ground-truth compared to other methods. Our method suggests good phrase alignment based on the distribution of our predicted cue point positions, as well as the increase in precision from cues-only to 16 bar phrases. A slight increase in precision is expected for all methods, however, a significant increase is only associated with strong phrase alignment due to the decrease in false positive predictions. The higher number of possible ground-truth positions decreases recall in return. If our method successfully detects irregular sections, the phrasing algorithm from Section 3.2 can be applied in postprocessing, which could further increase the precision while keeping the recall score high.

Despite using a metronome-agnostic approach, for which we fixed the distances r to the length of a phrase in terms of the dataset median tempo, the chosen values for r yield results with higher precision compared to the other methods. We assume the relatively homogeneous nature of our dataset minimized the impact of different tempos in the test data. For more diverse styles of music, including the tempo and beat grid information, similar to MIK, might be beneficial. On the other hand, it might be possible to train a model on beat and cue detection simultaneously.

The beat detection could then be used during postprocessing to identify the tempo, making the need for additional ground-truth beat grid or tempo information redundant.

One key limitation remains in the availability of training data, despite building our own dataset. Since we only had access to data with high similarity in style, we would like to investigate the performance of our method over a broader domain of electronic music in the future. Furthermore, our dataset annotations were provided by DJs who specialize in club DJing. Therefore, annotations from other types of DJs, such as scratch DJs or mobile DJs, would likely result in a largely different cue point distribution. We believe one main difference would lie in more cue points distributed around vocals or pickups instead of the first downbeat of phrases.

5. CONCLUSION

In this work we introduced CUE-DETR, an object detection model fine-tuned on Mel spectrograms capable of estimating cue points in EDM tracks. Candidate cue points produced by CUE-DETR demonstrate high adherence to the underlying music structure and exhibit a higher resemblance to manually placed cue points compared to previous approaches. Furthermore, we created EDM-CUE, a dataset containing 21k manually annotated cue points from four professional DJs. EDM-CUE also contains tempo, beat, and downbeat annotations for almost 5k EDM tracks. Our implementation includes a postprocessing step to filter the model predictions for the best positions, including a conversion of the results to timestamps. For the evaluation, we presented a complementary phrasing-based evaluation method, which is useful to assess cue point predictions in a more objective manner.

Furthermore, we demonstrated that CUE-DETR is capable of detecting large structural boundaries in music, despite only seeing small excerpts of the entire track. Our findings further acknowledge the potential of transformer-based architectures for the detection of time-based events in music.

6. REFERENCES

- [1] M. J. Butler, *Unlocking the groove: Rhythm, meter, and musical design in electronic dance music*. Indiana University Press, 2006.
- [2] M. Zehren, M. Alunno, and P. Bientinesi, “M-djcue: A manually annotated dataset of cue points,” 2019.
- [3] D. Cliff, “Hang the dj: Automatic sequencing and seamless mixing of dance-music tracks,” *Tech. report, HP Laboratories*, 2000.
- [4] H. Ishizaki, K. Hoashi, and Y. Takishima, “Full-automatic dj mixing system with optimal tempo adjustment based on measurement function of user discomfort,” in *International Society for Music Information Retrieval Conference*, 2009.
- [5] T. Hirai, H. Doi, and S. Morishima, “Musicmixer: computer-aided dj system based on an automatic song mixing,” *Proceedings of the 12th International Conference on Advances in Computer Entertainment Technology*, 2015.
- [6] L. Veire and T. Bie, “From raw audio to a seamless mix: creating an automated dj system for drum and bass,” *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2018, 2018.
- [7] A. Kim, S. Park, J. Park, J.-W. Ha, T. Kwon, and J. Nam, “Automatic dj mix generation using highlight detection,” *Proc. ISMIR, late-breaking demo paper*, 2017.
- [8] H.-W. Huang, M. Fadli, A. K. Nugraha, C.-W. Lin, and R.-G. Cheng, “Ai dj system for electronic dance music,” *2022 International Symposium on Electronics and Smart Devices (ISESD)*, pp. 1–6, 2022.
- [9] B.-Y. Chen, W.-H. Hsu, W.-H. Liao, M. A. M. Ramirez, Y. Mitsufuji, and Y.-H. Yang, “Automatic dj transitions with differentiable audio effects and generative adversarial networks,” *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 466–470, 2021.
- [10] R. M. Bittner, M. Gu, G. Hernandez, E. J. Humphrey, T. Jehan, H. McCurry, and N. Montecchio, “Automatic playlist sequencing and transitions,” in *International Society for Music Information Retrieval Conference*, 2017.
- [11] T. Kim, M. Choi, E. Sacks, Y.-H. Yang, and J. Nam, “A computational analysis of real-world dj mixes using mix-to-track subsequence alignment,” *ArXiv*, vol. abs/2008.10267, 2020.
- [12] J. Foote, “Automatic audio segmentation using a measure of audio novelty,” *2000 IEEE International Conference on Multimedia and Expo. ICME2000. Proceedings. Latest Advances in the Fast Changing World of Multimedia (Cat. No.00TH8532)*, vol. 1, pp. 452–455 vol.1, 2000.
- [13] D. Schwarz, D. A. Schindler, and S. Spadavecchia, “A heuristic algorithm for dj cue point estimation,” in *Sound and Music Computing*, 2018.
- [14] M. Zehren, M. Alunno, and P. Bientinesi, “Automatic detection of cue points for dj mixing,” *ISMIR*, vol. abs/2007.08411, 2020.
- [15] D. Schwarz and D. Fourer, “Methods and datasets for dj-mix reverse engineering,” in *Perception, Representations, Image, Sound, Music: 14th International Symposium, CMMR 2019, Marseille, France, October 14–18, 2019, Revised Selected Papers 14*. Springer, 2021, pp. 31–47.
- [16] L. Werthen-Brabants, “Ground truth extraction & transition analysis of dj mixes,” 2018, master Thesis, Ghent University, Ghent, Belgium.
- [17] D. Schwarz and D. Fourer, “Unmixdb: A dataset for dj-mix information retrieval,” *19th International Symposium on Music Information Retrieval (ISMIR)*, 09 2018.
- [18] J. Schlüter and S. Böck, “Improved musical onset detection with convolutional neural networks,” *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6979–6983, 2014. [Online]. Available: <https://api.semanticscholar.org/CorpusID:1262099>
- [19] S. Durand, J. P. Bello, B. David, and G. Richard, “Robust downbeat tracking using an ensemble of convolutional networks,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, pp. 76–89, 2016.
- [20] K. Ullrich, J. Schlüter, and T. Grill, “Boundary detection in music structure analysis using convolutional neural networks,” in *International Society for Music Information Retrieval Conference*, 2014.
- [21] Y. Gong, Y.-A. Chung, and J. R. Glass, “Ast: Audio spectrogram transformer,” *ArXiv*, 2021.
- [22] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, “An image is worth 16x16 words: Transformers for image recognition at scale,” *International Conference on Learning Representations (ICLR)*, 2021.
- [23] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, “End-to-end object detection with transformers,” *ArXiv*, vol. abs/2005.12872, 2020.
- [24] K. Miyazaki, T. Komatsu, T. Hayashi, S. Watanabe, T. Toda, and K. Takeda, “Conformer-based sound event detection with semi-supervised learning and data augmentation,” in *Workshop on Detection and Classification of Acoustic Scenes and Events*, 2020.

- [25] I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” in *International Conference on Learning Representations*, 2017.
- [26] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, “Microsoft coco: Common objects in context,” 2015.
- [27] S. Forsgren and H. Martiros, “Riffusion - Stable diffusion for real-time music generation,” 2022. [Online]. Available: <https://github.com/riffusion/riffusion>
- [28] M. Davies, N. Degara Quintela, and M. Plumbley, “Evaluation methods for musical audio beat tracking algorithms,” Centre for Digital Music, Queen Mary University of London, Tech. Rep., 10 2009.

THE LISTENBRAINZ LISTENS DATASET

Kartik Ohri

MetaBrainz Foundation Inc.
lucifer@metabrainz.org

Robert Kaye

MetaBrainz Foundation Inc.
rob@metabrainz.org

ABSTRACT

The ListenBrainz listens dataset is a continually evolving repository of music listening history events submitted by all ListenBrainz users. Currently totalling over 800 million entries, each datum within the dataset encapsulates a timestamp, a pseudonymous user identifier, track metadata, and optionally MusicBrainz identifiers facilitating seamless linkage to external resources and datasets. This paper discusses the process of raw data acquisition, the subsequent steps of data synthesis and cleaning, the comprehensive contents of the refined dataset, and the diverse potential applications of this invaluable resource. Although not the largest dataset in terms of music listening events (yet), its distinctiveness lies in its perpetual evolution, with users contributing data daily. This paper underscores the significance of the ListenBrainz listens dataset as a significant asset for researchers and practitioners alike, offering insights into music consumption patterns, user preferences, and avenues for further exploration in the fields of music information retrieval and recommendation systems.

Keywords: novel datasets, digital archives, metadata, linked data

1. INTRODUCTION

The advent of digital music streaming has led to an explosion of data on user listening habits. As the most prevalent form of music consumption today, with streaming accounting for 84% of total U.S. music revenue in 2023¹, this data holds immense potential for understanding trends, developing recommendation systems, and personalizing the user experience. However, most of this data is locked within commercial platforms and inaccessible to researchers or the public [1]. This lack of transparency hinders open-source development and independent research efforts in the music information retrieval field. AI-driven music recommendation systems, personalized playlists, and even music generation algorithms rely heavily on vast datasets of user

¹ U.S. Recorded Music Revenues Data by Format taken from the RIAA U.S. Music Revenue Database <https://www.riaa.com/u-s-sales-database/>



© K. Ohri and R. Kaye. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** K. Ohri and R. Kaye, "The ListenBrainz Listens Dataset", in *Proc. of the 25th Int. Society for Music Information Retrieval Conf.*, San Francisco, United States, 2024.

behavior to function effectively [2]. Open access to music consumption habits datasets is crucial in ensuring that these algorithms are developed and trained in a manner that is fair, transparent, unbiased, and representative of diverse musical tastes.

This paper introduces the ListenBrainz listens dataset, explores its contents, and potential applications. We will discuss the unique characteristics that distinguish it from other music datasets and highlight its significance as a valuable resource for researchers, practitioners, and music enthusiasts alike. Our goal is to provide the research community with a valuable resource for analyzing evolving music consumption patterns, exploring user preferences, and advancing open-source music information retrieval systems.

2. RELATED WORK

A few public music listening history datasets exist, most built upon data extracted from the social music platform Last.fm. These include the Last.fm Dataset-360K [5]; the Last.fm Dataset-1K [5], the LFM-1B dataset [6] and the LFM-2B dataset [7]. The LFM-1B dataset² and the LFM-2B dataset³ are not available anymore due to licensing issues.

All of these datasets were superseded by the introduction of the Music Listening History Dataset (MLHD) in 2017. MLHD stands out as one of the largest and most comprehensive publicly available datasets of music listening histories even today. It contains over 27 billion timestamped listening events from 583,000 users, enriched with demographic information and MusicBrainz identifiers for linking with external resources [3]. MLHD has been extensively used in research on music recommendation, user behavior analysis, and temporal trends in music consumption. To our knowledge, no newer datasets of comparable size and scope surpassing MLHD have been released since, highlighting the continued relevance and value of this resource. However, it is no longer possible to update MLHD with new data from Last.fm as the API endpoints originally used to curate the dataset have now been taken down [8].

The Music Streaming Sessions Dataset (MSSD), unveiled by Spotify, takes a unique approach by centering on listening sessions rather than individual track plays. It encompasses 160 million sessions, each providing in-

² Hosting page for LFM-1B dataset
<http://www.cp.jku.at/datasets/LFM-1b/>

³ Hosting page for LFM-2B dataset
<http://www.cp.jku.at/datasets/LFM-2b/>

Feature	MLHD [3]	MSSD [4]	ListenBrainz
Source	Last.fm Scrobbles	Spotify Streaming Logs	ListenBrainz User Submissions
Size	27 Billion Listening Events	160 Million Listening Sessions	800+ Million (and growing) Listening Events
Scope	Individual Track Plays	Listening Sessions (upto 20 tracks)	Individual Track Plays
Content	Timestamp, Basic Track Metadata, Limited MBIDs, User Demographics	Timestamp, User Actions, Track Metadata, Audio Features, Playlist Snapshots	Timestamp, Extended Track Metadata, Comprehensive MBIDs Links
Updates	Static (Last Updated 2017)	Static (Last Updated 2019)	Dynamic (Continuously Updated)
Strengths	Large size, Comprehensive user demographics, MBIDs for linking	Focus on listening sessions, Includes audio features, Counterfactual evaluation subset	Continuously updated, User-controlled data, Diverse data sources (streaming, local files), Extended Metadata Coverage, MBIDs for linking

Table 1: Comparison of the important music listening datasets

sights into user actions within the session, audio features of the tracks, and corresponding track metadata [4]. While MSSD offers valuable data for analyzing the dynamics of listening sessions, its scope is more confined compared to MLHD. MSSD encompasses a smaller user base and covers a shorter time frame. As of today, the MSSD dataset is not available for download publicly ⁴.

A common limitation shared by all the mentioned datasets, including both MLHD and MSSD, is their static nature. They represent snapshots of data frozen at a particular moment in time, lacking updates since their initial release. This inherent static nature raises concerns about their ability to accurately reflect contemporary music consumption patterns and trends. Furthermore, these datasets are missing data on music released after their creation, potentially restricting their usefulness for research inquiries focused on recent musical trends and user preferences. Additionally, the track metadata provided in MLHD and MSSD is limited to basic information such as artist, track, and album names. In contrast, ListenBrainz allows users to submit any additional metadata they deem relevant alongside their listening events, providing a richer and more comprehensive dataset for analysis.

Table 1 offers a concise overview of the key characteristics and differences between the Music Listening History Dataset, the Music Streaming Sessions Dataset, and the ListenBrainz Listens Dataset.

3. BACKGROUND

3.1 Music Listening History

Music listening histories serve as extensive timelines of an individual’s music consumption, offering valuable insights into their preferences, habits, and evolving tastes. Aggregating these histories across different timeframes uncovers

broader patterns and trends in listening behavior [9] [10]. The open availability of such data holds immense potential for advancing music information retrieval research, enhancing recommendation systems, and fostering a deeper understanding of the relationship between individuals and music.

3.2 Data Donation

Data donation is a method of data collection which typically involves users proactively sharing their digital trace data, often by requesting and exporting their data from online platforms, with researchers [11]. Data donations are commonly used in the field of communications, especial social media, research [12]. The usefulness of data donations in music research being increasingly recognized as exemplified by the Fair Muse project [13].

3.3 The ListenBrainz Project

The ListenBrainz listens dataset has been developed as a part of the broader open source project, ListenBrainz ⁵. The project is maintained by the MetaBrainz Foundation, a non-profit organization dedicated to promoting open data initiatives in the music domain. The organization is renowned for its over two-decade-long stewardship of the comprehensive free and open source MusicBrainz database ⁶. All ListenBrainz data is generously licensed under the CC0 license, granting unrestricted use and creating a collaborative environment for research and development.

4. THE LISTENBRAINZ LISTENS DATASET

4.1 Data Collection

The ListenBrainz dataset is entirely crowdsourced, with users actively contributing their listening histories. Lis-

⁴ MSSD Dataset download page <https://www.aicrowd.com/challenges/spotify-sequential-skip-prediction-challenge>

⁵ <https://listenbrainz.org/>

⁶ History and details of the MetaBrainz Foundation Inc. and the MusicBrainz project can be found at <https://metabrainz.org/about>

	Traditional Data Donation	ListenBrainz
Data Acquisition	Users request data from platform and donate to researchers.	Data submitted directly to ListenBrainz (automatically or manually).
Temporality	One-time or infrequent bulk data donations.	Continuous, regular data contribution.
User Effort	Active user involvement required in export and donation	Minimizes user effort after initial setup.
Data Scope	Limited to a single platform or service.	Aggregates data from multiple sources.
User Control	Limited control post data donation.	Offers ongoing user control (editing, deletion, or contribution cessation).
Data Utilization	Often for specific research projects with limited broader application.	Continuously growing, multi-purpose dataset for diverse research and the music community.

Table 2: Data Collection: Traditional Data Donation vs. ListenBrainz Approach

tenBrainz’s data collection approach shares its ethos with traditional data donation approaches. Both involve voluntary participation and aim to provide transparency regarding data usage.

However, the traditional data donation approach has some limitations. The donated data is retrospective and represents a one-time export. Repeated donations require users to navigate potentially complex processes which discourage participation [12]. To overcome these limitations, ListenBrainz provides multiple ways for users to setup automatic submission of listening events from their music streaming platforms and local music players on a continuous basis. Table 2 sums up the differences between the traditional and ListenBrainz approach.

Users can submit their data through various methods.

1. APIs and local media players: ListenBrainz provides a free and open API⁷ allowing manual submission of listening histories and facilitates the development of plugins for music players, automating the process for seamless and reliable data collection⁸. There is a Last.fm compatible API available as well which allows existing Last.fm clients to readily integrate with ListenBrainz⁹.
2. Streaming services integration: ListenBrainz integrates with popular streaming services like Spotify, enabling users to effortlessly link their accounts and contribute their streaming listening history.
3. Mobile applications and browser extensions: Various mobile applications can be used to submit listen events from mobile devices. Browser Extensions like WebScrobber¹⁰ provide convenient tools for submitting listening data from web-based music platforms.
4. Import of streaming services data exports: ListenBrainz supports conventional data donation methods, allowing users to upload data packages from streaming platforms like Spotify’s extended stream-

ing data export.

It is important to note that ListenBrainz empowers users with complete control over their data. They can edit, delete, or export their listening history as desired, ensuring transparency and user agency.

```
{
  "user_id": 1,
  "user_name": "rob",
  "timestamp": 1720644002,
  "track_metadata": {
    "track_name": "Tokara",
    "artist_name": "Fakear",
    "release_name": "All Glows",
    "additional_info": {
      "duration_ms": 206230,
      "tracknumber": 9,
      "artist_mbids": [
        "7c707d22-1c9c-4e72-bc8d-640baa5e2ba5"
      ]
    },
    "release_mbid":
      ↪ "2524b5bd-03d2-48ea-b85c-8cdebc8bbfe4",
    "recording_mbid":
      ↪ "ba97f6e5-f4ff-404f-b95b-e3aabade5e2e",
    "submission_client": "navidrome",
    "submission_client_version": "0.51.0
      ↪ (fd61b29a)",
    "recording_msid":
      ↪ "886bf922-8041-4e02-9991-596ffebddb7a"
  }
},
"recording_msid":
  ↪ "886bf922-8041-4e02-9991-596ffebddb7a"
}
```

Listing 1: A listen event in the ListenBrainz dataset

4.2 Data Cleaning and Synthesis

ListenBrainz ensures data quality through a robust cleaning and synthesis process. Every listening event requires a UTC epoch timestamp, a user identifier assigned by ListenBrainz, track name, and artist name. The *additional_info* field permits users to submit free-form JSON data. This flexibility empowers users to contribute any relevant information they deem valuable, fostering a richer understanding of music listening behaviors. Commonly used additional metadata fields include release name, MusicBrainz identifiers, track position, duration, and music service or media player used. A MBID is a 36 character

⁷ ListenBrainz API documentation is available at <https://listenbrainz.readthedocs.io/en/latest/users/api-usage.html>

⁸ A list of known music player supporting ListenBrainz submission can be found at <https://listenbrainz.org/add-data/>

⁹ Last.fm compatible API documentation at <https://listenbrainz.readthedocs.io/en/latest/users/api-compat.html>

¹⁰ WebScrobber <https://web-scrobber.com/>

Universally Unique Identifier that is permanently assigned to each entity in the MusicBrainz database. The range of MusicBrainz identifiers (MBIDs) supported by the ListenBrainz dataset is broader than MLHD [3] and hence, opens doors to a wealth of additional information. For example, a release MBID allows access to detailed label data and cover art from the MusicBrainz ecosystem. Listing 1 shows an example of a listen history event in the ListenBrainz dataset.

To prevent duplicates, ListenBrainz employs a real-time deduplication system based on the unique combination of user ID, timestamp, and a MessyBrainz identifier (MSID). MSIDs are random UUIDs assigned to the hash of the track, artist, and release names, serving as a robust method for identifying unique listening events.

While submitting MusicBrainz identifiers (MBIDs) alongside listening events greatly enhances the dataset’s connectivity and analytical potential, it’s not always a straightforward task for users. Local music collections often lack MBIDs in their ID3 tags, necessitating additional efforts to improve metadata quality. ListenBrainz encourages users to utilize tools like MusicBrainz Picard¹¹ to tag their collections effectively. Tagging collections becomes impractical when users engage with music through streaming services, where control over metadata submission is limited. To address this challenge, ListenBrainz employs a sophisticated background service known as the MBID mapper. This service automatically searches and associates relevant MBIDs with listening events based on the available metadata, enriching the dataset’s interconnectedness which is very helpful in downstream analysis. The inner workings of the MBID mapper involve complex algorithms and matching techniques beyond the scope of this paper. The MBIDs linked by the mapper are stored separately from user-submitted identifiers, empowering users of the dataset to choose whether or not to incorporate them into their analyses.

4.3 Dataset Format and Updates

The ListenBrainz dataset is available in two formats: ListenBrainz full export Dumps and ListenBrainz Spark Dumps. The ListenBrainz full export dumps contain the entire data submitted to ListenBrainz split in monthly chunks. Monthly data is organized into JSON lines files within yearly directories, providing comprehensive information for each listening event. The ListenBrainz spark dumps consist of chronologically ordered parquet files offering a subset of relevant fields optimized for batch processing and analysis.

The entire dataset is updated every 15 days, while incremental dumps capturing the listening events of the last 24 hours are produced daily. This ensures researchers and developers have access to both the comprehensive historical record and the most recent trends in music consumption.

5. DATASET ANALYSIS

As of today, the ListenBrainz listens dataset boasts a substantial collection of 876 million listening events contributed by approximately 28,000 users. Impressively, 764 million of these entries have been successfully linked with MusicBrainz identifiers, allowing for deeper analysis and connections with external music information resources. The dataset encompasses a diverse musical landscape, representing 900 thousand artists, 2.07 million albums, and a staggering 12.1 million recordings. Table 3 provides a summary of these key figures and a comparison with the corresponding figures of the MLHD dataset.

	MLHD [3]	ListenBrainz
Users	583 K	28,419
Listens (All)	27 B	876 M
Listens (with MBIDs)	-	764 M
Recordings	7 M	12.1 M
Albums	900 K	2.07 M
Artists	555 K	900 K

Table 3: Comparison of the size of the MLHD and ListenBrainz dataset

While the number of users and listening events in ListenBrainz is currently smaller compared to MLHD, it excels in its coverage of musical content, with several times the number of unique recordings, albums, and artists represented. This richness shows the potential of ListenBrainz for exploring a wider range of musical tastes and preferences.

The additional metadata recorded by ListenBrainz introduces several innovative features not present in the MLHD dataset. Specifically, 11% of listening events in ListenBrainz include track number information, while 12% of entries offer track duration data, which facilitates the analysis of listening session lengths and potential skipping behaviors. Additionally, 68% of listening events record the submission client. Although more than half of these clients are from Last.fm imports and Spotify, the remaining entries encompass a diverse array of user setups, including self-hosted music servers such as Navidrome and Funkwhale, as well as popular applications like Plex, PanoScrobbler, and WebScrobbler. This additional metadata enables new research opportunities to explore platform-specific listening behaviors and the influence of various music access modes on consumption patterns.

The temporal span of the ListenBrainz dataset is noteworthy, encompassing listening events dating back to 2005 and extending to the present year, 2024. Figure 2 illustrates the distribution of listening events across different years. Notably, the ability to submit past listening data to ListenBrainz suggests that the representation of earlier years may continue to grow over time. The lower number of events for 2024 is expected, given that only a portion of the year has elapsed.

Figure 1 shows the global coverage of the ListenBrainz

¹¹ MusicBrainz Picard <https://picard.musicbrainz.org/>



Figure 1: Artist Origins: Logarithm of number of listens of artists originating from a country

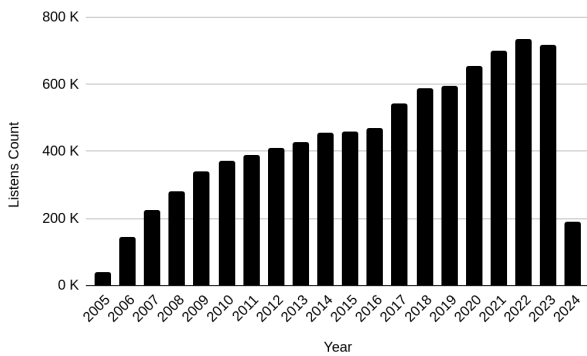


Figure 2: Temporal distribution of listening events

dataset, the artists in the dataset originate from a wide array of countries and regions. Although there is a noticeable concentration of artists originating from the United States, as evidenced by the darker shading, the dataset encompasses a diverse representation of artists from across the world particularly prominent in Europe, parts of South America, and Australia. This exploration also acts as an example of how MBIDs in the ListenBrainz dataset can be used to obtain useful information from the MusicBrainz database, in this case the country of an artist’s origin.

Figure 3 displays another temporal aspect of the dataset, the distribution of listening events based on the release year of the music. The graph reveals a clear trend towards a preference for newer music, with a significant surge in listening events observed from the 1990s onwards. This pattern aligns with the increasing availability and accessibility of digital music during this period. Nevertheless, the presence of listening events for music spanning several decades, dating back to the 1960s and earlier, emphasizes the assorted range of musical interests within the ListenBrainz community and the enduring appeal of older music.

6. USE CASES

The dataset is actively by the ListenBrainz project itself internally to power collaborative filtering algorithms that generate personalized recommendations, playlists, and engaging user reports. By combining these collaborative filtering techniques with content-based recommendations derived from MusicBrainz’s genre and folksonomy data, ListenBrainz creates a multifaceted and tailored music discovery experience for its users¹². In a further commitment to open-source music recommendation development, the ListenBrainz team has created the Troi recommendation toolkit¹³. This standalone toolkit adopts an API-first philosophy, enabling the construction of diverse and engaging playlists by utilising ListenBrainz data alongside other compatible datasets. Similarly, the Calliope project is an external initiative that leverages the ListenBrainz dataset to curate playlists and aid research and development in the field of open-source music recommendation systems¹⁴.

Beyond its applications in understanding general music preferences and trends, the listens data in ListenBrainz has proven valuable in exploring the impact of music recommendation diversity on listeners’ long-term attitudes and engagement [14]. Researchers have leveraged listens data available in ListenBrainz to develop and evaluate sequential music recommendation systems that utilize the powerful BERT transformer model [15].

Like MLHD, ListenBrainz is built upon a foundation of user-generated listening histories, making it conceptually similar and offering comparable data for analysis. Although the user base and overall size differ, the core data structure allows for the application of similar research methodologies and comparisons between findings. Addi-

¹² Weekly Recommendation Playlists <https://community.metabrainz.org/t/our-weekly-recommendations-are-now-live/646950?u=lucifer>

¹³ Troi recommendation toolkit <https://troi.readthedocs.io/en/latest/>

¹⁴ Calliope Project <https://calliope-music.readthedocs.io/>

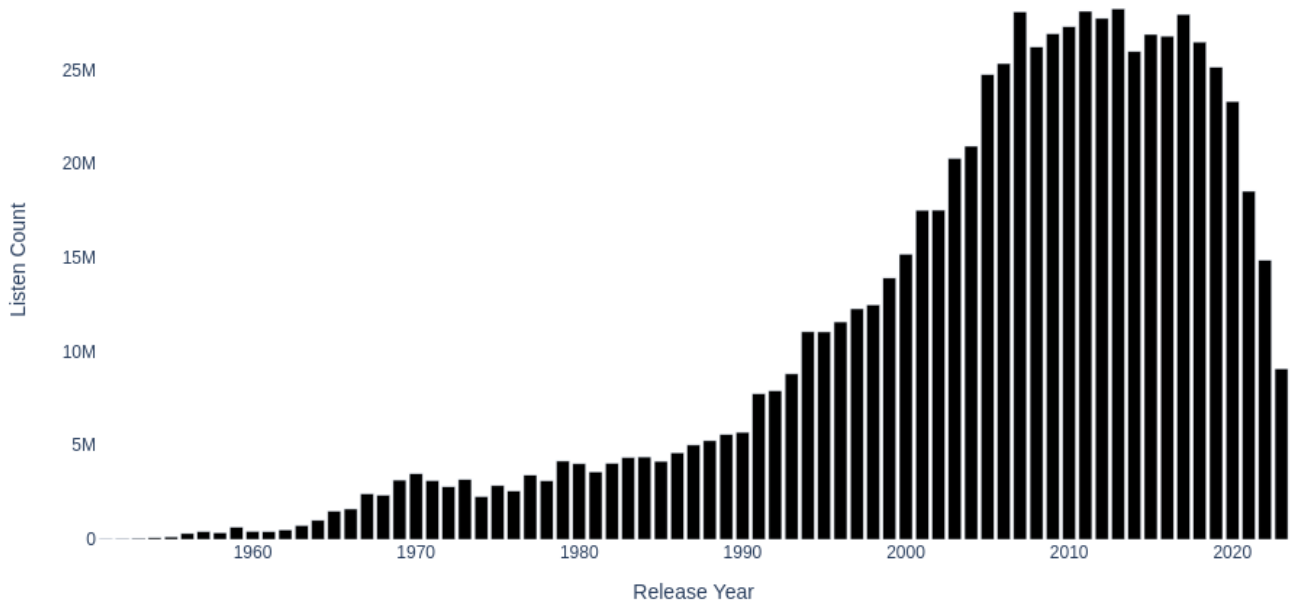


Figure 3: Listening events by original release year of albums

tionally, music sessions can be extracted from individual listening events of the ListenBrainz dataset to reproduce and extend studies initially conducted on the session-based MSSD dataset. Consequently, we are certain that this dataset holds immense potential for reproducing and validating previously conducted studies on similar datasets.

ListenBrainz also presents itself as a significant advancement in music consumption research tools. Instead of developing custom data collection and processing tools for data donations, researchers can leverage ListenBrainz. Researchers are relieved from the technical burdens and logistical complexities of data collection, allowing them to dedicate their time and resources to the core aspects of their studies and analytical inquiries. The ListenBrainz platform provides participants with insights into their listening behavior which can potentially increasing study engagement as well. In return, the listening events submitted by the participants enrich the overall listens dataset.

7. LIMITATIONS

The dataset utilizes UTC timestamps which prevents its usage in temporal analyses involving time zones, such as the diurnal music preferences explored by Park et. al [10]. Future iterations of the dataset aim to incorporate timestamps aligned with users' respective time zones, further enhancing its analytical capabilities.

The dataset can only as diverse as the individuals who choose to share their listening histories, potentially creating limitations in representing the full spectrum of music consumption across various cultures, genres, and communities. For instance, Figure 1 reveals a geographic bias in ListenBrainz's user demographics, with a disproportionate number of users located in the Anglosphere. Efforts are underway to integrate demographic data, such as user region and gender, to provide additional context to detect and eliminate such biases.

An inherent challenge within music listening datasets, including ListenBrainz, is the difficulty in discerning

whether a listening event reflects a user's genuine music preference or merely their exposure to a track due to algorithmic recommendations or shuffle mechanisms within music streaming services. This ambiguity makes it difficult to determine if a specific listening event represents an active choice by the user or a passive encounter with a suggested track.

Further, growing concerns surrounding online privacy may lead users to be hesitant in sharing their personal data, including seemingly benign information like music listening habits, impacting the growth of the dataset. Individuals are becoming increasingly aware of data collection practices and harbor reservations about potential privacy risks and the possible misuse of their information [16].

8. CONCLUSION

In conclusion, the ListenBrainz listens dataset provides a rich and dynamic resource for understanding the complexities of music consumption. Its comprehensive collection of user listening histories, accurate to the second, offers valuable insights into individual preferences and general trends. The inclusion of MusicBrainz identifiers further enhances its utility, enabling seamless integration with external music databases and facilitating in-depth analyses.

To reiterate, the ListenBrainz listens dataset addresses a significant gap in the field by providing a continuously updated resource that can represent rapidly changing music preferences. As the ListenBrainz project is run by a non-profit entity devoid of vested corporate interests, we believe that it will emerge as an indispensable resource for future research endeavors. By embracing openness, user agency, and continuous growth, ListenBrainz listens dataset paves the way for a deeper understanding of how we engage with music.

The dataset can be downloaded from <https://data.metabrainz.org/pub/musicbrainz/listenbrainz/fullexport/>.

9. REFERENCES

- [1] S. M. West, "Data capitalism: Redefining the logics of surveillance and privacy," *Business & Society*, vol. 58, no. 1, pp. 20–41, 2019. [Online]. Available: <https://doi.org/10.1177/0007650317718185>
- [2] K. Choi, G. Fazekas, K. Cho, and M. Sandler, "A tutorial on deep learning for music information retrieval," 2018.
- [3] G. Vigiensoni and I. Fujinaga, "The music listening histories dataset," in *Proceedings of the 18th International Society for Music Information Retrieval Conference*, Suzhou, People's Republic of China, 2017, pp. 96–102.
- [4] B. Brost, R. Mehrotra, and T. Jehan, "The music streaming sessions dataset," in *The World Wide Web Conference*, ser. WWW '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 2594–2600. [Online]. Available: <https://doi.org/10.1145/3308558.3313641>
- [5] O. Celma, *Music recommendation and discovery*. Springer-Verlag Berlin Heidelberg, 2010.
- [6] M. Schedl, "The lfm-1b dataset for music retrieval and recommendation," in *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval*, ser. ICMR '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 103–110. [Online]. Available: <https://doi.org/10.1145/2911996.2912004>
- [7] A. B. Melchiorre, N. Rekabsaz, E. Parada-Cabaleiro, S. Brandl, O. Lesota, and M. Schedl, "Investigating gender fairness of recommendation algorithms in the music domain," *Information Processing & Management*, vol. 58, no. 5, p. 102666, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0306457321001540>
- [8] G. Vigiensoni and I. Fujinaga, "Identifying time zones in a large dataset of music listening logs," in *Proceedings of the First International Workshop on Social Media Retrieval and Analysis*, ser. SoMeRA '14. New York, NY, USA: Association for Computing Machinery, 2014, p. 27–32. [Online]. Available: <https://doi.org/10.1145/2632188.2632203>
- [9] P. Herrera, Z. Resa, and M. Sordo, "Rocking around the clock eight days a week: an exploration of temporal patterns of music listening," in *1st Workshop On Music Recommendation And Discovery (WOMRAD), ACM RecSys, 2010, Barcelona, Spain*, 2010.
- [10] M. Park, J. Thom, S. Mennicken, H. Cramer, and M. Macy, "Global music streaming data reveal diurnal and seasonal patterns of affective preference," *Nature Human Behaviour*, vol. 3, no. 3, pp. 230–236, Mar 2019. [Online]. Available: <https://doi.org/10.1038/s41562-018-0508-z>
- [11] J. Ohme, T. Araujo, L. Boeschoten, D. Freelon, N. Ram, B. B. Reeves, and T. N. Robinson, "Digital trace data collection for social media effects research: Apis, data donation, and (screen) tracking," *Communication Methods and Measures*, vol. 18, no. 2, pp. 124–141, 2024. [Online]. Available: <https://doi.org/10.1080/19312458.2023.2181319>
- [12] N. Pfiffner and T. N. Friemel, "Leveraging data donations for communication research: Exploring drivers behind the willingness to donate," *Communication Methods and Measures*, vol. 17, no. 3, pp. 227–249, 2023. [Online]. Available: <https://doi.org/10.1080/19312458.2023.2176474>
- [13] G. Mazziotti and H. Ranaivoson, "Can online music platforms be fair? an interdisciplinary research manifesto," *IIC - International Review of Intellectual Property and Competition Law*, vol. 55, no. 2, pp. 249–279, Feb 2024. [Online]. Available: <https://doi.org/10.1007/s40319-023-01420-w>
- [14] L. Porcaro, E. Gómez, and C. Castillo, "Assessing the impact of music recommendation diversity on listeners: A longitudinal study," *ACM Trans. Recomm. Syst.*, vol. 2, no. 1, mar 2024. [Online]. Available: <https://doi.org/10.1145/3608487>
- [15] N. Yadav and A. K. Singh, "Bi-directional encoder representation of transformer model for sequential music recommender system," in *Proceedings of the 12th Annual Meeting of the Forum for Information Retrieval Evaluation*, ser. FIRE '20. New York, NY, USA: Association for Computing Machinery, 2021, p. 49–53. [Online]. Available: <https://doi.org/10.1145/341501.3441503>
- [16] A. Acquisti, L. Brandimarte, and G. Loewenstein, "Privacy and human behavior in the age of information," *Science*, vol. 347, no. 6221, pp. 509–514, Jan. 2015.

SPECMASKGIT: MASKED GENERATIVE MODELING OF AUDIO SPECTROGRAMS FOR EFFICIENT AUDIO SYNTHESIS AND BEYOND

Marco Comunità^{*1,2} Zhi Zhong^{*2} Akira Takahashi² Shiqi Yang² Mengjie Zhao²
 Koichi Saito³ Yukara Ikemiya⁴ Takashi Shibuya⁴ Shusuke Takahashi² Yuki Mitsufuji^{2,3}

¹ Queen Mary University of London, UK ² Sony Group Corporation, Japan

³ Sony AI, US ⁴ Sony AI, Japan

m.comunita@qmul.ac.uk, Zhi.Zhong@sony.com

ABSTRACT

Recent advances in generative models that iteratively synthesize audio clips sparked great success in text-to-audio synthesis (TTA), but at the cost of slow synthesis speed and heavy computation. Although there have been attempts to accelerate the iterative procedure, high-quality TTA systems remain inefficient due to the hundreds of iterations required in the inference phase and large amount of model parameters. To address these challenges, we propose SpecMaskGIT, a light-weight, efficient yet effective TTA model based on the masked generative modeling of spectrograms. First, SpecMaskGIT synthesizes a realistic 10s audio clip in less than 16 iterations, an order of magnitude less than previous iterative TTA methods. As a discrete model, SpecMaskGIT outperforms larger VQ-Diffusion and auto-regressive models in a TTA benchmark, while being real-time with only 4 CPU cores or even 30× faster with a GPU. Next, built upon a latent space of Mel-spectrograms, SpecMaskGIT has a wider range of applications (*e.g.*, zero-shot bandwidth extension) than similar methods built on latent wave domains. Moreover, we interpret SpecMaskGIT as a generative extension to previous discriminative audio masked Transformers, and shed light on its audio representation learning potential. We hope that our work will inspire the exploration of masked audio modeling toward further diverse scenarios.

1. INTRODUCTION

Text-to-audio synthesis (TTA) allows users to synthesize realistic audio and sound event signals by natural language prompts. TTA can assist the sound design and editing in the music, movie, and game industries, accelerating creators’ workflow [1]. Therefore, TTA has earned increasing attention in the research community.

Recent advances in deep generative models, especially iterative methods such as diffusion [2–5] and auto-

^{*}Equal contribution. Marco Comunità was an intern at Sony.

© M. Comunità and Z. Zhong. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** M. Comunità and Z. Zhong, “SpecMaskGIT: Masked Generative Modeling of Audio Spectrograms for Efficient Audio Synthesis and Beyond”, in *Proc. of the 25th Int. Society for Music Information Retrieval Conf.*, San Francisco, United States, 2024.

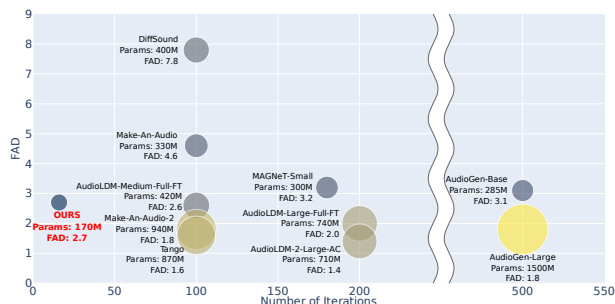


Figure 1. Audio synthesis performance and number of synthesis iterations of different methods. The size of circle represents the model size. SpecMaskGIT achieves good quality with only 16 iterations and a small model size.

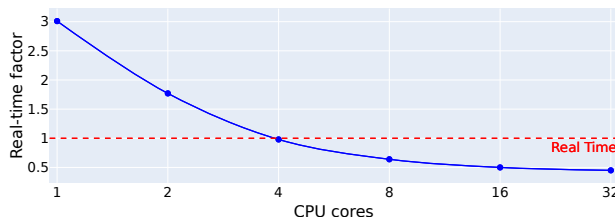


Figure 2. Real-time factor of SpecMaskGIT on different Xeon CPU cores with standard Python implementation.

regressive models [6–8], have brought significant success to the sound quality and controllability in TTA tasks, but at the cost of slow synthesis speed. Since the synthesis speed of iterative methods is dominated by the number of iterations required at inference, techniques have been introduced to reduce iterations, *e.g.*, higher compression rate of raw audio signals [6] or more efficient diffusion samplers [4, 9]. Nevertheless, these iterative methods remain slow in synthesis speed and demanding for computing resources, as they typically require hundreds of iterations to synthesize a short audio clip. Moreover, the runtime of a single iteration increases due to the huge model size.

To further improve inference efficiency, Garcia *et al.* introduced the MaskGIT [10] synthesis strategy from computer vision to the audio domain and proposed VampNet [11]. Although VampNet can inpaint a 10-second clip with 24 iterations, 6 seconds are needed on GPU [11], which is still heavy for non-GPU environments. Moreover, VampNet is not compatible with text prompts or TTA tasks. Concurrent to our work, MAGNeT extended VampNet to text-conditional audio synthesis [12]. However, the method is less efficient as it requires 180 iterations, which is heav-

ier than some diffusion models that only require 100 iterations [4, 9, 13, 14]. Since both VampNet and MAGNeT work in a wave-domain latent space, it is difficult to conduct frequency-domain inpainting tasks such as bandwidth extension (BWE) in a zero-shot manner. Besides the aforementioned limitations, the audio representation learning potential of a masked generative Transformer has not been investigated yet.

As a summary, an audio synthesis method that is compatible with text prompts, highly efficient in synthesis speed, and flexible for various downstream tasks is yet to be explored. To this end, we propose SpecMaskGIT, an efficient and flexible TTA model based on the masked generative modeling of audio spectrograms. Our contributions lie in the following aspects:

- **Efficient and effective TTA.** SpecMaskGIT synthesizes a realistic 10-second audio clip in less than 16 iterations, which is one order of magnitude smaller than previous iterative methods (Fig. 1. As a discrete generative model, SpecMaskGIT outperforms larger VQ-Diffusion (Diff-Sound [2]) and auto-regressive (AudioGen-base [6]) models in a TTA benchmark, while being real-time with 4 CPU cores (Fig. 2) or even $30\times$ faster on a GPU.
- **Flexibility in downstream tasks.** SpecMaskGIT is interpreted and implemented as a generative extension to previous discriminative audio masked Transformers [15–18]. The masked spectrogram modeling principle and architecture design similar to Audio Masked Auto-encoder (MAE) [16–18] is believed to have contributed to the representation learning potential of SpecMaskGIT. Unlike prior art about finetuning MAE-like architectures for BWE [18, 19], SpecMaskGIT enables BWE in a zero-shot manner.

We hope this efficient, effective and flexible framework paves the way to the exploration of masked audio modeling toward further diverse scenarios [20].¹

2. RELATED WORKS

Synthesizing audio signals in raw waveform is challenging and computationally demanding [21]. Therefore, the mainstream approach to audio synthesis is to first generate audio in a compressed latent space, and then restore waveforms from latent representations. Auto-regressive models such as Jukebox [22], AudioGen [6] and MusicGen [23] use vector-quantized (VQ) variational auto-encoders (VAE) [24] to tokenize raw waveforms into a discrete latent space. While AudioGen and MusicGen use a higher compression rate than Jukebox, 500 iterations are required to synthesize a 10-second clip, slowing generation down.

Advances in audio representation learning such as Audio MAE ([16–18]) indicate that Mel-spectrogram is an effective compression of raw audio signals, as it emphasizes acoustic features of sound events while maintaining sufficient details to reconstruct raw waveforms. Inspired by the above success of representation learning, several methods used discrete [2] or continuous [3, 4, 9, 13, 14] diffusion

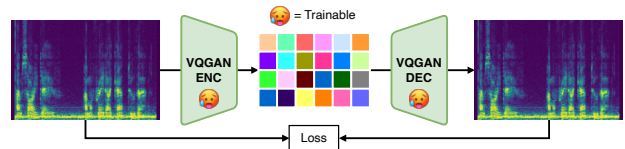


Figure 3. SpecVQGAN encodes/decodes non-overlapping 16-by-16 time-mel patches into/from discrete tokens.

models upon the latent Mel-spectrogram space created by a VAE or SpecVQGAN [25]. These diffusion models require up to 200 iterations for high-fidelity synthesis, which is still challenging for low-resource platforms and interactive use cases. While distilling a diffusion model can effectively reduce the required iterations [26–28], we limit our discussion to non-distilled methods for a fair comparison. For Mel-based synthesis methods, waveforms are reconstructed from Mel-spectrogram with a neural vocoder, such as HiFiGAN [29] or BigVSAN [30].

In pursuit of higher synthesis efficiency, VampNet [11] and the concurrent MAGNeT [12] adopted the parallel iterative synthesis strategy from MaskGIT [10]. Originally proposed for class-conditional image synthesis tasks, MaskGIT uses a Transformer with bi-directional attention - instead of the uni-directional counterpart of auto-regressive methods - to reduce the required number of iterations. Although VampNet and MAGNeT reduced the number of iterations compared to their auto-regressive counterparts, VampNet does not support text prompts, while MAGNeT takes 180 iterations, which is even heavier than some diffusion models that only require 100 iterations [4, 9, 13, 14]. Moreover, it is difficult for methods built upon wave-domain latent spaces to address frequency domain tasks such as BWE, limiting their applications.

3. SPECMASKGIT

The efficiency, effectiveness and flexibility of SpecMaskGIT is due to a combination of efforts, including among other, the high compression rate in the tokenizer, the small model size, and the fast synthesis algorithm.

3.1 Spectrogram Tokenizer and Vocoder

A modified SpecVQGAN [25] is trained to tokenize non-overlapping 16-by-16 time-mel patches into discrete tokens, and recover the tokens back to Mel-spectrogram as in Fig. 3. Reconstructed Mel-spectrograms are then transformed to waveforms by a pre-trained vocoder. On top of the $3.2\times$ compression offered by the wave-to-mel transform in our configuration, SpecVQGAN further offers $256\times$ compression of the spectrogram, resulting in a total of over $800\times$ compression of the raw waveform, effectively reducing the number of tokens to synthesize.

We utilize the standard Mel transform widely used in vocoders [29–32] for optimal Mel computation, as hyperparameters of Mel transform have an impact on tokenizers’ performance [9]. To stabilize the training, we keep the spectrogram normalization in the original SpecVQGAN, which clips Mel bins lower than -80 dB or louder than 20 dB, and then maps the spectrogram into the $[-1.0, 1.0]$ range. Our modified SpecVQGAN is shown competitive in reconstruction quality in Sec. 5.1.

¹ Demo: <https://zzaudio.github.io/SpecMaskGIT>

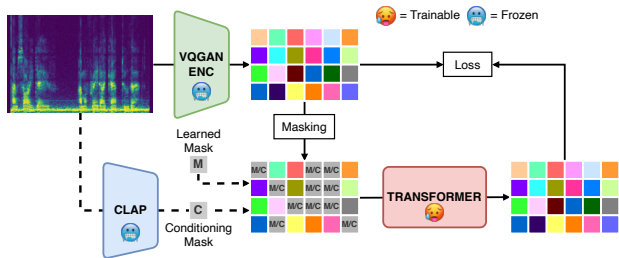


Figure 4. Self-supervised training of SpecMaskGIT. The Transformer is trained to reconstruct SpecVQGAN token sequences - randomly masked with varying ratios - unconditionally via a learned mask token ("M"); or conditioned on a semantic token from the CLAP encoder ("C").

3.2 Masked Generative Modeling of Spectrograms

We train a masked generative Transformer upon the discrete latent space of the pretrained SpecVQGAN as in Fig. 4. First, the pretrained CLAP encoder [33] maps the input audio to a semantic embedding aligned with its corresponding text descriptions. Meanwhile, the input audio is tokenized by SpecVQGAN. Finally, similar to representation learning such as Audio MAE [16–18], a bi-directional Transformer is trained to reconstruct Mel-spectrogram token sequences from a randomly masked input.

There are two major differences from Audio MAE. First, the masking ratio is *not* a fixed value but sampled on-the-fly from a truncated Gaussian distribution that is centered at 55% [34] and ranges from 0% to 100% [10]. As a result, although in each training step SpecMaskGIT behaves similarly to Audio MAE, it learns the training data distribution from various masking ratios, hence gaining the ability to iteratively refine audio tokens by gradually decreasing the masking ratio across multiple iterations, which is explained in Sec. 3.4. Second, while Audio MAE works on raw Mel-spectrogram, optimizing the mask reconstruction by mean square error; SpecMaskGIT works in a discrete latent space, which means the reconstruction of a masked position evolves to retrieval of the correct code from the SpecVQGAN codebook, *i.e.*, a multi-class single-label classification procedure. Therefore, the loss function becomes the cross entropy (CE) loss with label smoothing equal to 0.1. Following Audio MAE, visible positions in the input are not considered in the loss calculation:

$$\text{Loss} = \text{CE}(\text{prediction}[\text{mask}], \text{label}[\text{mask}]). \quad (1)$$

3.3 Text Conditioning via Sequential Modeling

Similarly to [4], we train SpecMaskGIT without audio-text pairs by using a pretrained CLAP model [33], for which audio and text embeddings are aligned in a shared latent space. Leveraging such alignment, after training with the audio branch of CLAP (see Fig. 4), we can directly condition our pretrained model with the text branch as shown in Fig. 5. We use a publicly available CLAP checkpoint ("630k-audioset-best.pt" [33]) for better reproducibility.

Although the above design is inspired by AudioLDM [4], SpecMaskGIT is different in the way CLAP embeddings are injected. Besides the FiLM mechanism ([35]) used in AudioLDM, prior works inject text conditions via

cross-attention [2, 3, 9, 13, 14], even for methods based on sequential modeling such as AudioGen [6] and MAG-NeT [12], which inevitably involves efforts to modify basic DNN modules. We believe that reusing modules, such as the Vision Transformer (ViT) [36], across different tasks is beneficial for efficient development, so we choose to achieve text-conditional audio synthesis by pure sequential modeling, *i.e.*, prepending the CLAP embedding to the input sequence to the Transformer. Note that the CLAP embedding is mapped to the same dimension as the Transformer by a linear layer in advance. As a result, SpecMaskGIT can be implemented with the same ViT used in Audio MAE [16–18], thus we view SpecMaskGIT as a generative extension to previous discriminative masked spectrogram modeling methods. We hypothesize the masked modeling and ViT implementation similar to Audio MAE has contributed to the representation learning potential of SpecMaskGIT, as is shown in Sec. 5.2.

While the common practice in [10, 16–18] is to use a learnable but input-independent token to indicate which parts in the sequence are masked ("M" in Fig. 4), the mask reconstruction task is challenging as the input-independent mask offers no hint for a better reconstruction. To further guide the mask reconstruction procedure, we propose to directly use the input-dependent CLAP embedding as a conditional mask ("C" in Fig. 4), which offers semantic hints like "a dog barking sound" to the model, and is found beneficial to TTA performance in Sec. 5.1.

3.4 Iterative Synthesis with Classifier-free Guidance

We follow the parallel iterative synthesis strategy proposed in MaskGIT [10] in general, but additionally employ classifier-free guidance (CFG) [37] to improve the synthesis quality. This iterative algorithm allows SpecMaskGIT to synthesize multiple high-quality tokens at each iteration, reducing the number of iterations to a value one order of magnitude smaller than previous TTA methods.

To enable CFG, we replace the CLAP embedding with the learned mask token on a random 10% of training steps. At inference phase, both the conditional (ℓ_c) and unconditional (ℓ_u) logits for each masked token are computed. The final logits ℓ_g are made by a linear combination of the two logits based on t , the guidance scale:

$$\ell_g = \ell_u + t(\ell_c - \ell_u). \quad (2)$$

Intuitively, CFG balances between diversity and audio-text alignment. Inspired by [38], we introduce a linear scheduler to the guidance scale t , which linearly increases t from 0.0 to an assigned value through the synthesis iterations. This allows the result of early iterations to be more diverse (unconditional) with low guidance, but increases the influence of the conditioning for late iterations, and is proved beneficial to synthesis quality in Sec. 5.1.

The parallel iterative synthesis of SpecMaskGIT shown in Fig. 5 is explained as follows:

1. Estimating. For each masked position, the Transformer estimates the probability of each code in the SpecVQGAN codebook to be the correct one, *i.e.*, the categorical distribution in the SpecVQGAN latent space.

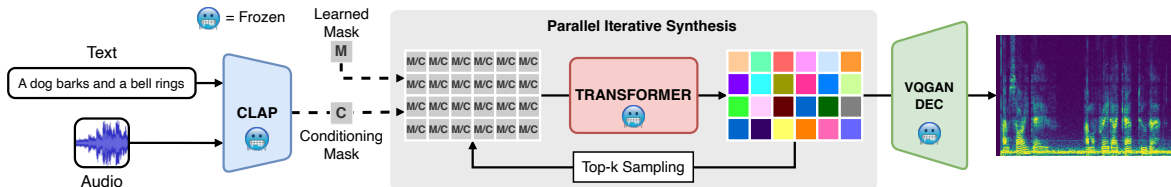


Figure 5. The iterative text-to-audio synthesis in SpecMaskGIT.

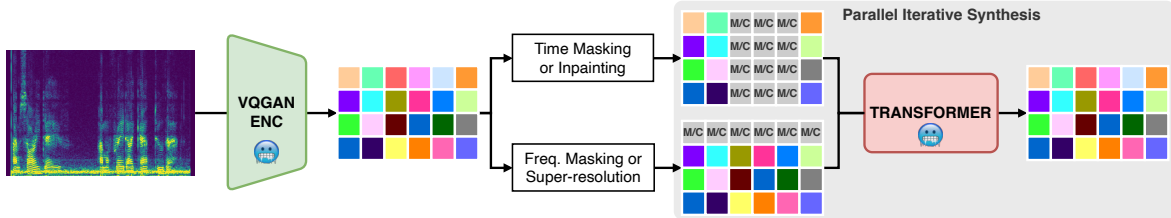


Figure 6. Zero-shot time inpainting and bandwidth extension for general audio data via SpecMaskGIT.

2. Unmasking. Given the categorical distribution over the codebook for each masked position, a code is randomly sampled. This step is different from the deterministic unmasking in Audio MAE.

3. Scheduling. Although SpecMaskGIT can unmask all positions at once, the quality of the synthesized audio is low. To iteratively refine the synthesis, we need to re-mask the result to a masking ratio that is lower than the current iteration. We follow the common practice in [10–12, 34] to use a cosine scheduler to decide the masking ratio at each iteration. The cosine scheduler re-masks a larger portion of the synthesized tokens for early iterations, which is intuitive as the quality in earlier iterations is lower.

4. Top-*k* sampling. Given the masking ratio for the next iteration, we know *k* tokens are going to be re-masked. The log-likelihood of unmasked tokens is used to decide the *k* worst tokens. Since it is observed that a deterministic top-*k* retrieval leads to the synthesis of monotonous images in [39], we follow [11, 34] and add Gumbel noise to the log-likelihood, making the top-*k* sampling stochastic:

$$\text{confidence} = \log(p) + t_{\text{gumbel}} \cdot n_{\text{gumbel}}, \tag{3}$$

where *p* is the probability of each unmasked token calculated from the CFG logits in Eq. 2, *n_{gumbel}* is the Gumbel noise, and *t_{gumbel}* is the noise temperature. Following [34], we linearly anneal *t_{gumbel}* by a coefficient defined as *iter/num_iter*, with “*iter*” index of the current iteration and “*num_iter*” the total number of scheduled iterations.

5. Repeating. Repeat all steps until the cosine scheduler reduces the masking ratio to 0.

For TTA, SpecMaskGIT starts the above iterative procedure from a fully masked sequence as in Fig. 5. Nevertheless, the iterative algorithm is also valid when the masking ratio of an input sequence is lower than 100%, which automatically enables zero-shot inpainting in both time and frequency domain as is shown in Fig. 6. It is worth noticing that since VampNet [11] and MAGNeT [12] employ a wave-domain tokenizer, frequency inpainting or bandwidth extension (BWE) are difficult.

4. EXPERIMENTS

We pretrained the SpecVQGAN [25] and two vocoders (HiFiGAN [29] & BigVSAN [32]) on AudioSet (AS) un-

balanced and balanced subset [40] for 1.5M steps. The AS we collected contains around 1.8 million 10-second audio segments of diverse sound sources and recording environments. AS has been widely used in general audio representation learning [16–18]. We followed the “VGGSound” configuration in the original SpecVQGAN repository [25] without using LPAPS loss as suggested in the repository itself. Our SpecVQGAN has around 75M parameters, and a codebook of 1024 codes, each of which is represented by a 256-dim embedding. As mentioned in Sec. 3.1, the standard Mel-spectrogram transform from vocoders [29, 30] is utilized, which transforms a 10-second audio clip at sampling rate 22.05kHz into 848 frames with 80 Mel bins. The Mel-spectrogram is further tokenized into 265 tokens.

SpecMaskGIT employs the ViT implementation widely used in previous audio masked Transformers [15, 16, 18, 41]. To be consistent with the image MaskGIT [10], 24 Transformer blocks are used, in which the attention dimension is 768 with 8 heads and the feedforward dimension is 3072, resulting in around 170M parameters. We trained SpecMaskGIT on AS for 500k steps with a batch size of 112. When training the model on AudioCaps (AC) [42], we train for 250k steps with a batch size of 48, as AC only contains 50k 10-second audio clips. To stably train SpecMaskGIT, we follow the common practice in [16–18] to employ a linear warmup and then a cosine annealing of the learning rate (LR). We warmup 16k steps for AS and 5k steps for AC. The base LR is set to 1e-3, and the LR equates to the base LR times the batch size divided by 256 [17, 34]. The iterative synthesis algorithm is based on the open-source implementation of [34].

To evaluate the **TTA synthesis** quality of SpecMaskGIT, we benchmark on the AudioCaps (AC) test set with the text prompts released by [4] for fair comparison. To investigate the flexibility of SpecMaskGIT in downstream tasks, we use the checkpoint trained on AS for 500k steps in the following tasks: **Zero-shot time inpainting.** We manually mask out the 25th to 35th Mel-spec frames (around 1.9s) of AC test set, and employ SpecMaskGIT to inpaint the lost regions in a zero-shot manner, *i.e.*, no task-specific finetuning. **Zero-shot audio bandwidth extension.** The top 16 Mel-spec bins (*i.e.*, components beyond 4.3kHz) of AC test set are masked, which creates

Table 1. Comparing SpecMaskGIT with other discrete TTA methods on AudioCaps test set.

Method	Params	Text	Num_iter	FAD
DiffSound [2]	400M	Yes	100	7.8
MAGNeT-small [12]	300M	Yes	180	3.2
AudioGen-base [6]	285M	Yes	500	3.1
AudioGen-large [6]	1.5B	Yes	500	1.8
SpecMaskGIT (ours)				2.7
- w HiFiGAN				2.8
- w/o conditional mask	170M	No	16	3.2
- w/o CFG				3.1
- w/o CFG linear scheduler				3.1

a $2.5\times$ BWE task. For all tasks above, we compute the Fréchet Audio Distance (FAD) using [43] since FAD is widely adopted to evaluate TTA [4, 9, 13, 14], time inpainting [4] and BWE [44] tasks. To investigate the representation learning potential of SpecMaskGIT, we further linear probe the model for the multi-label (genre, instrument and mood) **music tagging** task in MagnaTagATune (MTAT) [45] - a dataset widely used to evaluate music tagging models [46–49] - with ROC-AUC and mAP as metrics [46]. We use a single linear layer with batch normalization and 0.1 dropout as the probe.

5. RESULTS

5.1 Text-to-audio Synthesis

We report FAD scores of SpecMaskGIT in Tab. 1 together with other discrete models. Our model is first trained on AS for 500k steps and then finetuned on AC train set for 250k steps. The CFG scale is set to 3.0 empirically. SpecMaskGIT outperforms DiffSound (VQ-Diffusion), MAGNeT-small (similar to SpecMaskGIT but in latent wave domain), as well as AudioGen-base (auto-regressive) in terms of FAD with one order of magnitude fewer iterations. The FAD score is achieved training without any audio-text pairs, which proves the effectiveness of such self-supervised approach for discrete models. We also find the proposed conditional mask described in Sec. 3.3 to improve FAD score without additional parameters or computations. Both CFG and its linear scheduler contribute to improve the FAD.

Given the small number of iterations and model size, SpecMaskGIT can synthesize realistic 10-second audio clips in real-time with only 4 cores of a Xeon CPU (Fig. 2), or $30\times$ faster than real-time on an RTX-A6000 GPU, making it attractive for interactive applications and low-resource environments.

When compared to state-of-the-art (SOTA) continuous diffusion models in Tab. 2, SpecMaskGIT could not achieve a comparable FAD score, but we emphasize that the proposed method offers good performance with high efficiency, *i.e.*, smaller model size and fewer iterations, which can be clearly seen in Fig. 1. Overall, continuous methods are advantageous in terms of FAD with respect to discrete methods. We leave the further improvement of our discrete generative model as future work.

Ablation study: Gumbel noise and iterations number. We use HiFiGAN in all ablation studies. As mentioned in Sec. 3.4, Gumbel noise is essential to the top- k sam-

Table 2. Benchmarking on AudioCaps test set. Dis.: discrete methods. Con.: continuous methods.

Method	Params	Dis.	Con.	Num_iter	FAD
DiffSound [2]	400M	✓		100	7.8
Make-an-Audio [3]	330M		✓	100	4.6
MAGNeT-small [12]	300M	✓		180	3.2
AudioGen-base [6]	285M	✓		500	3.1
AudioLDM-Medium-full-FT [4]	420M		✓	100	2.6
AudioLDM-Large-full-FT [4]	740M		✓	200	2.0
Make-an-Audio 2 [9]	940M		✓	100	1.8
AudioGen-large [6]	1.5B	✓		500	1.8
AudioLDM2-Small-AC [14]	350M		✓	200	1.7
TANGO-AC [13]	870M		✓	100	1.6
AudioLDM2-Large-AC [14]	710M		✓	200	1.4
SpecMaskGIT (ours)	170M	✓		16	2.7

pling during iterative synthesis. Fig. 7 shows that a temperature of 1.5 is optimal. SpecMaskGIT achieves good quality (FAD = 3.4) with only 8 iterations, and reaches its best (FAD = 2.8) with 16. More iterations do not improve performance, which is consistent with MaskGIT [10].

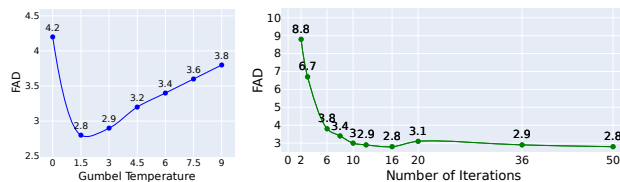


Figure 7. Left: FAD vs. Gumbel temperature. Right: FAD vs. Number of iterations.

Ablation study: Audio reconstruction quality. We evaluate the reconstruction FAD (rFAD) scores of two vocoders and SpecVQGAN in Tab. 3 with previous methods reported in [9]. Even with a similar architecture, rFAD of DiffSound and SpecMaskGIT can vary a lot due to different Mel computation and vocoder. Our pipeline achieves SOTA level rFAD scores for Mel-spectrogram methods while maintaining the highest compression rate (*i.e.*, the lowest latent rate) which helped SpecMaskGIT to outperform methods such as DiffSound and Make-an-audio by a large margin, yet with higher efficiency. We further analyze the rFAD of vocoders using ground truth input Mel-spectrograms, and find a significant performance gap between HiFiGAN and BigVSAN, which is not observed when vocoders are combined with SpecVQGAN. This indicates that SpecVQGAN is the bottleneck for reconstruction quality and asks for future improvements.

Ablation study: Bias in AudioCaps benchmark. The dataset gap between AC and other larger, more diverse datasets is investigated. It is observed in [4] that finetuning (FT) a TTA model on AC improves the TTA perfor-

Table 3. rFAD of Mel-spectrogram VAEs and Vocoders on AudioCaps test set. **Bold:** best overall rFAD.

Method	Mel-spec VAE	Vocoder	Latent rate	rFAD
DiffSound [2]	SpecVQGAN	MelGAN	27Hz	6.2
Make-an-audio [3]	VAE-GAN	HiFiGAN	78Hz	6.0
AudioLDM [4]	VAE-GAN	HiFiGAN	410Hz	1.2
Make-an-audio 2 [9]	VAE-GAN	BigVGAN	31Hz	1.0
SpecMaskGIT (ours)	-	HiFiGAN	27Hz	0.4
	SpecVQGAN	-	-	1.1
	SpecVQGAN	BigVSAN	27Hz	0.1
				1.0

Table 4. Music tagging performance on MTAT.

Method	CLMR [47]	MusiCNN [48]	MERT-330M [46]	MULE-contrastive [49]	Jukebox [22, 50]	SpecMaskGIT
mAP (%)	36.1	38.3	40.2	40.4	41.4	40.5
ROC-AUC (%)	89.4	90.6	91.3	91.4	91.5	91.5

Table 5. AC test set performance w/ or w/out AC finetune.

Method	Params	Num_iter	FAD	
			before FT	after FT
AudioLDM-Small-full [4]	180M	200	4.9	2.3
AudioLDM-Large-full [4]	740M	200	4.2	2.0
SpecMaskGIT (ours)	170M	16	4.2	2.8

Table 6. Small-scale AudioCaps training results in better scores than large-scale dataset.

Method	Params	Num_iter	FAD	
			Other datasets	AudioCaps
AudioLDM-Small [4]	180M	200	4.9	2.4
AudioLDM-Large [4]	740M	200	4.2	2.1
AudioLDM2-Small [14]	350M	200	2.1	1.7
AudioLDM2-Large [14]	710M	200	1.9	1.4
SpecMaskGIT (ours)	170M	16	4.2	2.9

mance in terms of FAD, though the model is pretrained on a larger dataset. We reproduced this phenomenon with SpecMaskGIT as shown in Tab. 5. We also observed that training on the small-scale AC alone brought better FAD score than the model trained with larger datasets in Tab. 6, which is consistent with [13, 14].

We hypothesize that there is a data distribution gap between AC and other datasets, such that when a model fully fits other datasets, the distribution of its synthesis deviates from AC, resulting in worse FAD. Therefore, we continued to train SpecMaskGIT on AS until 800k steps, and depict the “FAD vs. training step” curves on both the valid and test set of AC to verify our hypothesis. It is clear in Fig. 8 that SpecMaskGIT learns to synthesize audio in the early stage and keeps improving the FAD on AC. As the training goes on, SpecMaskGIT just fits toward AS, which worsens the FAD on AC.

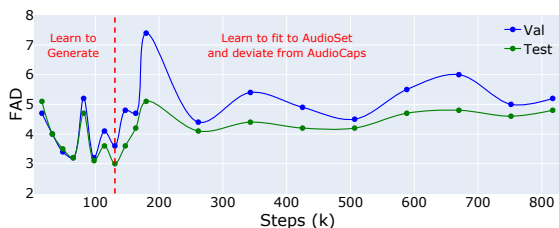


Figure 8. FAD vs. AudioSet training steps.

Inspired by audio classification tasks in which early stop is applied to prevent the model from overfitting to the train set, we propose to apply early stop to the SpecMaskGIT model trained solely on AS, and report the competitive FAD score with other methods that are without AC finetuning or AC-alone training in Tab. 7. We believe that a more comprehensive and less biased benchmark would contribute to future advances in TTA research.

5.2 Downstream Inpainting, BWE and Tagging Tasks

Results of the time inpainting and audio BWE tasks are shown in Tab. 8. We utilize the pipeline in Fig. 6 unconditionally, with Gumbel temperature 1.5 and 16 iterations. SpecMaskGIT significantly improves the input signals in terms of FAD, validating its zero-shot ability in

Table 7. Benchmarking on AudioCaps test set without AC finetuning or AC-alone training.

Method	Params	Dis.	Con.	Num_iter	FAD
DiffSound [2]	400M	✓		100	7.8
AudioLDM-Small-full [4]	180M		✓	200	4.9
Make-an-Audio [3]	330M		✓	100	4.6
AudioLDM-Large-full [4]	740M		✓	200	4.2
MAGNeT-small [12]	300M	✓		180	3.2
AudioGen-base [6]	285M	✓		500	3.1
AudioLDM2-Small-full [14]	350M		✓	200	2.1
AudioLDM2-Large-full [14]	710M		✓	200	1.9
Make-an-Audio 2 [9]	940M		✓	100	1.8
AudioGen-large [6]	1.5B	✓		500	1.8
SpecMaskGIT-AS-EarlyStop (ours)	170M	✓		16	2.9

Table 8. Zero-shot time inpainting and BWE FAD scores.

	BWE	Time inpaint
Unprocessed	2.7	1.6
SpecMaskGIT (ours)	1.5	1.2
- w/ LFR	0.4	-
Ground truth	0.0	0.0

such tasks. BWE performance can be further improved by applying low-frequency replacement (LFR) [51, 52]. Unlike prior arts that finetune MAE-like architectures for BWE [18, 19], SpecMaskGIT achieves it zero-shot. In Tab. 4, the potential of SpecMaskGIT in representation learning is confirmed by the music tagging performance on the MTAT dataset. As a TTA model, SpecMaskGIT outperforms classification-specialized models such as CLMR, MusiCNN, MULE, and MERT (the MAE-like model in wave domain). SpecMaskGIT achieves an ROC-AUC comparable to Jukebox, which contains 5B parameters. We hypothesize the tagging capability comes from the masked spectrogram modeling and ViT implementation similar to Audio MAE, as explained in Sec. 3. We leave the in-depth investigation of SpecMaskGIT in downstream tasks as future work.

6. CONCLUSION

Generative models that iteratively synthesize audio clips sparked great success to text-to-audio synthesis (TTA). However, due to the hundreds of iterations required for inference and the large amount of model parameters, high-quality TTA systems remain inefficient. To address the challenges, we propose SpecMaskGIT, a light-weight, efficient yet effective TTA model based on masked generative modeling of spectrograms. SpecMaskGIT synthesizes realistic audio clips in less than 16 iterations, an order of magnitude less than previous iterative TTA methods. It also outperforms larger discrete models in a TTA benchmark, while being real-time with 4 CPU cores and 30× faster with a GPU. Compared to similar methods, SpecMaskGIT is more flexible for downstream tasks such as zero-shot bandwidth extension. Moreover, we interpret SpecMaskGIT as a generative extension to Audio MAE and shed light on its audio representation learning potential. We hope our work inspires the exploration of masked audio modeling toward further diverse scenarios.

7. ETHICAL STATEMENT

SpecMaskGIT is supposed to assist creators in the sound design and editing workflow. Our method presents a huge advancement in the efficiency of TTA technology, which makes TTA accessible to a broader range of users, including creators who do not have GPUs. Despite of the technical advances, there is concern for the potential reflection of training data biases. The model may not be able to maintain a consistent sound quality or audio-text alignment when prompted by text descriptions or audio clips that are rarely presented in the training data. We also pointed out that the benchmark widely used to evaluate TTA models in the research community is biased, and hope our findings here can contribute to a less biased benchmark in the future. The challenge in dataset bias emphasizes the importance for in-depth consideration and collaboration with stakeholders across various communities.

8. REFERENCES

- [1] Y. Zhang, Y. Ikemiya, G. Xia, N. Murata, M. Martínez, W.-H. Liao, Y. Mitsufuji, and S. Dixon, “Musicmagus: Zero-shot text-to-music editing via diffusion models,” *arXiv preprint arXiv:2402.06178*, 2024.
- [2] D. Yang, J. Yu, H. Wang, W. Wang, C. Weng, Y. Zou, and D. Yu, “Diffsound: Discrete diffusion model for text-to-sound generation,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.
- [3] R. Huang, J. Huang, D. Yang, Y. Ren, L. Liu, M. Li, Z. Ye, J. Liu, X. Yin, and Z. Zhao, “Make-an-audio: Text-to-audio generation with prompt-enhanced diffusion models,” in *International Conference on Machine Learning*. PMLR, 2023, pp. 13 916–13 932.
- [4] H. Liu, Z. Chen, Y. Yuan, X. Mei, X. Liu, D. Mandic, W. Wang, and M. D. Plumbly, “Audioldm: Text-to-audio generation with latent diffusion models,” *arXiv preprint arXiv:2301.12503*, 2023.
- [5] M. Comunità, R. F. Gramaccioni, E. Postolache, E. Rodolà, D. Comminiello, and J. D. Reiss, “Syncfusion: Multimodal onset-synchronized video-to-audio foley synthesis,” in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 936–940.
- [6] F. Kreuk, G. Synnaeve, A. Polyak, U. Singer, A. Défossez, J. Copet, D. Parikh, Y. Taigman, and Y. Adi, “Audiogen: Textually guided audio generation,” *arXiv preprint arXiv:2209.15352*, 2022.
- [7] Z. Borsos, R. Marinier, D. Vincent, E. Kharitonov, O. Pietquin, M. Sharifi, D. Roblek, O. Teboul, D. Grangier, M. Tagliasacchi *et al.*, “Audioldm: a language modeling approach to audio generation,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.
- [8] G. Li, X. Xu, L. Dai, M. Wu, and K. Yu, “Diverse and vivid sound generation from text descriptions,” in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.
- [9] J. Huang, Y. Ren, R. Huang, D. Yang, Z. Ye, C. Zhang, J. Liu, X. Yin, Z. Ma, and Z. Zhao, “Make-an-audio 2: Temporal-enhanced text-to-audio generation,” *arXiv preprint arXiv:2305.18474*, 2023.
- [10] H. Chang, H. Zhang, L. Jiang, C. Liu, and W. T. Freeman, “Maskgit: Masked generative image transformer,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022*, pp. 11 315–11 325.
- [11] H. F. Garcia, P. Seetharaman, R. Kumar, and B. Pardo, “Vampnet: Music generation via masked acoustic token modeling,” *arXiv preprint arXiv:2307.04686*, 2023.
- [12] A. Ziv, I. Gat, G. L. Lan, T. Remez, F. Kreuk, A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, “Masked audio generation using a single non-autoregressive transformer,” *arXiv preprint arXiv:2401.04577*, 2024.
- [13] D. Ghosal, N. Majumder, A. Mehrish, and S. Porri, “Text-to-audio generation using instruction-tuned llm and latent diffusion model,” *arXiv preprint arXiv:2304.13731*, 2023.
- [14] H. Liu, Q. Tian, Y. Yuan, X. Liu, X. Mei, Q. Kong, Y. Wang, W. Wang, Y. Wang, and M. D. Plumbly, “Audioldm 2: Learning holistic audio generation with self-supervised pretraining,” *arXiv preprint arXiv:2308.05734*, 2023.
- [15] K. Koutini, J. Schlüter, H. Eghbal-Zadeh, and G. Widmer, “Efficient training of audio transformers with patchout,” *arXiv preprint arXiv:2110.05069*, 2021.
- [16] D. Niizumi, D. Takeuchi, Y. Ohishi, N. Harada, and K. Kashino, “Masked spectrogram modeling using masked autoencoders for learning general-purpose audio representation,” *arXiv:2204.12260*, 2022.
- [17] P.-Y. Huang, H. Xu, J. Li, A. Baevski, M. Auli, W. Galuba, F. Metze, and C. Feichtenhofer, “Masked autoencoders that listen,” *NeurIPS*, vol. 35, pp. 28 708–28 720, 2022.
- [18] Z. Zhong, H. Shi, M. Hirano, K. Shimada, K. Tateishi, T. Shibuya, S. Takahashi, and Y. Mitsufuji, “Extending audio masked autoencoders toward audio restoration,” in *IEEE WASPAA 2023*, 2023, pp. 1–5.
- [19] S.-B. Kim, S.-H. Lee, H.-Y. Choi, and S.-W. Lee, “Audio super-resolution with robust speech representation learning of masked autoencoder,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 32, pp. 1012–1022, 2024.

- [20] S. Yang, Z. Zhong, M. Zhao, S. Takahashi, M. Ishii, T. Shibuya, and Y. Mitsufuji, “Visual echoes: A simple unified transformer for audio-visual generation,” *arXiv preprint arXiv:2405.14598*, 2024.
- [21] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, “Wavenet: A generative model for raw audio,” *arXiv preprint arXiv:1609.03499*, 2016.
- [22] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, and I. Sutskever, “Jukebox: A generative model for music,” *arXiv preprint arXiv:2005.00341*, 2020.
- [23] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Défossez, “Simple and controllable music generation,” *Advances in Neural Information Processing Systems*, vol. 36, 2024.
- [24] A. Van Den Oord, O. Vinyals *et al.*, “Neural discrete representation learning,” *Advances in neural information processing systems*, vol. 30, 2017.
- [25] V. Iashin and E. Rahtu, “Taming visually guided sound generation,” *arXiv preprint arXiv:2110.08791*, 2021.
- [26] Y. Song, P. Dhariwal, M. Chen, and I. Sutskever, “Consistency models,” *arXiv preprint arXiv:2303.01469*, 2023.
- [27] D. Kim, C.-H. Lai, W.-H. Liao, N. Murata, Y. Takida, T. Uesaka, Y. He, Y. Mitsufuji, and S. Ermon, “Consistency trajectory models: Learning probability flow ode trajectory of diffusion,” *arXiv preprint arXiv:2310.02279*, 2023.
- [28] K. Saito, D. Kim, T. Shibuya, C.-H. Lai, Z. Zhong, Y. Takida, and Y. Mitsufuji, “Soundctm: Uniting score-based and consistency models for text-to-sound generation,” *arXiv preprint arXiv:2405.18503*, 2024.
- [29] J. Kong, J. Kim, and J. Bae, “Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis,” *NeurIPS*, vol. 33, pp. 17 022–17 033, 2020.
- [30] T. Shibuya, Y. Takida, and Y. Mitsufuji, “Bigvsan: Enhancing gan-based neural vocoders with slicing adversarial network,” *arXiv preprint arXiv:2309.02836*, 2023.
- [31] K. Kumar, R. Kumar, T. De Boissiere, L. Gestin, W. Z. Teoh, J. Sotelo, A. De Brebisson, Y. Bengio, and A. C. Courville, “Melgan: Generative adversarial networks for conditional waveform synthesis,” *Advances in neural information processing systems*, vol. 32, 2019.
- [32] S.-g. Lee, W. Ping, B. Ginsburg, B. Catanzaro, and S. Yoon, “Bigvgan: A universal neural vocoder with large-scale training,” *arXiv preprint arXiv:2206.04658*, 2022.
- [33] Y. Wu, K. Chen, T. Zhang, Y. Hui, T. Berg-Kirkpatrick, and S. Dubnov, “Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation,” in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.
- [34] T. Li, H. Chang, S. Mishra, H. Zhang, D. Katabi, and D. Krishnan, “Mage: Masked generative encoder to unify representation learning and image synthesis,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 2142–2152.
- [35] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville, “Film: Visual reasoning with a general conditioning layer,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.
- [36] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, “An image is worth 16x16 words: Transformers for image recognition at scale,” *ICLR 2021*, 2021.
- [37] J. Ho and T. Salimans, “Classifier-free diffusion guidance,” *arXiv preprint arXiv:2207.12598*, 2022.
- [38] H. Chang, H. Zhang, J. Barber, A. Maschinot, J. Lezama, L. Jiang, M.-H. Yang, K. Murphy, W. T. Freeman, M. Rubinstein *et al.*, “Muse: Text-to-image generation via masked generative transformers,” *arXiv preprint arXiv:2301.00704*, 2023.
- [39] V. Besnier and M. Chen, “A pytorch reproduction of masked generative image transformer,” *arXiv preprint arXiv:2310.14400*, 2023.
- [40] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, “Audio set: An ontology and human-labeled dataset for audio events,” in *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2017, pp. 776–780.
- [41] R. Wightman, “Pytorch image models,” <https://github.com/rwightman/pytorch-image-models>, 2019.
- [42] C. D. Kim, B. Kim, H. Lee, and G. Kim, “Audiocaps: Generating captions for audios in the wild,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 119–132.
- [43] “A lightweight library of frechet audio distance (fad) calculation,” <https://github.com/gudgud96/frechet-audio-distance>.
- [44] E. Moliner, M. Turunen, F. Elvander, and V. Välimäki, “A diffusion-based generative equalizer for music restoration,” *arXiv preprint arXiv:2403.18636*, 2024.

- [45] E. Law, K. West, M. I. Mandel, M. Bay, and J. S. Downie, "Evaluation of algorithms using games: The case of music tagging," in *Proceedings of the 10th International Society for Music Information Retrieval Conference, ISMIR 2009, Kobe International Conference Center, Kobe, Japan, October 26-30, 2009*, K. Hirata, G. Tzanetakis, and K. Yoshii, Eds. International Society for Music Information Retrieval, 2009, pp. 387–392. [Online]. Available: <http://ismir2009.ismir.net/proceedings/OS5-5.pdf>
- [46] Y. Li, R. Yuan, G. Zhang, Y. Ma, X. Chen, H. Yin, C. Lin, A. Ragni, E. Benetos, N. Gyenge, R. Dannenberg, R. Liu, W. Chen, G. Xia, Y. Shi, W. Huang, Y. Guo, and J. Fu, "Mert: Acoustic music understanding model with large-scale self-supervised training," 2023.
- [47] J. Spijkervet and J. A. Burgoyne, "Contrastive learning of musical representations," *arXiv preprint arXiv:2103.09410*, 2021.
- [48] J. Pons and X. Serra, "musicnn: Pre-trained convolutional neural networks for music audio tagging," *arXiv preprint arXiv:1909.06654*, 2019.
- [49] M. C. McCallum, F. Korzeniowski, S. Oramas, F. Gouyon, and A. Ehmann, "Supervised and unsupervised learning of audio representations for music understanding," in *ISMIR 2022*, 2022.
- [50] R. Castellon, C. Donahue, and P. Liang, "Codified audio language modeling learns useful representations for music information retrieval," *arXiv preprint arXiv:2107.05677*, 2021.
- [51] H. Liu, W. Choi, X. Liu, Q. Kong, Q. Tian, and D. Wang, "Neural vocoder is all you need for speech super-resolution," *arXiv preprint arXiv:2203.14941*, 2022.
- [52] H. Liu, K. Chen, Q. Tian, W. Wang, and M. D. Plumbley, "Audiosr: Versatile audio super-resolution at scale," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 1076–1080.

LONG-FORM MUSIC GENERATION WITH LATENT DIFFUSION

Zach Evans
Zack Zukowski

Julian D. Parker
Josiah Taylor

CJ Carr
Jordi Pons

Stability AI

ABSTRACT

Audio-based generative models for music have seen great strides recently, but so far have not managed to produce full-length music tracks with coherent musical structure from text prompts. We show that by training a generative model on long temporal contexts it is possible to produce long-form music of up to 4m 45s. Our model consists of a diffusion-transformer operating on a highly downsampled continuous latent representation (latent rate of 21.5 Hz). It obtains state-of-the-art generations according to metrics on audio quality and prompt alignment, and subjective tests reveal that it produces full-length music with coherent structure.

1. INTRODUCTION

Generation of musical audio using deep learning has been a very active area of research in the last decade. Initially, efforts were primarily directed towards the unconditional generation of musical audio [1, 2]. Subsequently, attention shifted towards conditioning models directly on musical metadata [3]. Recent work has focused on adding natural language control via text conditioning [4–7], and then improving these architectures in terms of computational complexity [8–11], quality [12–15] or controlability [16–19].

Existing text-conditioned models have generally been trained on relatively short segments of music, commonly of 10–30s in length [4–7] but in some cases up to 90s [14]. These segments are usually cropped from longer compositions. Although it is possible to generate longer pieces using (e.g., autoregressive [8]) models trained from short segments of music, the resulting music shows only local coherence and does not address long-term musical structure (see Table 4, MusicGen-large-stereo results). Furthermore, the analysis of a dataset of metadata from 600k popular music tracks¹ (Figure 1) confirms that the majority of songs are much longer than the lengths addressed by previous works. Therefore, if we want to produce a model

¹ www.kaggle.com/yamaerenay/spotify-tracks-dataset-19222021



© Z. Evans, J. D. Parker, CJ Carr, Z. Zukowski, J. Taylor and J. Pons. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Z. Evans, J. D. Parker, CJ Carr, Z. Zukowski, J. Taylor and J. Pons, “Long-form music generation with latent diffusion”, in *Proc. of the 25th Int. Society for Music Information Retrieval Conf.*, San Francisco, United States, 2024.

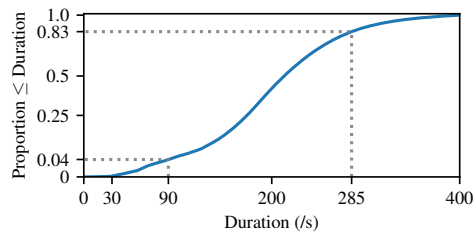


Figure 1: Cumulative histogram showing the proportion of music that is less than a particular length, for a representative sample of popular music¹. Dotted lines: proportion associated with the max generation length of our model (285s) and of previous models (90s). The vertical axis is warped with a power law for greater readability.

that can understand and produce natural musical structure, it is likely necessary to train and generate on a longer time window. We identify 285s (4m 45s) as a target length, as it is short enough to be within reach of current deep learning architectures, can fit into the VRAM of modern GPUs, and covers a high percentage of popular music.

In previous works [4, 20] it has been hypothesized that “semantic tokens enable long-term structural coherence, while modeling the acoustic tokens conditioned on the semantic tokens enables high-quality audio synthesis” [20]. Semantic tokens are time-varying embeddings derived from text embeddings, aiming to capture the overall characteristics and evolution of music at a high level. This intermediate representation is practical because it operates at low temporal resolution. Semantic tokens are then employed to predict acoustic embeddings, which are later utilized for waveform reconstruction.² Semantic tokens are commonly used in autoregressive modeling to provide guidance on what and when to stop generating [4, 20].

Another line of work [14] implicitly assumes that conditioning on semantic tokens is unnecessary for long-form music structure to emerge. Instead, it assumes that structure can emerge by training end-to-end without semantic tokens. This involves generating the entire music piece at once (full-context generation), rather than generating audio autoregressively guided by semantic tokens [4, 20]. This approach has the potential to simplify the pipeline from four stages² to three (text→text-embedding→acoustic-token→waveform) or even one (text→waveform). While the single-stage approach represents the closest approxi-

² [4, 20] are typically conformed by four stages (denoted here as →): text→text-embedding→semantic-token→acoustic-token→waveform.

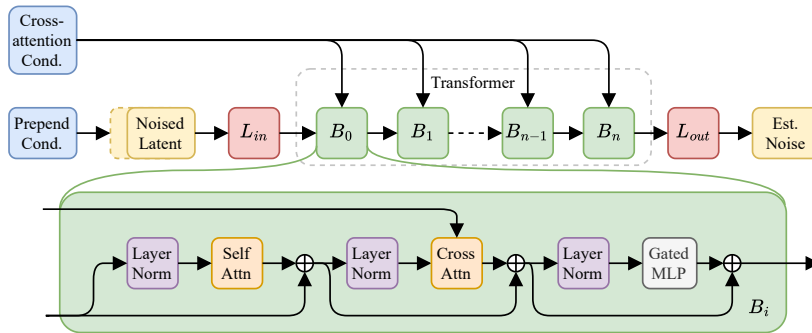


Figure 2: Architecture of the diffusion-transformer (DiT). Cross-attention includes timing and text conditioning. Prepend conditioning includes timing conditioning and also the signal conditioning on the current timestep of the diffusion process.

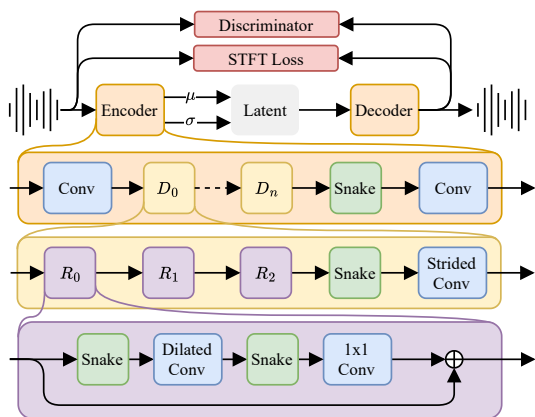


Figure 3: Architecture of the autoencoder.

mation to end-to-end learning, its may be challenging to implement due to the VRAM limitations of current GPUs. Our model consists of three stages able to generate an entire music piece of 4m 45s at once without semantic tokens.

Most music generation works rely on autoencoders to condense the long waveforms into compact latent representations (acoustic tokens or embeddings). Prominent examples utilize residual-vector-quantizers to provide discrete acoustic tokens [21–23] for autoregressive or masked token modeling [8–10, 16]. Another prominent line of work focuses on variational autoencoders to provide a continuous and normalized acoustic embedding [5, 7, 12, 14] for latent diffusion modelling. Our work relies on latent diffusion modeling to generate music from text prompts. Yet, and differently from prior works operating with latent rates of 40Hz to 150Hz [14, 23, 24], our autoencoder relies on a highly downsampled latent operating at 21.5Hz (Table 5). We argue that maintaining perceptual quality at low latent rates can be essential for training generative models on long temporal contexts, enabling the creation of long-form music without the need to rely on semantic tokens.

In our work we scale a generative model to operate over the 285s (4m 45s) time interval. This is achieved by using a highly compressed continuous latent, and a generative model relying on latent diffusion (Sections 2 and 3). The resulting model obtains state-of-the-art results in terms of audio quality and text-prompt coherence (Section 4.1), and is also capable of generating long-form music with coherent structure (Sections 4.2 and 4.4) in 13s on a GPU.

Code to reproduce our model ³ and demos ⁴ are online.

2. LATENT DIFFUSION ARCHITECTURE

Our model generates variable-length (up to 4m 45s) stereo music at 44.1kHz from text prompts. It comprises three main components: an autoencoder that compresses waveforms into a manageable sequence length, a contrastive text-audio embedding model based on CLAP [25, 26] for text conditioning, and a transformer-based diffusion model that operates in the latent space of the autoencoder. Check their exact parametrizations online in our code repository.³

2.1 Autoencoder

We employ an autoencoder structure that operates on raw waveforms (Figure 3). The encoder section processes these waveforms by a series of convolutional blocks, each of which performs downsampling and channel expansion via strided convolutions. Before each downsampling block, we employ a series of ResNet-like layers using dilated convolutions and Snake [27] activation functions for further processing. All convolutions are parameterized in a weight-normalised form. The decoder is almost identical to the encoder structure, but employs transposed strided convolutions for upsampling and channel contraction at the start of each upsampling block. The encoder and decoder structures are similar to that of DAC [23], but with the addition of a trainable β parameter in the Snake activation, which controls the magnitude of the periodicity in the activation. We also remove the $\tanh()$ activation used in DAC at the output of the decoder, as we found it introduced harmonic distortion into the signal. The bottleneck of the autoencoder is parameterized as a variational autoencoder.

We train it using a variety of objectives. First, the reconstruction loss, consisting of a perceptually weighted multi-resolution STFT [28] that deals with stereo audio as follows: the STFT loss is applied to the mid-side (M/S) representation of the stereo audio, as well as the left and right (L/R) channels separately. The L/R component is weighted by 0.5 compared to the M/S one, and exists to mitigate potential ambiguity around L/R placement. Second, an adversarial loss term with feature matching, utilizing 5 con-

³ <https://github.com/Stability-AI/stable-audio-tools/>

⁴ <https://stability-ai.github.io/stable-audio-2-demo/>

	DiT	AE	CLAP	Total
Parameters	1.1B	157M	125M	1.3B

Table 1: Number of learnable parameters of our models.

volitional discriminators [22] with hyperparameters consistent with previous work [14], but with channel count scaled to give ≈ 4 times the parameter count. And third, the KL divergence loss term that is weighted by $\times 10^{-4}$.

2.2 Diffusion-transformer (DiT)

Instead of the widely used convolutional U-Net structure [5–7, 12], we employ a diffusion-transformer (DiT). This approach has seen notable success in other modalities [29], and has recently been applied to musical audio [30]. The used transformer (Figure 2) follows a standard structure with stacked blocks consisting of serially connected attention layers and gated multi-layer perceptrons (MLPs), with skip connections around each. We employ layer normalization at the input to both the attention layer and the MLP. The key and query inputs to the attention layer have rotary positional embedding [31] applied to the lower half of the embedding. Each transformer block also contains a cross-attention layer to incorporate conditioning. Linear mappings are used at the input and output of the transformer to translate from the autoencoder latent dimension to the embedding dimension of the transformer. We utilize efficient block-wise attention [32] and gradient checkpointing [33] to reduce the computational and memory impact of applying a transformer architecture over longer sequences. These techniques are crucial to viable training of model with this context length.

The DiT is conditioned by 3 signals: *text* enabling natural language control, *timing* enabling variable-length generation, and *timestep* signaling the current timestep of the diffusion process. Text CLAP embeddings are included via cross-attention. Timing conditioning [3, 14] is calculated using sinusoidal embeddings [34] and also included via cross-attention. Timing conditioning is also prepended before the transformer, along with a sinusoidal embedding describing the current timestep of the diffusion process.

2.3 Variable-length music generation

Given that the nature of long-form music entails varying lengths, our model also allows for variable-length music generation. We achieve this by generating content within a specified window length (e.g., 3m 10s or 4m 45s) and relying on the timing condition to fill the signal up to the length specified by the user. The model is trained to fill the rest of the signal with silence. To present variable-length audio outputs shorter than the window length to end-users, one can easily trim the appended silence. We adopt this strategy, as it has shown its effectiveness in previous work [14].

2.4 CLAP text encoder

We rely on a contrastive model trained from text-audio pairs, following the structure of CLAP [26]. It consists

of a HTSAT-based [35] audio encoder with fusion and a RoBERTa-based [36] text encoder, both trained from scratch on our dataset with a language-audio contrastive loss. Following previous work [14], we use as text features the next-to-last hidden layer of the CLAP text encoder.

3. TRAINING SETUP

Training the model is a multi-stage process and was conducted on a cluster of NVIDIA A100 GPUs. Firstly, the autoencoder and CLAP model are trained. The CLAP model required approximately 3k GPU hours⁵ and the autoencoder 16k GPU hours⁴. Secondly, the diffusion model is trained. To reach our target length of 4m 45s, we first pre-train the model for 70k GPU hours⁴ on sequences corresponding to a maximum of 3m 10s of music. We then take the resulting model and fine-tune it on sequences of up to 4m 45s for a further 15k GPU hours⁴. Hence, the diffusion model is first pre-trained to generate 3m 10s music (referred to as the *pre-trained* model), and then fine-tuned to generate 4m 45s music (the *fully-trained* model).

All models are trained with the AdamW optimiser, with a base learning rate of $1e - 5$ and a scheduler including exponential ramp-up and decay. We maintain an exponential moving average of the weights for improved inference. Weight decay, with a coefficient of 0.001, is also used. Parameter counts for the networks are given in Table 1, and the exact hyperparameters we used are detailed online³.

The DiT is trained to predict a noise increment from noised ground-truth latents, following the v-objective [37]. We sample from our model using DPM-Solver++ [38] (100 steps), with classifier-free guidance [39] (scale of 7.0).

3.1 Training data and prompt preparation

Our dataset consists of 806,284 files (19,500h) containing music (66% or 94%)⁶, sound effects (25% or 5%)⁵, and instrument stems (9% or 1%)⁵. This audio is paired with text metadata that includes natural-language descriptions of the audio file’s contents, as well as metadata such as BPM, genre, moods, and instruments for music tracks. All of our dataset (audio and metadata) is available online⁷ for consultation. This data is used to train all three components of the system from scratch: the CLAP text encoder, the autoencoder and the DiT. The 285s (4m 45s) target temporal context encompasses over 90% of the dataset.

During the training of the CLAP text encoder and the DiT, we generate text prompts from the metadata by concatenating a random subset of the metadata as a string. This allows for specific properties to be specified during inference, while not requiring these properties to be present at all times. For half of the samples, we include the metadata-type (e.g., Instruments or Moods) and join them with a delimiting character (e.g., Instruments: Guitar, Drums, Bass Guitar|Moods: Uplifting, Energetic). For the other half, we do not include the metadata-type and join the

⁵ GPU hours represent one hour of computation on a single GPU. The training process was distributed across multiple GPUs for efficiency.

⁶ Percentages: number of files or GBs of content, respectively.

⁷ <https://www.audiosparx.com/>

properties with a comma (e.g., Guitar, Drums, Bass Guitar, Uplifting, Energetic). For metadata-types with a list of values, we shuffle the list. Hence, we perform a variety of random transformations of the resulting string, including two variants of delimiting character (“;” and “|”), shuffling orders and transforming between upper and lower case.

4. EXPERIMENTS

4.1 Quantitative evaluation

We evaluate a corpus of generated music using previously established metrics [14], as implemented in *stable-audio-metrics*.⁸ Those include the Fréchet distance on OpenL3 embeddings [40], KL-divergence on PaSST tags [41], and distance in LAION-CLAP space [26, 42]⁹.

We set MusicGen-large-stereo (MusicGen) [8] as baseline, since it is the only publicly available model able to generate music at this length in stereo. This autoregressive model can generate long-form music of variable length due to its sequential (one-sample-at-a-time generation) sampling. However, note that MusicGen is not conditioned on semantic tokens that ensure long-term structural coherence, and it was not trained to generate such long contexts.

The prompts and ground-truth audio used for the quantitative study are from the Song Descriptor Dataset [43]. We select this benchmark, with 2m long music, because other benchmarks contain shorter music segments [4] and are inappropriate for long-form music evaluation. As vocal generation is not our focus and MusicGen is not trained for this task either, we opted to ensure a fair evaluation against MusicGen by curating a subset of 586 prompts that exclude vocals.¹⁰ This subset, referred to as the Song Descriptor Dataset (no-singing), serves as our benchmark for comparison. We assess 2m generations to remain consistent with the ground-truth and also evaluate our models at their maximum generation length—which is 3m 10s for the pre-trained model or 4m 45s for the fully-trained one (Tables 2 and 3, respectively). For each model and length we study, we generate one render per prompt in our benchmark. This results in 586 generations per experiment.

Our model is first pre-trained to generate 3m 10s music (pre-trained model) and then fine-tuned to generate 4m 45s music (fully-trained model). Tables 2 and 3 show the quantitative results for both models and inference times. Comparing metrics between the pre-trained model and the fully-trained one shows no degradation, confirming the viability of extending context length via this mechanism. The proposed model scores better than MusicGen at all lengths while being significantly faster.

4.2 Qualitative evaluation

We evaluate the corpus of generated music qualitatively, with a listening test developed with webMUSHRA [44].

Mixed in with our generated music are generations from MusicGen and also ground-truth samples from the Song Descriptor Dataset (no-singing). Generations of our fully-trained model are included at both 4m 45s and 2m long, whilst ground-truth is only available at 2m. We selected two samples from each use case that were competitive for both models. For MusicGen it was difficult to find coherently structured music, possibly because it is not trained for long-form music generation. For our model, we found some outstanding generations that we selected for the test. Test material is available on our demo page.

Test subjects were asked to rate examples on a number of qualities including audio quality, text alignment, musical structure, musicality, and stereo correctness. We report mean opinion scores (MOS) in the following scale: *bad* (1), *poor* (2), *fair* (3), *good* (4), *excellent* (5). We observed that assessing stereo correctness posed a significant challenge for many users. To address this, we streamlined the evaluation by seeking for a binary response, correct or not, and report percentages of stereo correctness. All 26 test subjects used studio monitors or headphones, and self-identified as music producers or music researchers. In order to reduce test time and maximise subject engagement, we split the test into two parts. Each participant can choose one of the parts, or both, depending on their available time.

Results in Table 4 indicate that the generations from our system are comparable to the ground-truth in most aspects, and superior to the existing baseline. Our model obtains *good* (4) MOS scores across the board and stereo correctness scores higher than 95%, except for 2m long generations where its musical structure is *fair* (3). Differently from our quantitative results in Table 3, qualitative metrics show that 2m long generations are slightly worse than the 4m 45s generations (specially musical structure). We hypothesize that this could be due to the relative scarcity of full-structured music at this length in our dataset, since most music at this length might be repetitive loops. These results confirm that semantic tokens are not strictly essential for generating music with structure, as it can emerge through training with long contexts. Note, however, that the temporal context must be sufficiently long to obtain structured music generation. It was not until we scaled to longer temporal contexts (4m 45s), that we observed music with *good* structure, reflecting the inherent nature of the data. It is also noteworthy that the perceptual evaluation of structure yields to a wide diversity of responses, as indicated by the high standard deviations in Table 4. This highlights the challenge of evaluating subjective musical aspects. Finally, MusicGen achieves a stereo correctness rate of approximately 60%. This may be attributed to its tendency to generate mixes where instruments typically panned in the center (such as bass or kick) are instead panned to one side, creating an unnaturally wide mix that was identified as incorrect by the music producers and researchers participating in our test.

⁸ <https://github.com/Stability-AI/stable-audio-metrics>

⁹ <https://github.com/LAION-AI/CLAP>

¹⁰ Prompts containing any of those words were removed: speech, speech synthesizer, hubbub, babble, singing, male, man, female, woman, child, kid, synthetic singing, choir, chant, mantra, rapping, humming, groan, grunt, vocal, vocalist, singer, voice, and acapella.

	channels/sr	length	FD _{openl3} ↓	KL _{passt} ↓	CLAP _{score} ↑	inference time
MusicGen-large-stereo [8]	2/32kHz	2m	204.03	0.49	0.28	6m 38s
Ours (pre-trained)	2/44.1kHz	2m [†]	78.70	0.36	0.39	8s
MusicGen-large-stereo [8]	2/32kHz	3m 10s	213.76	0.50	0.28	9m 32s
Ours (pre-trained)	2/44.1kHz	3m 10s	89.33	0.34	0.39	8s

Table 2: *Song Describer Dataset (no-singing subset)*: results of the 3m 10s pre-trained model. [†]Our pre-trained model generates 3m 10s outputs, but during inference it can generate 2m outputs by relying on the timing conditioning. We trim audios to 2m (discarding the end silent part) for a fair quantitative evaluation against the state-of-the-art (see Section 2.3).

	channels/sr	output length	FD _{openl3} ↓	KL _{passt} ↓	CLAP _{score} ↑	inference time
MusicGen-large-stereo [8]	2/32kHz	2m	204.03	0.49	0.28	6m 38s
Ours (fully-trained)	2/44.1kHz	2m [†]	79.09	0.35	0.40	13s
MusicGen-large-stereo [8]	2/32kHz	4m 45s	218.02	0.50	0.27	12m 53s
Ours (fully-trained)	2/44.1kHz	4m 45s	81.96	0.34	0.39	13s

Table 3: *Song Describer Dataset (no-singing subset)*: results of the 4m 45s fully-trained model. [†]Our fully-trained model generates 4m 45s outputs, but during inference it can generate 2m outputs by relying on the timing conditioning. We trim audios to 2m (discarding the end silent part) for a fair quantitative evaluation against the state-of-the-art (see Section 2.3).

Results with the fully-trained model:	2m long			4m 45s long	
	Stable Audio 2	MusicGen-large-stereo	ground truth	Stable Audio 2	MusicGen-large-stereo
Audio Quality	4.0±0.6	2.8±0.8	4.6±0.4	4.5±0.4	2.8±0.8
Text Alignment	4.3±0.7	3.1±0.8	4.6±0.5	4.6±0.4	2.9±1.0
Structure	3.5±1.3	2.4±0.7	4.3±0.8	4.0±1.0	2.1±0.7
Musicality	4.0±0.8	2.7±0.9	4.6±0.5	4.3±0.7	2.6±0.7
Stereo correctness	96%	61%	96%	100%	57%

Table 4: *Qualitative results*. Top: mean opinion score ± standard deviation. Bottom: percentages.

	sampling rate	STFT distance ↓	MEL distance ↓	SI-SDR ↑	latent rate	latent (channels)
DAC [23]	44.1kHz	0.96	0.52	10.83	86 Hz	discrete
AudioGen [24]	48kHz	1.17	0.64	9.27	50 Hz	discrete
Encodec [8, 22]	32kHz	1.82	1.12	5.33	50 Hz	discrete
AudioGen [24]	48kHz	1.10	0.64	8.82	100 Hz	continuous (32)
Stable Audio [14]	44.1kHz	1.19	0.67	8.62	43 Hz	continuous (64)
Ours	44.1kHz	1.19	0.71	7.14	21.5 Hz	continuous (64)

Table 5: *Autoencoder reconstructions on the Song Describer Dataset (all the dataset)*. Although different autoencoders operate at various sampling rates, the evaluations are run at 44.1kHz bandwidth for a fair comparison. Sorted by latent rate.

4.3 Autoencoder evaluation

We evaluate the audio reconstruction quality of our autoencoder in isolation, as it provides a quality ceiling for our system. We achieve this by comparing ground-truth and reconstructed audio via a number of established audio quality metrics [22, 23]: STFT distance, MEL distance and SI-SDR (as in AuraLoss library [28], with its default parameters). The reconstructed audio is obtained by encoding-decoding the ground-truth audio from the Song Describer Dataset (all the dataset, 706 tracks) through the autoencoder. As a comparison, we calculate the same metrics on a number of publicly available neural audio codecs in-

cluding Encodec [22], DAC [23] and AudioGen [24]. Encodec and DAC are selected because are widely used for generative music modeling [4, 8, 10]. We select the Encodec 32kHz variant because our MusicGen baseline relies on it, and DAC 44.1kHz because its alternatives operate at 24kHz and 16kHz. Further, since our autoencoder relies on a continuous latent, we also compare against AudioGen, a state-of-the-art autoencoder with a continuous latent. Notably, AudioGen presents both continuous and discrete options, and we report both for completeness. All neural audio codecs are stereo, except DAC 44.1kHz and Encodec 32 kHz. In those cases, we independently project left and right channels and reconstruct from those.

The results in Table 5 show that the proposed autoencoder is comparable or marginally worse in raw reconstruction quality with respect to the other available baselines, whilst targeting a significantly larger amount (2x-5x) of temporal downsampling, and hence a lower latent rate. Our results are not strictly comparable against discrete neural audio codecs, but are included as a reference. For a qualitative assessment of our autoencoder’s reconstruction quality, listen to some examples on our demo site.

4.4 Musical structure analysis

We explore the plausibility of the generated structures by visualizing the binary self-similarity matrices (SSMs) [45] of randomly chosen generated music against real music of the same genre. Real music is from the Free Music Archive (FMA) [46]. Similarly to real music, our model’s generations can build structure with intricate shifts, including repetition of motives that were introduced at the first section. Red marks in Figure 4 show late sections that are similar to early sections. In MusicGen examples, early sections rarely repeat (e.g., see diagonal lines in Figure 4c) or music gets stuck in a middle/ending section loop (repetitive/loop sections are marked in blue in Figure 4). Note that our model’s middle sections can also be repetitive, while still maintaining an intro/outro. We omit MusicGen’s second row because most of its SSMs exhibit a similar behaviour.

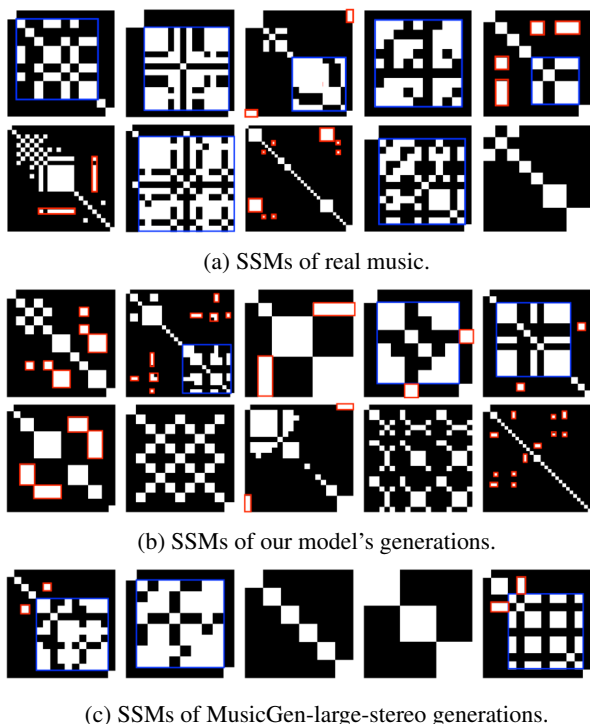


Figure 4: Each column shows the SSMs of different genres (left to right): rock, pop, jazz, hip-hop, and classical.

4.5 Memorization analysis

Recent works [47, 48] examined the potential of generative models to memorize training data, especially for repeated elements in the training set. Further, musicLM [4] conducted a memorization analysis to address concerns on the

potential misappropriation of creative content. Adhering to principles of responsible model development, we also run a comprehensive study on memorization [4, 47, 48].

Considering the increased probability of memorizing repeated music within the dataset, we start by studying if our training set contains repeated data. We embed all our training data using the LAION-CLAP⁸ audio encoder to select audios that are close in this space based on a manually set threshold. The threshold is set such that the selected audios correspond to exact replicas. With this process, we identify 5566 repeated audios in our training set.

We compare our model’s generations against the training set in LAION-CLAP⁸ space. Generations are from 5566 prompts within the repeated training data (in-distribution), and 586 prompts from the Song Descriptor Dataset (no-singing, out-of-distribution). We then identify the top-50 generated music that is closest to the training data and listen. We extensively listened to potential memorization candidates, and could not find memorization. We even selected additional outstanding generations, and could not find memorization. The most interesting memorization candidates, together with their closest training data, are online for listening on our demo page.

4.6 Additional creative capabilities

Besides text-conditioned long-form music generation, our model exhibits capabilities in other applications. While we do not conduct a thorough evaluation of these, we briefly describe those and showcase examples on our demo page.

Audio-to-audio — With diffusion models is possible to perform some degree of style-transfer by initializing the noise with audio during sampling [15, 49]. This capability can be used to modify the aesthetics of an existing recording based on a given text prompt, whilst maintaining the reference audio’s structure (e.g., a beatbox recording could be style-transferred to produce realistic-sounding drums). As a result, our model can be influenced by not only text prompts but also audio inputs, enhancing its controllability and expressiveness. We noted that when initialized with voice recordings (such as beatbox or onomatopoeias), there is a sensation of control akin to an instrument. Examples of audio-to-audio are on our demo page.

Vocal music — The training dataset contains a subset of music with vocals. Our focus is on the generation of instrumental music, so we do not provide any conditioning based on lyrics. As a result, when the model is prompted for vocals, the model’s generations contains vocal-like melodies without intelligible words. Whilst not a substitute for intelligible vocals, these sounds have an artistic and textural value of their own. Examples are given on our demo page.

Short-form audio generation — The training set does not exclusively contain long-form music. It also contains shorter sounds like sound effects or instrument samples. As a consequence, our model is also capable of producing such sounds when prompted appropriately. Examples of short-form audio generations are also on our demo page.

5. CONCLUSIONS

We presented an approach to building a text-conditioned music generation model, operating at long enough context lengths to encompass full musical tracks. To achieve this we train an autoencoder which compresses significantly more in the temporal dimension than previous work. We model full musical tracks represented in the latent space of this autoencoder via a diffusion approach, utilizing a diffusion-transformer. We evaluate the trained model via qualitative and quantitative tests, and show that it is able to produce coherent music with state-of-the-art results over the target temporal context of 4m45s.

6. ETHICS STATEMENT

Our technology represents an advancement towards aiding humans in music production tasks, facilitating the creation of variable-length, long-form stereo music based on textual input. This advancement greatly enhances the creative repertoire available to artists and content creators. However, despite its numerous advantages, it also brings inherent risks. A key concern lies in the potential reflection of biases inherent in the training data. Additionally, the nuanced context embedded within music emphasizes the necessity for careful consideration and collaboration with stakeholders. In light of these concerns, we are dedicated to ongoing research and collaboration with those stakeholders, including artists and data providers, to navigate this new terrain responsibly. Adhering to best practices in responsible model development, we conducted an exhaustive study on memorization. Employing our methodology, we found no instances of memorization.

7. REFERENCES

- [1] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, “Wavenet: A generative model for raw audio,” *arXiv*, 2016.
- [2] S. Mehri, K. Kumar, I. Gulrajani, R. Kumar, S. Jain, J. Sotelo, A. Courville, and Y. Bengio, “SampleRNN: An unconditional end-to-end neural audio generation model,” *arXiv*, 2016.
- [3] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, and I. Sutskever, “Jukebox: A generative model for music,” *arXiv*, 2020.
- [4] A. Agostinelli, T. I. Denk, Z. Borsos, J. Engel, M. Verzetti, A. Caillon, Q. Huang, A. Jansen, A. Roberts, M. Tagliasacchi, M. Sharifi, N. Zeghidour, and C. Frank, “MusicLM: Generating music from text,” *arXiv*, 2023.
- [5] F. Schneider, Z. Jin, and B. Schölkopf, “Moûsai: Text-to-music generation with long-context latent diffusion,” *arXiv*, 2023.
- [6] Q. Huang, D. S. Park, T. Wang, T. I. Denk, A. Ly, N. Chen, Z. Zhang, Z. Zhang, J. Yu, C. Frank, J. Engel, Q. V. Le, W. Chan, Z. Chen, and W. Han, “Noise2music: Text-conditioned music generation with diffusion models,” *arXiv*, 2023.
- [7] H. Liu, Z. Chen, Y. Yuan, X. Mei, X. Liu, D. Mandic, W. Wang, and M. D. Plumbley, “AudioLDM: Text-to-audio generation with latent diffusion models,” *arXiv*, 2023.
- [8] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Défossez, “Simple and controllable music generation,” *arXiv*, 2023.
- [9] H. F. Garcia, P. Seetharaman, R. Kumar, and B. Pardo, “VampNet: Music generation via masked acoustic token modeling,” *arXiv*, 2023.
- [10] A. Ziv, I. Gat, G. L. Lan, T. Remez, F. Kreuk, A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, “Masked audio generation using a single non-autoregressive transformer,” *arXiv*, 2024.
- [11] M. W. Y. Lam, Q. Tian, T. Li, Z. Yin, S. Feng, M. Tu, Y. Ji, R. Xia, M. Ma, X. Song, J. Chen, Y. Wang, and Y. Wang, “Efficient neural music generation,” *arXiv*, 2023.
- [12] H. Liu, Q. Tian, Y. Yuan, X. Liu, X. Mei, Q. Kong, Y. Wang, W. Wang, Y. Wang, and M. D. Plumbley, “AudioLDM 2: Learning holistic audio generation with self-supervised pretraining,” *arXiv*, 2023.
- [13] G. Cideron, S. Girgin, M. Verzetti, D. Vincent, M. Kastelic, Z. Borsos, B. McWilliams, V. Ungureanu, O. Bachem, O. Pietquin, M. Geist, L. Hussenot, N. Zeghidour, and A. Agostinelli, “MusicRL: Aligning music generation to human preferences,” *arXiv*, 2024.
- [14] Z. Evans, C. Carr, J. Taylor, S. Hawley, and J. Pons, “Fast timing-conditioned latent audio diffusion,” *arXiv*, 2024.
- [15] S. Pascual, G. Bhattacharya, C. Yeh, J. Pons, and J. Serrà, “Full-band general audio synthesis with score-based diffusion,” *IEEE Int. Conf. on Acoustics, Speech and Sig. Proc. (ICASSP)*, 2023.
- [16] J. Parker, J. Spijkervet, K. Kosta, F. Yesiler, B. Kuznetsov, J.-C. Wang, M. Avent, J. Chen, and D. Le, “StemGen: A music generation model that listens,” *IEEE Int. Conf. on Acoustics, Speech and Sig. Proc. (ICASSP)*, 2024.
- [17] Z. Novack, J. McAuley, T. Berg-Kirkpatrick, and N. J. Bryan, “Ditto: Diffusion inference-time t-optimization for music generation,” *arXiv*, 2024.
- [18] G. Mariani, I. Tallini, E. Postolache, M. Mancusi, L. Cosmo, and E. Rodolà, “Multi-source diffusion models for simultaneous music generation and separation,” *arXiv*, 2023.

- [19] M. Pasini, M. Grachten, and S. Lattner, “Bass accompaniment generation via latent diffusion,” *arXiv*, 2024.
- [20] Z. Borsos, R. Marinier, D. Vincent, E. Kharitonov, O. Pietquin, M. Sharifi, D. Roblek, O. Teboul, D. Grangier, M. Tagliasacchi *et al.*, “AudioLM: a language modeling approach to audio generation,” *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, 2023.
- [21] N. Zeghidour, A. Luebs, A. Omran, J. Skoglund, and M. Tagliasacchi, “Soundstream: An end-to-end neural audio codec,” *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, 2021.
- [22] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, “High fidelity neural audio compression,” *arXiv*, 2022.
- [23] R. Kumar, P. Seetharaman, A. Luebs, I. Kumar, and K. Kumar, “High-fidelity audio compression with improved rvqgan,” *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.
- [24] AudiogenAI, “Audiogenai/agc: Audiogen codec.” [Online]. Available: <https://github.com/AudiogenAI/agc>
- [25] B. Elizalde, S. Deshmukh, M. A. Ismail, and H. Wang, “Clap: Learning audio concepts from natural language supervision,” *arXiv*, 2022.
- [26] Y. Wu, K. Chen, T. Zhang, Y. Hui, T. Berg-Kirkpatrick, and S. Dubnov, “Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation,” *IEEE Int. Conf. on Acoustics, Speech and Sig. Proc. (ICASSP)*, 2023.
- [27] L. Ziyin, T. Hartwig, and M. Ueda, “Neural networks fail to learn periodic functions and how to fix it,” *arXiv*, 2020.
- [28] C. J. Steinmetz and J. D. Reiss, “auraloss: Audio focused loss functions in PyTorch,” in *Digital Music Research Network One-day Workshop (DMRN+15)*, 2020.
- [29] W. Peebles and S. Xie, “Scalable diffusion models with transformers,” in *Proc. of the IEEE/CVF Int. Conf. on Comp. Vision (ICCV)*, 2023.
- [30] M. Levy, B. Di Giorgi, F. Weers, A. Katharopoulos, and T. Nickson, “Controllable music production with diffusion models and guidance gradients,” *arXiv*, 2023.
- [31] J. Su, Y. Lu, S. Pan, A. Murtadha, B. Wen, and Y. Liu, “Roformer: Enhanced transformer with rotary position embedding,” *arXiv*, 2023.
- [32] T. Dao, D. Y. Fu, S. Ermon, A. Rudra, and C. Ré, “FlashAttention: Fast and memory-efficient exact attention with IO-awareness,” *arXiv*, 2022.
- [33] T. Chen, B. Xu, C. Zhang, and C. Guestrin, “Training deep nets with sublinear memory cost,” *arXiv*, 2016.
- [34] J. Ho, A. Jain, and P. Abbeel, “Denoising diffusion probabilistic models,” *arXiv*, 2020.
- [35] K. Chen, X. Du, B. Zhu, Z. Ma, T. Berg-Kirkpatrick, and S. Dubnov, “HTS-AT: A hierarchical token-semantic audio transformer for sound classification and detection,” in *IEEE Int. Conf. on Acoustics, Speech and Sig. Proc. (ICASSP)*, 2022.
- [36] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, “RoBERTa: A robustly optimized BERT pretraining approach,” *arXiv*, 2019.
- [37] T. Salimans and J. Ho, “Progressive distillation for fast sampling of diffusion models,” *arXiv*, 2022.
- [38] C. Lu, Y. Zhou, F. Bao, J. Chen, C. Li, and J. Zhu, “DPM-solver++: Fast solver for guided sampling of diffusion probabilistic models,” *arXiv*, 2022.
- [39] S. Lin, B. Liu, J. Li, and X. Yang, “Common diffusion noise schedules and sample steps are flawed,” *IEEE/CVF Winter Conf. on Applications of Comp. Vision*, 2024.
- [40] A. L. Cramer, H.-H. Wu, J. Salamon, and J. P. Bello, “Look, listen, and learn more: Design choices for deep audio embeddings,” *IEEE Int. Conf. on Acoustics, Speech and Sig. Proc. (ICASSP)*, 2019.
- [41] K. Koutini, J. Schlüter, H. Eghbal-zadeh, and G. Widmer, “Efficient training of audio transformers with patchout,” *Conf. of the Int. Speech Comm. Assoc. (INTERSPEECH)*, 2022.
- [42] R. Huang, J. Huang, D. Yang, Y. Ren, L. Liu, M. Li, Z. Ye, J. Liu, X. Yin, and Z. Zhao, “Make-an-audio: Text-to-audio generation with prompt-enhanced diffusion models,” *arXiv*, 2023.
- [43] I. Manco, B. Weck, S. Doh, M. Won, Y. Zhang, D. Bogdanov, Y. Wu, K. Chen, P. Tovstogan, E. Benetos, E. Quinton, G. Fazekas, and J. Nam, “The Song Descriptor Dataset: a corpus of audio captions for music-and-language evaluation,” *arXiv*, 2023.
- [44] M. Schoeffler, S. Bartoschek, F.-R. Stöter, M. Roess, S. Westphal, B. Edler, and J. Herre, “web-MUSHRA—a comprehensive framework for web-based listening tests,” *Journal of Open Research Software*, 2018.
- [45] J. Serra, M. Müller, P. Grosche, and J. L. Arcos, “Unsupervised music structure annotation by time series structure features and segment similarity,” *IEEE Trans. on Multimedia*, 2014.
- [46] M. Defferrard, K. Benzi, P. Vandergheynst, and X. Bresson, “Fma: A dataset for music analysis,” *arXiv*, 2016.

- [47] N. Carlini, J. Hayes, M. Nasr, M. Jagielski, V. Shwag, F. Tramer, B. Balle, D. Ippolito, and E. Wallace, “Extracting training data from diffusion models,” in *USENIX Security Symposium*, 2023.
- [48] P. Esser, S. Kulal, A. Blattmann, R. Entezari, J. Müller, H. Saini, Y. Levi, D. Lorenz, A. Sauer, F. Boesel *et al.*, “Scaling rectified flow transformers for high-resolution image synthesis,” *arXiv*, 2024.
- [49] S. Rouard and G. Hadjeres, “CRASH: Raw audio score-based generative modeling for controllable high-resolution drum sound synthesis,” *arXiv*, 2021.

COMPOSER’S ASSISTANT 2: INTERACTIVE MULTI-TRACK MIDI INFILLING WITH FINE-GRAINED USER CONTROL

Martin E. Malandro
Sam Houston State University
malandro@shsu.edu

ABSTRACT

We introduce Composer’s Assistant 2, a system for interactive human-computer composition in the REAPER digital audio workstation. Our work upgrades the Composer’s Assistant system (which performs multi-track infilling of symbolic music at the track-measure level) with a wide range of new controls to give users fine-grained control over the system’s outputs. Controls introduced in this work include two types of rhythmic conditioning controls, horizontal and vertical note onset density controls, several types of pitch controls, and a rhythmic interest control. We train a T5-like transformer model to implement these controls and to serve as the backbone of our system. With these controls, we achieve a dramatic improvement in objective metrics over the original system. We also study how well our model understands the meaning of our controls, and we conduct a listening study that does not find a significant difference between real music and music composed in a co-creative fashion with our system. We release our complete system, consisting of source code, pretrained models, and REAPER scripts.

1. INTRODUCTION

Composers using generative systems to help them create music desire the ability to steer the systems towards outputs reflective of their style and intent [1]. A study of the challenges that composers faced in the 2020 AI Song Writing Contest found that the systems used in that contest were not easily steerable, and called for new systems and interfaces that are more decomposable, steerable, and adaptive [2]. A 2023 user study of the MMM [3,4] multi-track MIDI infilling model, integrated into a digital audio workstation (DAW) with an interface containing only a temperature parameter, found that users desired additional steering control over the outputs of the model [5]. Another multi-track MIDI infilling model, Composer’s Assistant [6], was adopted by a team of composers to help recreate the lost music of the opera *Andromeda* [7]. Those composers expressed difficulty using the model to create melodies that fit the lyrics they already had, since lyrics

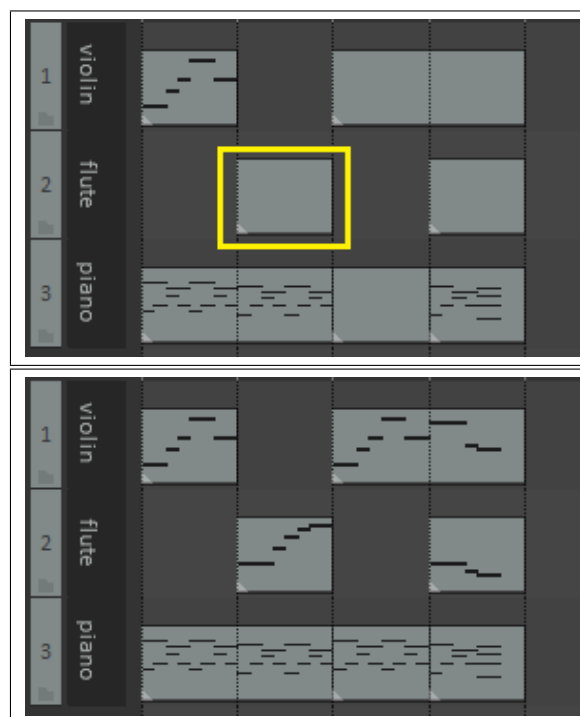


Figure 1. A 4-measure prompt in REAPER, followed by a model output. Users place empty MIDI items in REAPER to tell the model in which measures to write notes, and track names to tell the model what instrument is on each track. A track-measure in the prompt is boxed.

tend to have a natural rhythm to them and that model does not offer rhythmic control over its outputs [8].

Composer’s Assistant (hereafter, “CA”) is a DAW-integrated multi-track MIDI infilling model. The *multi-track infilling problem* is the following: Given a slice of measures from a multi-track song, where the notes have been deleted from some of the track-measures (a *track-measure* is a measure within a track), fill in the notes for the deleted track-measures using the notes that remain as the context—see Figure 1. A model trained to complete this task without any further instruction might write parts that are musically coherent but different from what the user had in mind. For instance, a composer who generates a guitar track to accompany a drum track and bass track might receive a busy, high-pitched solo, a medium-speed, medium-pitched solo, a strummed rhythmic part, or any number of other types of outputs that may not match the composer’s intent for the track. It would be useful for the user to have

the ability to condition the output on parameters such as rhythm (or, if a specific rhythm is not provided, horizontal note density instead), vertical note density, and pitch range.

In this work, we build upon CA to train a new model that offers a wide range of user controls. This work was guided by conversations with several composers who have used CA for co-creative composition, and represents an effort to remedy perceived shortcomings of that system. New controls introduced to the CA system in this work include two types of rhythmic conditioning, horizontal and vertical density controls, pitch step and leap propensity controls, several types of pitch range controls, and a rhythmic interest control. We also include a control that instructs the model not to generate octave-shifted copies of music that exists in the prompt. All of our controls are designed with a DAW-integrated interface in mind. We study the power of our controls via objective metrics, we study the extent to which the model has learned the meaning of our controls, and we conduct a listening study to evaluate music created in a co-creative fashion with our model. We release our complete, DAW-integrated system and our source code.¹

2. PREVIOUS WORK

A wide range of generative music models predate this work, including MusicVAE [9], Piano Transformer [10], Coconet [11], and many others [12–19]. FIGARO [20] explored symbolic music generation with fine-grained user control, and Music SketchNet [21] explored single-track monophonic infilling with pitch and rhythm controls.

Previous DAW-integrated models include DeepBach [22], the Piano Inpainting Application [23], and Magenta Studio [24]. Cococo [25] is a DAW-like interface to Coconet, supporting 4 tracks and arbitrary user-driven infilling/part rewriting, similar to DeepBach. NONOTO [26] is a model-agnostic interface for symbolic music infilling.

Prior multi-track infilling models include MMM [3, 4], MusIAC [27], and CA [6], all of which are transformer models. The 8-bar web demo of MMM can handle up to 6 tracks, and is limited to a 4/4 time signature. MMM has two DAW-integrated versions; however, at the time of this writing they are not publicly available. MusIAC limits inputs to 3 tracks (melody, bass, and accompaniment), 16 bars, and a collection of four time signatures. CA can handle an arbitrary collection of tracks and time signatures, provided that the time signatures contain no more than 8 quarter notes per bar. CA is integrated into the REAPER DAW, and the underlying model runs locally on the user’s machine. Each of these models offers its own set of user controls for infilling: CA offers polyphony controls, MMM offers note density and polyphony controls, and MusIAC offers five controls including note density.

In most previous works, note density is computed simply by dividing the number of notes by the number of time steps. This means that a slow part with many thick chords can have the same density as a fast monophonic part, making it difficult for a user to steer the model towards the de-

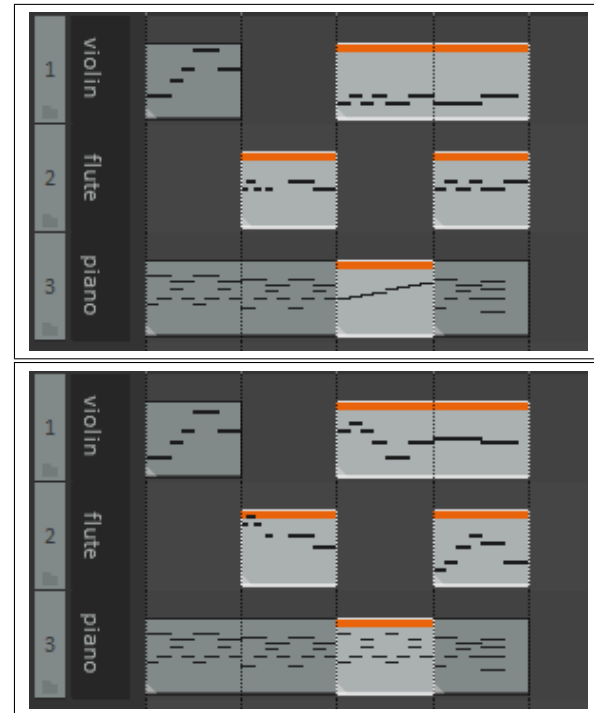


Figure 2. A prompt with 1D rhythmic conditioning in REAPER, followed by a model output. Users draw the rhythms they want in the selected MIDI items, and the model chooses pitches for these rhythms that fit with the rest of the prompt. Unselected MIDI items are included in the prompt to the encoder, and remain unchanged.

sired rhythmic speed with a density control. Additionally, most prior works have density control sliders (with values ranging from, e.g., 1–10) whose quantiles were defined by the training data. While this approach is attractive from a training perspective, it is difficult to navigate from a user perspective—e.g., what does a density of 7/10 mean?

In this work, we take a different approach to note density, first by factoring note density into horizontal (rhythmic) and vertical densities, and second by adopting musically meaningful quantiles for these measurements. We note that MuseMorphose [28] also decomposed note density into horizontal and vertical densities, albeit with a different definition of vertical density (in their work, “polyphony score”) than ours. We also develop a wide range of additional steering controls, including a user option for explicit rhythmic control. With this option, the user can supply rhythms in their model prompts, and the model chooses only the pitches—see Figure 2. To our knowledge, the controls we implement in this paper comprise the most comprehensive and user-friendly set of steering controls for multi-track MIDI infilling to date.

3. MEASUREMENTS FOR USER CONTROLS

In this section we describe the measurements underlying the user controls implemented in this work. Recall that music in MIDI format is time-quantized to a uniform grid of some number of *ticks* per quarter note.

¹ <https://github.com/m-malandro/composers-assistant-REAPER>

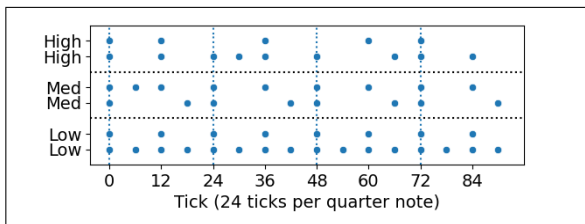


Figure 3. Six examples of rhythmic interest levels. Points mark note onsets in a 4/4 measure.

3.1 Horizontal Measurements

Horizontal note onset density. We define the *horizontal note onset density* of a collection of measures from a track to be the number of ticks with a note onset divided by the total number of ticks. In interval notation, we quantize horizontal note onset densities to the following six bins: Less than half notes; [Half notes, Quarter notes]; [Quarter notes, Eighth notes]; [Eighth notes, 16th notes]; [16th notes, 4.5 onsets per quarter note]; ≥ 4.5 onsets per quarter note.

Rhythmic interest. Given a slice of measures from a track, let v denote the binary rhythm vector of those measures. Let $\hat{v} = v - \bar{v}$ denote v , re-centered at 0. We compute dot products of \hat{v} with its nontrivial shifts and record the highest of their absolute values as a measure of rhythmic uniformity. Rhythmic uniformity is scaled by $1/||\hat{v}||^2$ and subtracted from 1 to yield *rhythmic interest*, which we divide into Low (< 0.14), Medium (≥ 0.14 and < 0.4), and High (≥ 0.4) bins. These quantiles were hand-selected by looking at many examples. See Figure 3 for examples.

3.2 Vertical Measurements

Vertical note onset density. We define the *vertical note onset density* of a collection of track-measures from a track to be the number of notes divided by the number of ticks containing an onset. In interval notation, we quantize vertical note onset densities into the following five bins: 1 note per onset; (1 note per onset, 2 notes per onset]; (2 notes per onset, 3 notes per onset]; (3 notes per onset, 4 notes per onset]; > 4 notes per onset.

Average number of pitch classes per note onset. This measure is the same as vertical note onset density, but with pitches replaced with pitch classes. It is $(\sum_t \# \text{pitch classes at } t) / \# \text{ticks containing an onset}$. We use the same bins as vertical note onset density: 1 pitch class per onset; (1 pitch class per onset, 2 pitch classes per onset]; (2 pitch classes per onset, 3 pitch classes per onset]; (3 pitch classes per onset, 4 pitch classes per onset]; > 4 pitch classes per onset.

3.3 Pitch Measurements

Pitch step and leap propensity. Given two consecutive notes, a *step* is a difference in pitch of 1–2 semitones, while a *leap* is a difference of more than 2 semitones. Pitch repetitions are neither steps nor leaps. We generalize to chords as follows: Given a chord C_1 followed by a chord C_2 , define the chord distance $d(C_1, C_2)$ to be the average of the

minimum pitch movements needed to get from the notes in C_1 to the notes in C_2 :

$$d(C_1, C_2) = \frac{1}{|C_1|} \sum_{n_1 \in C_1} \min_{n_2 \in C_2} |\text{pitch}(n_1) - \text{pitch}(n_2)|.$$

Going from a chord C_1 to a chord C_2 , we have a *repetition* when $d(C_1, C_2) = 0$, a *step* when $0 < d(C_1, C_2) \leq 2$, and a *leap* when $d(C_1, C_2) > 2$. Given a slice of measures from a track containing n chords, we count the number of steps and leaps and divide by $n - 1$ to obtain the *pitch step propensity* and *pitch leap propensity* for the slice. We quantize pitch step and leap propensities into the following seven bins: [0, 0.01), [0.01, 0.2), [0.2, 0.4), [0.4, 0.6), [0.6, 0.8), [0.8, 0.99), [0.99, 1.0).

Note onset chromagrams. When prompted to generate new tracks to accompany an arrangement already containing several tracks, we observed that the CA model would often generate a copy (possibly shifted by some number of octaves) of one of the tracks in the prompt. While this is often “correct,” it is not particularly useful for co-creative composition—if that is what the composer wanted, they could easily create this themselves. To create a control that tells the model to write genuinely new parts, for each track-measure in a song, we record whether that track-measure has the same note onset chromagram as another track-measure in its measure. (Given a track-measure T , the set of ordered pairs $\{(\text{pitch}(n) \pmod{12}, \text{onset tick}(n)) : n \text{ is a note whose onset is in } T\}$ is the note onset chromagram of T .)

3.4 Other Measurements

Pitch range. The *pitch range* of a collection of track measures is simply a record of their lowest and highest pitch.

Rhythmic information. We define the *1D rhythmic information* of a collection of measures from a track to be the set of note onset ticks and corresponding note durations after flattening all notes to the same pitch. If multiple notes share an onset, we record only the longest of their durations. The *2D rhythmic information* of a collection of measures from a track is the same, but with the number of note onsets and number of pitch classes at each onset also recorded.

4. CONTROLS AND MODEL

We use the MIDI-like token-based language from [6] to tokenize music. This language supports masking of arbitrary subsets of track-measures from any slice of measures from a song. We add additional tokens to the language to serve as control tokens for the measurements introduced in Section 3. For each of the measurements that are quantized into bins in Sections 3.1–3.3, for each bin, we create a separate control token. We also create a control token to indicate when a track-measure has a different note onset chromagram from all other track-measures in its measure—we call this token the different-note-onset-chromagram (DNOC) token. We also create pitch range

controls and explicit rhythmic controls. For pitch range control, we create four control tokens: high (strict), low (strict), high (loose), and low (loose), and we follow such a token by a pitch token to indicate the value of the measurement. With strict pitch range controls, which can be supplied on a per-track or per-track-measure basis, the model is expected to generate at least one pitch at each extreme. With loose pitch range conditioning, the model is expected to generate at least one note within 7 semitones of each extreme and not extend beyond the extremes. The idea is that a user could supply a loose pitch range for, e.g., a vocal melody, whose bounds are given by the range of the vocalist. For rhythmic conditioning, we create masked pitch tokens, which are included with rhythmic tokens (describing note onset position and note duration) in prompts. For 1D rhythmic conditioning, we include a single masked pitch token at each tick containing any number of note onsets. For 2D rhythmic conditioning, we include masked pitch tokens describing both the number of note onsets and the number of pitch classes at each onset.

As in [6], we train a T5-like [29] encoder-decoder transformer [30] model. Our main model (which we refer to as our *large* model) is 512-dimensional, with 16 encoder layers and 16 decoder layers. This model has about $3.5\times$ the number of parameters of the CA model, which is 384-dimensional, with 10 encoder layers and 10 decoder layers. To examine the effect of model scaling on performance, we also train a *small* model having the same dimension and number of layers as the CA model. For inference, we use nucleus sampling [31] with a threshold of $p = 0.85$.

During training, we mask a random subset of track-measures from a slice of measures within a song, and we ask our model to generate the tokens for the masked track-measures. All unmasked track-measures within the slice are included in the prompt provided to the encoder. In each example, we include a random subset of our control tokens in our prompts. Control tokens operating on the track level are appended to the prompt, while control tokens operating on the track-measure level are inserted into the prompt in place of the masked tokens that the model is asked to generate. For training, values for control tokens are computed using only the masked track-measures. For inference, this allows a user to specify attributes for track-measures to be filled that differ arbitrarily from the attributes of the unmasked track-measures in the prompt. During inference in the DAW, we apply only the control tokens supplied by the user.

We quantize music to a mixed grid that accommodates 32nd notes and 16th note triplets. This grid has 24 ticks per quarter note, of which 12 are valid locations for note onsets. To train our models, we use the CA training dataset (a dataset of public-domain and permissively-licensed MIDI files). As in [32], we take the “e” folder of the Lakh MIDI dataset (LMD) [33, 34] to be our validation set.

5. EVALUATION

In [6], CA generally outperformed MMM [3, 4] on objective and subjective measures. However, whether this

Model \ Task	Random infill	Track infill	Last-bar fill
Note F_1 results \uparrow			
CA2 large	77.01^a	70.74^a	78.34^a
CA2 small	76.27 ^b	69.67 ^b	77.24 ^b
CA	52.59 ^c	31.65 ^c	53.74 ^c
Precision \uparrow			
CA2 large	77.15^a	70.85^a	78.45^a
CA2 small	76.39 ^b	69.78 ^b	77.35 ^b
CA	53.02 ^c	33.76 ^c	54.72 ^c
Recall \uparrow			
CA2 large	76.90^a	70.64^a	78.24^a
CA2 small	76.15 ^b	69.58 ^b	77.15 ^b
CA	52.67 ^c	32.22 ^c	53.75 ^c
Pitch class histogram entropy difference \downarrow			
CA2 large	10.78^a	14.69^a	8.90^a
CA2 small	11.28 ^b	15.49 ^b	9.75 ^b
CA	31.65 ^c	50.53 ^c	31.60 ^c
Groove similarity \uparrow			
CA2 large	99.97 ^b	99.97^a	99.97^a
CA2 small	99.98^a	99.97^a	99.98^a
CA	97.84 ^c	96.01 ^b	97.87 ^b

Table 1. Objective infilling summary statistics. All cells are percentages of the form mean^s, where s is a letter. Different letters within a metric and column indicate significant location differences ($p < 0.01$) in the samples for those table entries according to a Wilcoxon signed rank test with Holm-Bonferroni correction.

was due to training approach, training dataset, model size, and/or other factors is unclear. To make a direct comparison between our work and previous work, we adopt CA as our baseline for comparison on objective metrics.

5.1 Objective Evaluation

We take the “f” folder of the LMD to be our test set. From each file, we select random 8-measure slices and attempt to prepare three test examples:

- Random infilling: Each track-measure in the slice is masked with probability 0.5.
- Track infilling: One track from the slice, containing note onsets in at least 7 measures, is masked completely.
- Last-bar infilling: Every track-measure in the last measure of the slice is masked.

We take 5000 examples of each type of infilling to be our test set. For each example, we take the masked notes to be the ground truth, and after evaluating the example with our models we compute precision and recall as in [6]. The F_1 score for an example is the harmonic mean of its precision and recall: $F_1 = (2 \cdot \text{precision} \cdot \text{recall}) / (\text{precision} + \text{recall})$. We also compute the pitch class histogram entropy difference and the groove similarity (as defined in [35]) between the ground truth and the output for each example.

We evaluate our test examples with our models, providing user controls describing the information in the masked track-measures to the maximum extent possible. In particular, we provide 2D rhythmic conditioning, pitch step and leap propensity, pitch range per track-measure, and the DNOC token (wherever applicable) to our models. Results are averaged and presented in Table 1. For examples evaluated by CA, we provide all of the control information available to that model—in particular, mono/poly switches at the track-measure level. We note that supplying rhythmic conditioning and pitch range information for track-measures containing only one pitch is equivalent to unmasking those track-measures. Therefore, to make a fair comparison between our models and CA, we unmask those track-measures in our prompts to CA and we give that model “credit” for those track-measures as if it had generated them itself. (14.46% of the track-measures from our test set fall into this category.) We see a dramatic increase in performance of our models relative to CA. As expected, our large model outperforms our small model (except for groove similarity for the random infilling task), but the differences in performance between our models are small. Note that the groove similarity score for our models is not 100%, indicating that our models sometimes (albeit rarely) fail to follow exactly the rhythms in their prompts.

Next, we examine the effect of each control introduced in this paper by repeating our test examples, but with only limited control information supplied to the model. For each control introduced in this paper, we examine the F_1 scores obtained by supplying only that control to the model. We also examine what happens when we supply a growing collection of controls, roughly in order of how much effort is required for a user to supply such controls in practice—specifically, we supply, in order: vertical controls, horizontal controls, our DNOC control, pitch/step leap controls, pitch range (per track), 1D rhythmic conditioning, 2D rhythmic conditioning, and finally pitch range (per track-measure). The resulting large table of F_1 scores is omitted, but the primary observation is that pitch range and rhythmic conditioning controls have the largest positive effect on the F_1 scores of model outputs. Model size has only a minor effect. Horizontal, vertical, pitch step/leap propensity, and DNOC controls also have only minor effects on F_1 scores. This raises the questions of whether the model understands the meaning of these controls and whether these controls are useful for co-creative composition, which we address in the next two sections.

5.2 Model Understanding of Control Tokens

Our model can generate music from scratch by prompting it with prompts containing no notes. To examine the understanding our model has of our control tokens, we use an empty prompting strategy: We prompt the model to generate some number of bars of single-track music from scratch, with no control tokens, and then we repeat the empty prompt with a single control token added. We use this approach to examine our horizontal, vertical, pitch step, pitch leap, and DNOC control tokens. For each con-

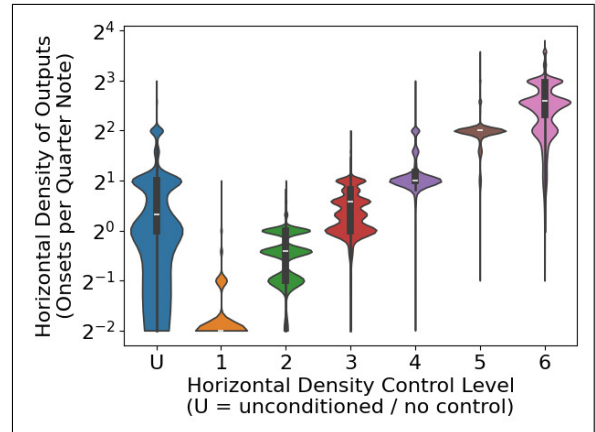


Figure 4. Horizontal density distributions.

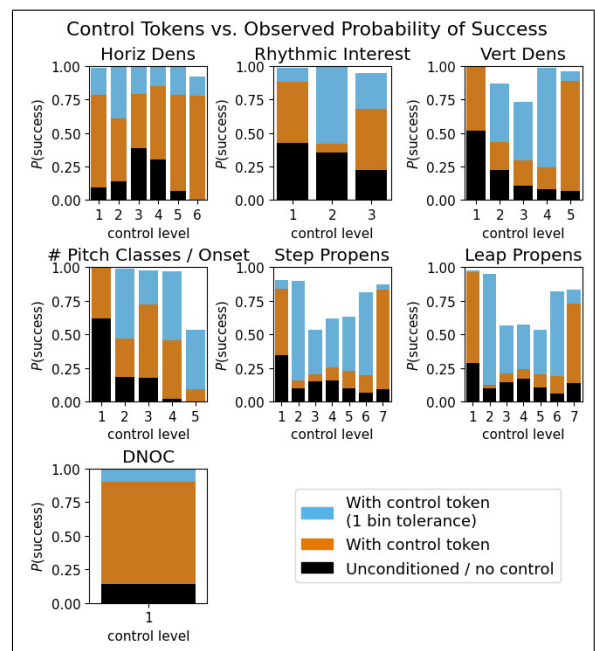


Figure 5. The observed probability of success for each control token in our empty prompting test.

trol token we generate 5000 examples, and we examine the resulting distributions. (For the DNOC control, we use a 1-bar example where a viola and cello play an ostinato one octave apart, and prompt the model to write a violin part to accompany them.)

A plot of our horizontal note onset density distributions is given in Figure 4. Plots of other distributions are similar.

For each control token, we also compare the observed probability of obtaining an output described by that token when the token is supplied versus when it is not supplied, finding that all 34 tokens mentioned above indeed push our model towards outputs having the characteristics they are intended to encode. Probabilities of success vary, however—see Figure 5. Our DNOC token performs well, with a success rate of 90.5%, as compared to unconditioned outputs having a different note onset chromagram 14.2% of the time. Our model has also done a good job of learning the concepts of horizontal density, high and

low rhythmic interest, high and low vertical density, monophonic versus polyphonic generation, and high and low step and leap propensity. However, when supplied with the appropriate control token, our model has trouble generating parts with medium rhythmic interest (with a 42% chance of success, versus 35.2% for unconditioned outputs), and with more than 4 pitch classes per onset (with a 8.92% chance of success, versus 0.04% for unconditioned outputs).

Additionally, our model has not learned the precise boundaries for our bins. For instance, when asking the model to generate a part with a horizontal density in the interval [Quarter notes, Eighth notes), we observed that our model would often generate a straight eighth note pattern. When allowing for a tolerance of up to one bin away, though, all of our control tokens have a success rate of over 50%, and 26/34 of them have a success rate of over 80%.

Our model learned the meaning of each of these control tokens solely via natural training data. In future work, it may be possible to achieve better control token understanding performance by training on synthetic examples and/or by changing the boundaries of the bins.

5.3 Subjective Evaluation

We used our model in an iterative, co-creative fashion to create 12 snippets of music, each about 16 bars long. To create these snippets, we started with 6 pieces of real music and deleted one or more tracks in the music. We then used our model interactively to fill these deleted tracks. Specifically, each time the model generated music, we kept the generated track-measures we liked, and continued using the model to fill the remaining track-measures until done. We carried out this process under two sets of rules:

- CA-: We did not use rhythmic conditioning, and did not hand-edit the notes in any outputs.
- CA+: We were allowed to use every control available, including rhythmic conditioning, and we were allowed to make small hand edits to outputs. This is the approach closest to how the model would be used in practice.

In both cases we were allowed as many generations as we wanted. We recorded the creation of these examples, which took about 2 hours in total.² Volunteers were asked to score as many of these 18 snippets of music as they wished, according to perceived **R**hythmic correctness, **P**itch correctness, **M**emorability, and **O**verall, each on a scale of 0 to 4. Volunteers were not told which pieces of music were real and which were composed with our system. 28 individuals volunteered and ranked an average of 10.4 snippets along each of our 4 axes. Aggregated results are presented in Figure 6. Each bar in this figure represents 94–99 data points.

We compare real music to music composed with our two approaches CA- and CA+ with paired *t*-tests, using data from whenever a volunteer ranked both. Each of these

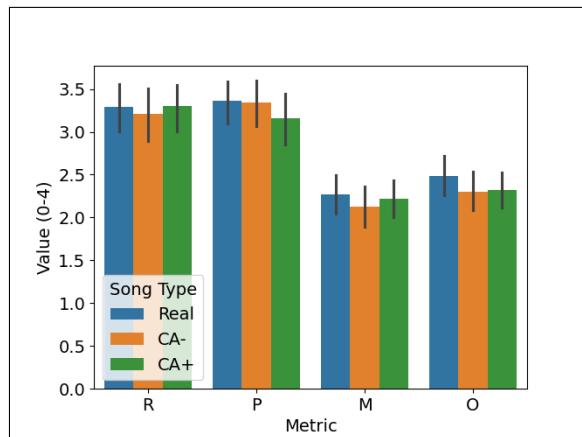


Figure 6. Subjective evaluation of our model.

Metric	Real	Metric	Real
R	CA-	M	CA-
	0.693	CA+	0.052
P	CA-	O	CA-
	0.887		CA+
M	CA-	CA-	0.039
	0.791	CA+	0.099
O	CA-		
	0.211		

Table 2. Uncorrected *p*-values from paired *t*-tests in our subjective comparisons.

8 tests involved 93–96 pairs of values. See Table 2. After Holm-Bonferroni correction, we do not find a significant ($p < 0.05$) difference between real music and music composed with either approach in any of these 8 tests. We know that our model is competent at avoiding rhythmic and pitch errors, so we are not surprised by our R and P results. However, we had some difficulty creating a proper drum part in one example with the CA- approach, and we expected to find at least one significant M or O difference. We suspect that there are subjective quality differences, at least between real music and the music we created with the CA- approach, that are not large enough for this study to detect reliably. Nevertheless, we believe that this indicates that music co-created with our model has the potential to be on par with human-composed music. We also believe that results of similar studies would vary greatly depending on the skill of the composer using the system. We invite the reader to listen to our samples³ and to try our system.

6. CONCLUSION

We have introduced a wide range of steering controls for multi-track MIDI infilling, and we have trained a transformer model to implement these controls. We have created an interface for our controls, and we have integrated our model and controls into the REAPER digital audio workstation for co-creative symbolic music composition. Our work in this paper comprises Composer’s Assistant 2, and we have released our complete system and source code.⁴

² <https://www.youtube.com/watch?v=WUQTIOEv3WM>

³ <https://www.youtube.com/watch?v=4xt9fBqluQg>

⁴ <https://github.com/m-malandro/composers-assistant-REAPER>

7. ETHICS STATEMENT

There are currently unresolved ethical and legal questions regarding the inclusion of copyrighted data in training sets for generative models. While we suspect it would be possible to obtain better objective (and possibly subjective) results by training a model on a larger and more varied dataset (e.g., the Lakh MIDI dataset [33, 34]), we chose to train our models only on copyright-free and permissively licensed files, primarily for the following two reasons:

First, we want our model outputs to be usable by composers. While there is a possibility that our models may output copyrighted musical information (even if such information was not present in the training dataset), we believe that training only on copyright-free and permissively-licensed musical data minimizes this possibility.

Second, for many composers, we view the models that we have released not as the models that the composers would actually use in their work, but rather as starting points for customization and personalization. Many composers we have spoken to have said that models that write “generic” music are not useful to them. Instead, they want generative systems that can suggest ideas in their own style. Due to our training set, the models we have released are most proficient at infilling classical, choral, and folk music. However, informal experiments suggest that our models can be finetuned on a relatively small number of MIDI files to write in the style of those files, and hence the style limitations of our released models may not matter. The code we have released supports finetuning by end users. While this can benefit composers who wish to use our system, there is also the risk that our models may be finetuned by users to impersonate the styles of others.

8. REFERENCES

- [1] M. Newman, L. Morris, and J. H. Lee, “Human-AI Music Creation: Understanding the Perceptions and Experiences of Music Creators for Ethical and Productive Collaboration,” in *Proc. 24th Int. Society for Music Information Retrieval Conf.*, Milan, Italy, 2023, pp. 80–88.
- [2] C.-Z. A. Huang, H. V. Koops, E. Newton-Rex, M. Dinulescu, and C. J. Cai, “AI Song Contest: Human-AI Co-Creation in Songwriting,” in *Proc. 21st Int. Society for Music Information Retrieval Conf.*, Montréal, Canada, 2020, pp. 708–716.
- [3] J. Ens and P. Pasquier, “MMM : Exploring Conditional Multi-Track Music Generation with the Transformer,” *arXiv preprint arXiv: 2008.06048*, 2020.
- [4] —, “Flexible Generation with the Multi-Track Music Machine,” in *Extended Abstracts for the Late-Breaking Demo Session of the 21st Int. Society for Music Information Retrieval Conf.*, Montréal, Canada, 2020.
- [5] R. B. Tchemeube, J. Ens, C. Plut, P. Pasquier, M. Safi, Y. Grabit, and J.-B. Rolland, “Evaluating Human-AI Interaction via Usability, User Experience and Acceptance Measures for MMM-C: A Creative AI System for Music Composition,” in *Proc. 32nd Int. Joint Conf. Artificial Intelligence*, 2023, pp. 5769–5778.
- [6] M. Malandro, “Composer’s Assistant: An Interactive Transformer for Multi-Track MIDI Infilling,” in *Proc. 24th Int. Society for Music Information Retrieval Conf.*, Milan, Italy, 2023, pp. 327–334.
- [7] “Andromeda | AI Opera,” September 2, 2023. [Online]. Available: <https://www.valdovurumai.lt/lt/renginiai/item/8413/>
- [8] “InMuse Vilnius | Day 3 | Forum on Innovation in Music, Stage Arts and Entertainment,” 2023. [Online]. Available: <https://www.youtube.com/watch?v=je7sUtPNyRs&t=12100s>
- [9] A. Roberts, J. Engel, C. Raffel, C. Hawthorne, and D. Eck, “A Hierarchical Latent Vector Model for Learning Long-Term Structure in Music,” in *Proc. 35th Int. Conf. Machine Learning*, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. PMLR, 10–15 Jul 2018, pp. 4364–4373.
- [10] C.-Z. A. Huang, A. Vaswani, J. Uszkoreit, I. Simon, C. Hawthorne, N. Shazeer, A. M. Dai, M. D. Hoffman, M. Dinulescu, and D. Eck, “Music Transformer,” in *Int. Conf. Learning Representations*, 2019.
- [11] C.-Z. A. Huang, T. Cooijmans, A. Roberts, A. Courville, and D. Eck, “Counterpoint by Convolution,” in *Proc. 18th Int. Society for Music Information Retrieval Conf.*, Suzhou, China, 2017, pp. 211–218.
- [12] C. Payne, “MuseNet,” openai.com/blog/musenet, 2019.
- [13] J. Liu, Y. Dong, Z. Cheng, X. Zhang, X. Li, F. Yu, and M. Sun, “Symphony Generation with Permutation Invariant Language Model,” in *Proc. 23rd Int. Society for Music Information Retrieval Conf.*, Bengaluru, India, 2022, pp. 551–558.
- [14] S. Wei, G. Xia, Y. Zhang, L. Lin, and W. Gao, “Music Phrase Inpainting Using Long-Term Representation and Contrastive Loss,” in *IEEE Int. Conf. Acoustics, Speech and Signal Processing*, 2022, pp. 186–190.
- [15] A. Pati, A. Lerch, and G. Hadjeres, “Learning to Traverse Latent Spaces for Musical Score Inpainting,” in *Proc. 20th Int. Society for Music Information Retrieval Conf.*, Delft, The Netherlands, 2019, pp. 343–351.
- [16] C. Benetatos and Z. Duan, “Draw and Listen! A Sketch-Based System for Music Inpainting,” *Trans. Int. Society for Music Information Retrieval*, vol. 5, no. 1, pp. 141–155, 2022.

- [17] C.-J. Chang, C.-Y. Lee, and Y.-H. Yang, “Variable-Length Music Score Infilling via XLNet and Musically Specialized Positional Encoding,” in *Proc. 22nd Int. Society for Music Information Retrieval Conf.*, online, 2021, pp. 97–104.
- [18] G. Mittal, J. Engel, C. Hawthorne, and I. Simon, “Symbolic Music Generation with Diffusion Models,” in *Proc. 22nd Int. Society for Music Information Retrieval Conf.*, online, 2021, pp. 468–475.
- [19] L. Min, J. Jiang, G. Xia, and J. Zhao, “Polyffusion: A diffusion model for polyphonic score generation with internal and external controls,” in *Proceedings of the 24th International Society for Music Information Retrieval Conference, ISMIR, 2023*.
- [20] D. von Rütte, L. Biggio, Y. Kilcher, and T. Hofmann, “FIGARO: Controllable music generation using learned and expert features,” in *Int. Conf. Learning Representations, 2023*.
- [21] K. Chen, C. i Wang, T. Berg-Kirkpatrick, and S. Dubnov, “Music SketchNet: Controllable Music Generation via Factorized Representations of Pitch and Rhythm,” in *Proc. 21st Int. Society for Music Information Retrieval Conf.*, Montréal, Canada, 2020, pp. 77–84.
- [22] G. Hadjeres, F. Pachet, and F. Nielsen, “DeepBach: a Steerable Model for Bach Chorales Generation,” in *Proc. 34th Int. Conf. Machine Learning*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. PMLR, 06–11 Aug 2017, pp. 1362–1371.
- [23] G. Hadjeres and L. Crestel, “The Piano Inpainting Application,” *arXiv preprint arXiv: 2107.05944*, 2021.
- [24] A. Roberts, C. Kayacik, C. Hawthorne, D. Eck, J. Engel, M. Dinculescu, and S. Nørly, “Magenta Studio: Augmenting Creativity with Deep Learning in Ableton Live,” in *Proc. Int. Workshop on Musical Metacreation (MUME)*, 2019.
- [25] R. Louie, A. Coenen, C. Z. Huang, M. Terry, and C. J. Cai, “Novice-AI Music Co-Creation via AI-Steering Tools for Deep Generative Models,” in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, ser. CHI ’20. New York, NY, USA: Association for Computing Machinery, 2020, p. 1–13. [Online]. Available: <https://doi.org/10.1145/3313831.3376739>
- [26] T. Bazin and G. Hadjeres, “NONOTO: A Model-agnostic Web Interface for Interactive Music Composition by Inpainting,” in *Proc. 10th Int. Conf. Computational Creativity*, 2019.
- [27] R. Guo, I. Simpson, C. Kiefer, T. Magnusson, and D. Herremans, “MusIAC: An Extensible Generative Framework for Music Infilling Applications with Multi-level Control,” in *Artificial Intelligence in Music, Sound, Art and Design. EvoMUSART 2022. Lecture Notes in Computer Science*, T. Martins, N. Rodríguez-Fernández, and S. M. Rebelo, Eds., vol. 13221. Springer, Cham, 2022, pp. 341–356.
- [28] S.-L. Wu and Y.-H. Yang, “MuseMorphose: Full-Song and Fine-Grained Piano Music Style Transfer with One Transformer VAE,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 1953–1967, 2023.
- [29] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, “Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer,” *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, 2020.
- [30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is All you Need,” in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.
- [31] A. Holtzman, J. Buys, L. Du, M. Forbes, and Y. Choi, “The Curious Case of Neural Text Degeneration,” in *Int. Conf. Learning Representations, 2020*.
- [32] J. Thickstun, D. L. W. Hall, C. Donahue, and P. Liang, “Anticipatory Music Transformer,” *Transactions on Machine Learning Research*, 2024. [Online]. Available: <https://openreview.net/forum?id=EBNJ33FcrI>
- [33] C. Raffel, “Learning-Based Methods for Comparing Sequences, with Applications to Audio-to-MIDI Alignment and Matching,” Ph.D. dissertation, 2016.
- [34] —, “The Lakh MIDI Dataset v0.1,” <https://colinraffel.com/projects/lmd/>.
- [35] S.-L. Wu and Y.-H. Yang, “The Jazz Transformer on the Front Line: Exploring the Shortcomings of AI-Composed Music Through Quantitative Measures,” in *Proc. 21st Int. Society for Music Information Retrieval Conf.*, Montréal, Canada, 2020, pp. 142–149.

TOWARDS ZERO-SHOT AMPLIFIER MODELING: ONE-TO-MANY AMPLIFIER MODELING VIA TONE EMBEDDING CONTROL

Yu-Hua Chen^{1,3} Yen-Tung Yeh^{2,3} Yuan-Chiao Cheng³
Jui-Te Wu³ Yu-Hsiang Ho³ Jyh-Shing Roger Jang¹ Yi-Hsuan Yang²

¹ Graduate Institute of Networking and Multimedia, National Taiwan University, Taiwan

² Department of Electrical Engineering, National Taiwan University, Taiwan

³ Positive Grid, Taiwan

ABSTRACT

Replicating analog device circuits through neural audio effect modeling has garnered increasing interest in recent years. Existing work has predominantly focused on a one-to-one emulation strategy, modeling specific devices individually. In this paper, we tackle the less-explored scenario of one-to-many emulation, utilizing conditioning mechanisms to emulate multiple guitar amplifiers through a single neural model. For condition representation, we use contrastive learning to build a tone embedding encoder that extracts style-related features of various amplifiers, leveraging a dataset of comprehensive amplifier settings. Targeting zero-shot application scenarios, we also examine various strategies for tone embedding representation, evaluating referenced tone embedding against two retrieval-based embedding methods for amplifiers unseen in the training time. Our findings showcase the efficacy and potential of the proposed methods in achieving versatile one-to-many amplifier modeling, contributing a foundational step towards zero-shot audio modeling applications.

1. INTRODUCTION

Neural audio effect modeling, the task of simulating analog circuitry and digital audio effects using neural networks, has garnered significant interest driven by advances in deep learning [1–9]. Various network architectures have been proposed for emulating different effect pedals and guitar amplifiers (amps). Through modeling nonlinearities, harmonic distortions, and transient responses inherent to analog circuitry, neural models offer an alternative to their physical counterparts. Such models enable widespread applications in automatic mixing [10–12], audio style transfer [13, 14] and beyond, contributing to new music production and sound design workflows.

Real-world audio effect pedals or amplifiers are known for their rich acoustic diversity, leading to different “tones.”

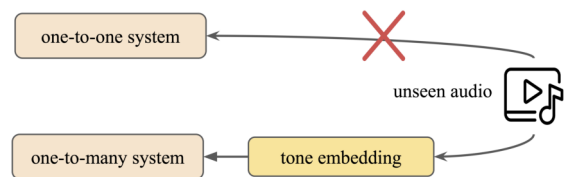


Figure 1: A one-to-one approach cannot emulate an unseen audio effect. In contrast, the proposed one-to-many approach can achieve zero-shot modeling by using a tone embedding encoder that turns a reference audio example of that effect into a conditioning input at inference time.

There are multiple types of pedals (e.g., compression, EQ, distortion, reverb, modulation), possessing different characteristics. Even pedals of the same effect type can sound fairly differently due to different implementations. A user can further adjust the device parameters via tuning the associated knobs (e.g., a gain knob) to shape the tone. Moreover, it is common to interconnect effect pedals in various orders and forms to constitute an “effect chain,” collectively creating a unique tone. A guitar amplifier can also be viewed as an effect chain as there are pre-amp, tone stack, power amp, and cabinet components arranged in specific order that vary across brands and underlying circuitries. The rich diversity of tones represents a central challenge for neural audio effect modeling.

While exciting progress is being made, to simplify the task, most prior research has concentrated on the *one-to-one* mapping setting, building a neural model for emulating the behavior of only *one* device (i.e., a pedal or an amp) at a time [15, 16]. Some models make further simplification and model only a certain parameter setting (a.k.a., “snapshot”) of a device and do not take any condition signals [3, 4, 6], while others incorporate device parameters as conditions for a “full” modeling of the device [2, 17].

Little work, if any, has been done to tackle the more challenging task of building a single universal model that can emulate multiple devices at once. This is harder as different devices are built from varying combinations and configurations of analog circuits, leading to distinct sonic characteristics. We posit that a transition to such a *one-to-many* setting is beneficial. On one hand, it holds the key towards neural audio effect modeling with broader versatility, better reflecting the complexity seen in the real world.



On the other hand, doing so may empower the single model to learn the similarities and distinctions among different audio effects, building up a “tone space” that permits interpolation and extrapolation of *seen* tones (i.e., tones made available to the model during training time) to approximate an *unseen* tone or to create a *new* tone.¹

We are in particular interested in the case of emulating unseen tones in this paper. Specifically, we envision a *zero-shot* scenario where we have a model that can take a reference audio signal in the style of an unseen tone as the input condition, and learn to *clone* the tone on-the-fly during the inference time, with no (i.e., “zero”) model re-training. See Figure 1 for an illustration.

In this paper, we set forth to tackle one-to-many neural modeling of multiple guitar amps by a single model. Our training data contains pairs of clean (dry) signal and the corresponding wet signal rendered by an amp, out of $N = 9$ possible guitar amps featuring low-gain and high-gain ones. Instead of building N models, one for each amp, our one-to-many model can take a condition signal indicating the specific amp tone of interest and convert a given clean signal into the corresponding wet signal at inference time. The attempt to build such an end-to-end one-to-many model represents the first contribution of this work.

While a straightforward approach to condition the generation process is via using a look-up table (LUT) to indicate the target amplifier, the LUT approach cannot deal with unseen amplifiers. Targeting zero-shot applications, we use contrastive learning [18] to build a *tone embedding encoder* to capture tone- (or style-) related information of a referenced audio, and then use the resulting tone embedding as the condition for generation, as illustrated in Figure 2. We show via experiments that, for seen guitar amps, the tone embedding seems information richer than the LUT embedding, leading to more effective one-to-many modeling. Moreover, the tone embedding approach also works well for zero-shot learning of unseen amps, as we can turn arbitrary referenced audio of that amp into a condition to clone its style. In our evaluation, we use not only two additional guitar amps not seen during training time, but also self-recorded audio signals of guitar playing to validate the model’s effectiveness in real-world conditions. The idea to realize zero-shot tone transfer in our one-to-many model stands as the second contribution of our work.

We invite readers to visit our demo website for audio samples demonstrating the result of our model.²

2. BACKGROUND

The process of applying effects to an audio signal in the real world can be described by $\mathbf{y} = f(\mathbf{x}, \phi)$, where $\mathbf{x} \in \mathbb{R}^{C \times T}$ denotes the input signal with C channels and T samples, $\mathbf{y} \in \mathbb{R}^{C' \times T}$ the processed output, and $\phi \in \mathbb{R}^M$ the control signal with M distinct control parameters. The function f encapsulates the accumulative transformation

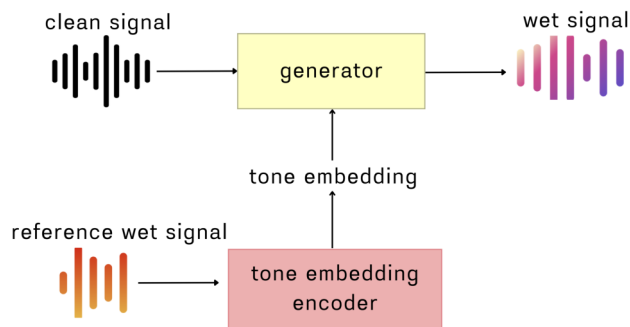


Figure 2: Diagram of the audio processing workflow. A clean signal \mathbf{x} is input into the generator \mathcal{G} , which uses the tone embedding from the tone embedding encoder to produce the wet signal \mathbf{y} . The encoder \mathcal{E} generates the tone embedding ϕ by analyzing a reference wet signal \mathbf{z} .

exerted by the devices involved in the effect chain connecting the input and the output. Neural audio effect modeling aims to replicate this implicit function f using either a single neural network [4, 11] (i.e., the “end-to-end” approach) or a cascade of modularized networks [10, 14]. The latter approach is feasible only when the devices involved in the transformation is known beforehand, not applicable to the envisioned zero-shot scenario. Therefore, we focus on the the end-to-end approach in this work.

Common *backbone* models for end-to-end audio effect modeling include convolution neural network (CNN)-based [1, 3, 6, 17], recurrent neural network (RNN)-based [2, 5], and hybrid models [4, 11]. As mentioned in Section 1, existing models mostly model one device at a time, sometimes even neglecting the control parameters ϕ . For those models that consider control parameters, a “condition representation” to represent the control signal and a “conditioning mechanism” to condition the generation process by the control signal is needed.

For the *condition representation*, the common approach in the field of neural audio effect modeling is via quantizing the device parameters and then using a one-hot encoding or look-up table (LUT) to indicate the specific parameter setting of interest. Such *ID-based* embeddings has also been used in other one-to-many audio modeling tasks, such as singing voice conversion (SVC) [19–21]. Importantly, the LUT embeddings are learned at training time and then fixed at inference time, thereby failing to accommodate unseen conditions. This is not a problem for modeling the parameters of a device, because there will not be unseen parameters for a known device. However, it is an issue in our one-to-many setting, as we are not able to exhaustively include all the possible devices at training time.

For the *conditioning mechanism*, a common approach is via “concatenation,” expanding ϕ over time to get $\phi^+ \in \mathbb{R}^{M \times T}$ and concatenating it with the input \mathbf{x} channel-wisely, forming a new input $\mathbf{x}^+ \in \mathbb{R}^{(C+M) \times T}$ to the model. Concatenation has been used by both CNN- [1] and RNN-based models [2]. Another common approach is via “feature-wise linear modulation” (FiLM) [22], which has usually been adopted by CNN-based models [7, 17].

¹ We note that, the idea of using “devices” to define tones is less applicable in the neighboring automatic mixing tasks [11, 12].

² <https://ss12f32v.github.io/Guitar-Zero-Shot/>

FiLM injects conditions to the model by using $\phi \in \mathbb{R}^M$ as the input to predict different scaling γ_l^c and shifting β_l^c coefficients through a few linear (dense) layers, and then performing element-wise affine transformation of the intermediate feature maps of each layer of the backbone model, i.e., $\text{FiLM}(\mathbf{F}_l^c, \gamma_l^c, \beta_l^c) = \gamma_l^c \mathbf{F}_l^c + \beta_l^c$, for each layer l and each c -th channel of the corresponding feature map.

Our work differs from the prior work in the following three aspects. First, we tackle multi-device modeling, conditioning our network by “devices” rather than “parameters” of a single device. Second, we adopt the idea of *content-based* embeddings developed in SVC [23–27] to compute the condition representation instead of the ID-based embeddings, so as to deal with unseen devices. This is specifically done via the proposed tone embedding encoder, whose details are introduced in Section 3.1. Finally, we report experiments on zero-shot tone transfer.

3. MULTI-TONE AMPLIFIER MODELING

Our goal is to develop a conditional generator, denoted as $\mathcal{G}(\mathbf{x}, \phi)$, that can replicate the effect of an amplifier’s audio effect chain f , guided by a tone embedding ϕ . The objective of the generator \mathcal{G} is to produce an output \mathbf{y} from the input \mathbf{x} , where \mathbf{x} and \mathbf{y} are temporally aligned and share the same musical content, but exhibit different tones (i.e., clean tone vs. a target amp tone). In our work, the tone embedding ϕ is a learnable representation of various tones. Specifically, as depicted in Figure 2, we employ a tone embedding encoder, denoted as \mathcal{E} , to derive the tone embedding ϕ from a reference audio signal \mathbf{z} , with $\phi = \mathcal{E}(\mathbf{z})$. It is important to note that the reference signal \mathbf{z} and the target \mathbf{y} *must* match in tone, but their musical content *can* be different. More details on this in Section 3.3.

3.1 Tone Embedding Encoder

Our goal is to train an encoder \mathcal{E} that can extract tone- (or style-) related features from a given wet guitar audio signal, while neglecting content-related information. Namely, this entails style/content disentanglement. We propose to employ the self-supervised contrastive learning framework of SimCLR [18], which was originally for images [28, 29], to train such an audio encoder. Specifically, our idea is to treat pairs of audio clips with *different* playing contents but the *same* tone as the “positive” pairs, and otherwise the “negative” pairs. Any audio clip would go through the same audio encoder to get an embedding representation of that clip, and the learning objective of SimCLR is to train the encoder such that the embeddings of clips from a positive pair are close to each other, while embeddings of clips from a negative pair are separated apart. By virtue of the way positive and negative data pairs is constructed, the encoder learns to project clips of the same tone to similar places in the embedding space, regardless of the underlying musical content.³

³ The use of contrastive learning for representation learning has been widely done in the musical audio domain before, for tasks such as music classification [30–33] and automatic mixing [12]. What’s different here is the application of contrastive learning to the domain of guitar amp tones

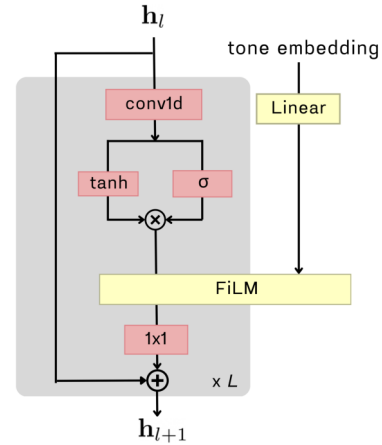


Figure 3: Diagram of a layer of the generator \mathcal{G} , which uses gated convolutional neural network (GCN) [6] as the backbone and FiLM [22] for conditioning; \mathbf{h}_l denotes the output of the previous layer and \mathbf{h}_{l+1} the current layer.

It is easier to train the encoder if we have a collection of realistic wet guitar signals featuring different combinations of contents and tones. Such a dataset, however, is hard to come by. Guitar signals collected in the wild need to be transcribed and labeled to get content- and style-related information. Alternatively, we may use software simulation to convert a set of clean signals into different wet signals, but the tones supported by open-source tools such as Pedalboard [34, 35] are limited in diversity. Through a joint work with Positive Grid, a leading guitar amp and effect modeling company, we have the advantage of accessing high-quality and diverse data. The training data for the encoder \mathcal{E} is from using a larger number of clean guitar signals as input to the company’s commercial software to render wet signals with a great diversity of tones using different combinations of amplifiers and effect pedals.

3.2 Conditional Generator

Paying more attention on the condition representation part (i.e., ϕ and \mathcal{E}), we use existing methods for the model backbone and the conditioning mechanism for our conditional generator \mathcal{G} . Specifically, we adapt the gated convolutional neural network (GCN) [6] as our model backbone, for its demonstrated efficacy in neural audio effect modeling. For the conditioning mechanism, we use FiLM [22]. As shown in Figure 3, each layer of GCN passes the output of the previous layer \mathbf{h}_l through a 1D convolution with a progressively increasing dilation factor, followed by sigmoid and tanh for gated activation. The resulting feature map is then modulated by the FiLM module, before being processed by a 1x1 conv1d and then added with the output of the previous layer with a residual link. The same tone embedding ϕ is used in all the L layers of GCN, but each layer l has its own learnable parameters. The outputs from all layers are finally concatenated and processed through a final 1x1

(which has not been attempted before, to our best knowledge), and the way we prepare positive/negative pairs for tone representation learning.

conv1d mixing layer to generate the output signal \mathbf{y} .⁴

The tone embedding encoder \mathcal{E} is trained with a large set of wet signals rendered with different combinations of amps and effect pedals, while the generator \mathcal{G} is trained separately with a smaller set of paired data of clean and wet signals rendered with different amps (not using effect pedals). The encoder \mathcal{E} is trained beforehand and then fixed (i.e., parameters frozen) while training the generator \mathcal{G} .

The training data of \mathcal{G} is also from Positive Grid, containing 30 minutes of clean input data (for \mathbf{x}) rendered with 9 different guitar amplifiers (for \mathbf{y}). For performance evaluation, we divide the clean signals into training, validation, and test sets with an 80/10/10 ratio. According to the company’s taxonomy, there are 3 types of amps:

- *High-gain* tones are perceived as highly distorted. We have Mesa Boogie Mark IV (amp1), PRS Archon 100 (amp2), and Soldano SLO-100 (amp3).
- *Low-gain* tones are often recognized as an overdrive sound. Our data contains Fender Tweed Deluxe (amp4), Vox AC30 (amp5), and Matchless DC30 (amp6).
- *Crunch* tones exhibit a mid-range gain level, in between low- and high-gain. Our data contains Vox AC30 Handwired Overdriven (amp7), Friedman BE100 (amp8), and Overdriven Marshall JTM45 (amp9).

3.3 Source of the Reference Audio Signal

While training \mathcal{G} , for each data pair $\{\mathbf{x}, \mathbf{y}\}$, the encoder \mathcal{E} takes a reference signal \mathbf{z} providing information of the tone of \mathbf{y} . Therefore, \mathbf{y} and \mathbf{z} must match in tone. However, interestingly, \mathbf{y} and \mathbf{z} can be different in content.

The naïve **paired reference** method of simply setting $\mathbf{z} = \mathbf{y}$ demands the reference signal and target output, and accordingly the input \mathbf{x} , to play the same content. This is fine at training time, but is too restrictive and not practical at inference time, especially for zero-shot scenarios. Following the idea of using “target-unaligned audio” of a prior work [8], we instead employ an **unpaired reference** method, selecting a signal \mathbf{z} *at random* from the training set as long as it is rendered with the same amp tone as \mathbf{y} .

Besides zero-shot capability, the proposed unpaired reference method may further encourage style/content disentanglement, because here ϕ only provides information concerning the tone of the target, not its content.

3.4 Zero-shot Tone Transfer

At inference time, the proposed model \mathcal{G} holds the potential to clone the tone of a reference signal \mathbf{z}^* for even unseen tones and unseen reference signals, via using $\mathcal{E}(\mathbf{z}^*)$ as the condition ϕ^* . This might be feasible as the encoder \mathcal{E} has actually been trained on a great diversity of tones besides the limited number of N tones used to train \mathcal{G} .

⁴ Unlike [6], we pad zeros at the start of the signal but not in the intermediate feature maps, for otherwise there would be unwanted impulse-like sounds. This means each layer ends up being shorter in our case. To keep the output size consistent across layers, we crop the residual part of each layer following the causal principle.

Besides using $\phi^* = \mathcal{E}(\mathbf{z}^*)$, we consider the following two *retrieval-based* alternatives to get ϕ^* , referred to as “nearest-embedding” and “mean-embedding” respectively.

- $\phi^* = \arg \max_{\phi \in \Phi} \text{sim}(\mathcal{E}(\mathbf{z}^*), \phi)$, where $\phi = \mathcal{E}(\mathbf{z})$ denotes the embedding for a reference signal *seen* and sampled from the training set, $\text{sim}(\cdot, \cdot)$ the cosine similarity between two vectors, and Φ the collection of embeddings for such seen reference signals from the training set to be compared against the query $\mathcal{E}(\mathbf{z}^*)$. Namely, this method picks the known reference signal \mathbf{z} whose embedding is closest to that of the unseen reference \mathbf{z}^* as the surrogate to condition the generation process. We set the size of the candidate set $|\Phi| = 3,600$ in our implementation.
- $\phi^* = \arg \max_{\phi \in \{\psi_1, \psi_2 \dots, \psi_N\}} \text{sim}(\mathcal{E}(\mathbf{z}^*), \phi)$, where ψ_n stands for the average of the embeddings associated with the n -th amp tone (out of the N seen amp tones) from the aforementioned candidate set Φ . This can be viewed as an LUT-like approach because we use one mean embedding to represent each amp tone for retrieval.

3.5 Implementation Details

The encoder \mathcal{E} is developed by Positive Grid using in-house data and implementation. It is an audio encoder that processes mel-spectrograms of guitar signals, trained with a batch size of 200 short audio clips sampled at 16kHz, along with data augmentation techniques such as adding noise and random cropping.

For the generator \mathcal{G} , we followed [14] and applied -12 dBFS peak normalization to the training data to balance the sound levels of different amplifiers and ensure headroom for distortion. We randomly paired a 3.5-second monaural clean input sampled at 44.1kHz with an amp output from the 9 amplifiers as the training examples. We trained the generator \mathcal{G} on an NVIDIA RTX 3090 GPU (with 24 GB memory), using the Adam optimizer [36] with a learning rate of $1e-3$ and a batch size of 12. While both the input and output of \mathcal{G} are time-domain waveforms, we used complex-valued spectral loss as the training objective, with an STFT window length of 2,048 and a hop length of 512, for this loss function led to better result in our pilot study.

For the architecture of \mathcal{G} , we configured the GCN with $L = 12$ conv1D layers, each with 16 channels, followed by a final 1×1 conv1d layer combining the outputs of all the preceding layers to a monaural waveform. For conditioning, we firstly projected each 512-dim tone embedding produced by \mathcal{E} to a 128-dim vector, then applied a series of 10 linear layers at each GCN layer to predict the scaling and shifting coefficients γ_l^c and β_l^c for FiLM. The total number of trainable parameters of \mathcal{G} is around 120k, and the model training of \mathcal{G} converged in 1/2 days.

4. EXPERIMENTAL RESULTS

4.1 Tone Embedding Visualization

We examine the tone embedding space of the 9 target amps first, using t-SNE [37] for dimension reduction and visual-

		GCN	FiLM-GCN			Concat-GCN	
		one-to-one	LUT	ToneEmb (paired)	ToneEmb (unpaired)	LUT	ToneEmb (paired)
high-gain	Amp1	0.0420	0.1441	0.1189	0.0777	0.1593	0.1523
	Amp2	0.0268	0.1951	0.0670	0.1189	0.1741	0.1208
	Amp3	0.0527	0.1659	0.1254	0.1143	0.1777	0.1304
low-gain	Amp4	0.0087	0.0698	0.0230	0.0275	0.0618	0.0775
	Amp5	0.0004	0.0813	0.0167	0.0138	0.0334	0.0166
	Amp6	0.0014	0.0947	0.0169	0.0121	0.0779	0.0275
crunch	Amp7	0.0393	0.1022	0.0860	0.0885	0.0733	0.0988
	Amp8	0.0124	0.1583	0.0760	0.0604	0.1562	0.0775
	Amp9	0.0035	0.1593	0.0375	0.0290	0.1211	0.0407

Table 1: Efficacy of various models in modeling 9 guitar amplifiers, measured in complex STFT loss on the test data. The leftmost column shows the result of the baseline *one-to-one* GCN approach, building one model per amp. The others are *one-to-many*, building a single model for all the 9 amps, using FiLM (middle three) or concatenation (last two) as the conditioning mechanism. For conditioning representation, we evaluate LUT and the proposed tone embedding (‘ToneEmb’) with either paired or unpaired reference (cf. Section 3.3). Best results of one-to-many models are highlighted.

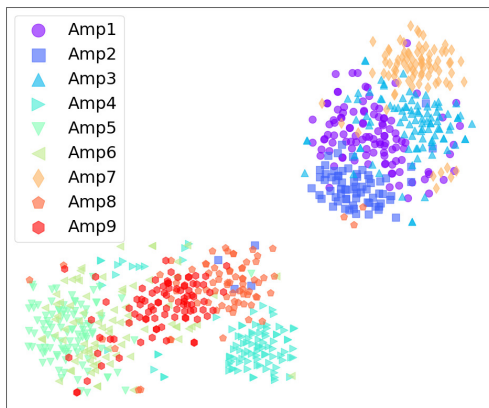


Figure 4: A t-SNE visualization of the tone embeddings from the wet signals of the $N = 9$ amps. Each point represents a tone embedding extracted from a wet signal, with color and shape indicating the category of the amp tone. We see 2 big cross-amp clusters and 9 small clusters for each amp, suggesting the ability of the encoder \mathcal{E} to distinguish between different tones based on their embeddings.

ization. Figure 4 shows that the embeddings of the wet signals of the same amp, while differing in musical content, indeed cluster together in the projected 2-D space, demonstrating the efficacy of our encoder \mathcal{E} in capturing tone-related information. Moreover, we see two big clusters, which somehow separate high-gain amps (amps 1–3) from low-gain amps (amps 4–6). There are also overlaps, suggesting some similarity among the amps. Amp 7 is closer to high-gain, while amps 8 and 9 closer to low-gain.

4.2 Efficacy of One-to-many Neural Amp Modeling

Next, we evaluate the efficacy of modeling the 9 target amps, comparing different approaches (one-to-one versus one-to-many), conditioning mechanisms (FiLM or the simpler concatenation), and condition representation (the pro-

posed tone embedding, ‘ToneEmb’ for short, or LUT). We implement all models using GCN as the backbone.

Result shown in Table 1 leads to several observations. From the leftmost column, we see low-gain amps seem easier to model than crunch amps, while high-gain amps are the most challenging (i.e., with higher testing losses). Signals from a high-gain amp are highly-distorted, with more high-frequency components that may be harder to be modeled. Similar trends can be seen from other columns.

The middle columns show the result of the one-to-many GCN with FiLM conditioning (‘FiLM-GCN’). Firstly, we see the losses are in general higher than those of the one-to-one non-conditional baseline. This is expected, as we treat the result of the one-to-one model as a performance upperbound, for one-to-one modeling is inherently easier. Among the three variants of FiLM-GCN, ToneEmb with unpaired reference leads to the best result for most amps, reducing greatly the performance gap between the one-to-one approach and the variant with the straightforward LUT-based conditioning representation. While LUT only learns N unique embeddings ϕ , one for each amp, ToneEmb is much more versatile as it computes a unique embedding for each reference wet signal. It seems that the ToneEmb embeddings are thereby information richer, benefiting multi-amp modeling.

For ToneEmb, we initially expect that the advantage of ‘unpaired reference’ over ‘paired reference’ is on zero-shot learning of unseen amps. For seen amps, paired reference simply uses the target wet signal \mathbf{y} as the reference signal \mathbf{z} , providing direct and potentially stronger condition signals. To our surprise, while both ‘ToneEmb (paired)’ and ‘ToneEmb (unpaired)’ greatly outperform the LUT approach, the unpaired approach slightly outperforms the paired approach for most amps. This may be due to the stronger incentive of style/content disentanglement induced by unpaired referencing, as discussed in Section 3.3, but more empirical studies (which we leave as future work) are needed to confirm this.

	non-retrieval	retrieval-based	
	$(\phi^* = \mathcal{E}(z^*))$	nearest	mean
unseen high gain	0.2511	0.2560	0.2593
unseen low gain	0.0338	0.0274	0.0404

Table 2: Efficacy of using different methods for FiLM-GCN (cf. Section 3.4) for zero-shot modeling of two unseen amps, measured again in complex STFT loss.

Finally, Table 1 shows that the ablated version of using concatenation as the conditioning mechanism leads to worse results most of the time. We hence use FiLM-GCN trained with unpaired referencing in experiments below.

4.3 Zero-shot Learning on Unseen Amplifiers

To investigate the potential of the proposed methodology on zero-shot learning, we create wet signals using two “unseen” amplifiers—*High Gain EL34 V2* (high gain) and *Dumble ODS 50* (low gain)—using the clean signals from the test set, and use them as the reference signals z^* for FiLM-GCN, to see whether it can learn the tones zero-shot. Namely, both the content and style are unseen at training time. Here, we evaluate the non-retrieval-based method of using $\phi^* = \mathcal{E}(z^*)$ and the two retrieval-based methods introduced in Section 3.4.

Table 2 shows that the non-retrieval-based method slightly outperforms the other two for the unseen high-gain amp, while nearest-embedding performs the best for the unseen low-gain amp. More importantly, comparing the losses tabulated in Tables 1 and 2, we see that the loss for the unseen low-gain amp is not greatly larger than the loss for the seen low-gain amps, only 1–2 times larger. The loss of the unseen high-gain amp is a bit high, but is only about 2 times larger than those of the seen high-gain amps. We take this as a positive indication of the efficacy of the proposed model in dealing with unseen tones.

Table 2 also shows that mean-embedding performs the worst, adding support of using more versatile embeddings for conditioning. In this regard, the non-retrieval-based method is actually more flexible than nearest-retrieval, as it can compute the reference embedding ϕ^* on-the-fly without referring to a pre-computed presumably large collection of embeddings. Future work can be done to study its effectiveness with more amps (i.e., larger N).

4.4 Case Study on Zero-shot Amp Tone Transfer

To further study the zero-shot scenario, we present finally a case study employing a self-recorded (by one of the authors) guitar solo audio signal with content and tone both unseen at the training time. This recording was captured using a Boss GT-1000 effects processor with a default factory preset based on a high-gain Marshall amplifier setting. The effect chain included not only this unseen amp but also an equalizer (EQ) with a high-cut filter at 10kHz.

The visualization of the spectrograms shown in Figure 5 suggests that the generated result possesses characteris-

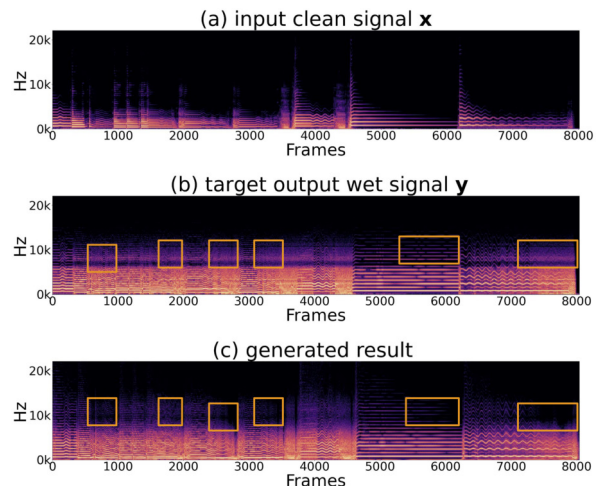


Figure 5: Spectrograms of the input clean signal, target wet signal, and the generated result of the proposed one-to-many FiLM-GCN model, in the zero-shot case study reported in Section 4.4. The orange squares show that our model still struggles to model high-frequency components.

tics similar to those of the target wet signal, but there are notable difference in the high-frequency area. Since we do not consider EQ as a modeling target, our model cannot produce filter-based effects on the signal, resulting in additional harmonics in the spectrum. Furthermore, the sustain of each note is not perfectly reproduced, as the harmonics in the highlighted orange squares are not sequentially connected compared to the case of the target, indicating a struggle to model the high-frequency content accurately.

Despite these limitations, our model can still generate reasonable harmonics according to the input. For the quick string-bending content around frames 6,000 to 7,000, the generated harmonics are correctly damped. Our tone embedding encoder recognizes that the tone of the reference signal is closer to high-gain, empowering the generator to process the input accordingly. We provide audio samples in the supplementary material, including a multi-track recording and a remixed audio created by rendering multiple track separately using our model.

5. CONCLUSION

In this paper, we have presented an end-to-end one-to-many methodology that uses conditions from a tone embedding encoder to emulate multiple guitar amps through a single model, providing empirical evidences of its potential for zero-shot amp modeling. Moving forward, several avenues for future work emerge. First, for more comprehensive audio effect modeling, we can apply our methodology with various configurations of audio effect chains. Second, we can further improve the model’s effectiveness on one-to-many modeling by incorporating more advanced architectures and conditioning mechanisms, such as hypernetwork-based conditioning [9, 38]. Finally, for more effective zero-shot tone transfer, we can train the model on a wider range of amplifier types, which might also pave the way for universal amplifier modeling.

6. ACKNOWLEDGEMENTS

We thank Positive Grid for assistance with the datasets, guitar effect plugins, and for sharing the tone embedding model mentioned in Section 3.1. This work is also partially supported by a grant from the National Science and Technology Council of Taiwan (NSTC 112-2222-E-002-005-MY2).

7. REFERENCES

- [1] E.-P. Damskagg, L. Juvela, E. Thuillier, and V. Välimäki, “Deep learning for tube amplifier emulation,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019.
- [2] A. Wright, E.-P. Damskagg, and V. Välimäki, “Real-time black-box modelling with recurrent neural networks,” in *International Conference on Digital Audio Effects*, 2019.
- [3] A. Wright, E.-P. Damskagg, L. Juvela, and V. Välimäki, “Real-time guitar amplifier emulation with deep learning,” *Applied Sciences*, vol. 10, no. 3, p. 766, 2020.
- [4] M. A. Martínez Ramírez, E. Benetos, and J. D. Reiss, “Deep learning for black-box modeling of audio effects,” *Applied Sciences*, vol. 10, no. 2, p. 638, 2020.
- [5] L. Juvela, E.-P. Damskagg, A. Peussa, J. Mäkinen, T. Sherson, S. I. Mimitakis, K. Rauhanen, and A. Gotsopoulos, “End-to-end amp modeling: from data to controllable guitar amplifier models,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023.
- [6] M. Comunità, C. J. Steinmetz, H. Phan, and J. D. Reiss, “Modelling black-box audio effects with time-varying feature modulation,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023.
- [7] H. Yin, G. Cheng, C. J. Steinmetz, R. Yuan, R. M. Stern, and R. B. Dannenberg, “Modeling analog dynamic range compressors using deep learning and state-space models,” *arXiv preprint arXiv:2403.16331*, 2024.
- [8] Y.-H. Chen, W. Choi, W.-H. Liao, M. A. Martínez Ramírez, K. W. Cheuk, Y. Mitsufuji, J.-S. R. Jang, and Y.-H. Yang, “Improving unsupervised clean-to-rendered guitar tone transformation using GANs and integrated unaligned clean data,” in *International Conference on Digital Audio Effects (DAFx)*, 2024.
- [9] Y.-T. Yeh, W.-Y. Hsiao, and Y.-H. Yang, “Hyper recurrent neural network: Condition mechanisms for black-box audio effect modeling,” in *International Conference on Digital Audio Effects (DAFx)*, 2024.
- [10] C. J. Steinmetz, J. Pons, S. Pascual, and J. Serrà, “Automatic multitrack mixing with a differentiable mixing console of neural audio effects,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021.
- [11] M. A. Martínez Ramírez, W. Liao, C. Nagashima, G. Fabbro, S. Uhlich, and Y. Mitsufuji, “Automatic music mixing with deep learning and out-of-domain data,” in *International Society for Music Information Retrieval (ISMIR)*, 2022.
- [12] J. Koo, M. A. Martínez-Ramírez, W.-H. Liao, S. Uhlich, K. Lee, and Y. Mitsufuji, “Music mixing style transfer: A contrastive learning approach to disentangle audio effects,” *arXiv preprint arXiv:2211.02247*, 2023.
- [13] S. I. Mimitakis, N. J. Bryan, and P. Smaragdis, “One-shot parametric audio production style transfer with application to frequency equalization,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 256–260.
- [14] C. J. Steinmetz, N. J. Bryan, and J. D. Reiss, “Style transfer of audio effects with differentiable signal processing,” *J. Audio Eng. Soc.*, vol. 70, no. 9, pp. 708–721, 2022.
- [15] A. Wright and V. Valimaki, “Neural modeling of phaser and flanging effects,” *Journal of the Audio Engineering Society*, vol. 69, no. 7/8, pp. 517–529, 2021.
- [16] A. Wright, V. Välimäki, and L. Juvela, “Adversarial guitar amplifier modelling with unpaired data,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023.
- [17] C. J. Steinmetz and J. D. Reiss, “Efficient neural networks for real-time analog audio effect modeling,” in *152nd Audio Engineering Society Convention*, 2022.
- [18] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, “A simple framework for contrastive learning of visual representations,” in *International Conference on Machine Learning*, 2020, pp. 1597–1607.
- [19] C. Deng, C. Yu, H. Lu, C. Weng, and D. Yu, “PitchNet: Unsupervised singing voice conversion with pitch adversarial network,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 7749–7753.
- [20] N. Takahashi, M. K. Singh, and Y. Mitsufuji, “Hierarchical disentangled representation learning for singing voice conversion,” in *International Joint Conference on Neural Networks (IJCNN)*, 2021, pp. 1–7.
- [21] S. Liu, Y. Cao, N. Hu, D. Su, and H. Meng, “FastSVC: Fast cross-domain singing voice conversion with feature-wise linear modulation,” in *IEEE International Conference on Multimedia and Expo (ICME)*, 2021, pp. 1–6.

- [22] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville, “FiLM: Visual reasoning with a general conditioning layer,” in *AAAI Conference on Artificial Intelligence*, 2018.
- [23] L. Zhang, C. Yu, H. Lu, C. Weng, C. Zhang, Y. Wu, X. Xie, Z. Li, and D. Yu, “DurIAN-SC: Duration informed attention network based singing voice conversion system,” in *International Speech Communication Association (INTERSPEECH)*, 2020, pp. 1231–1235.
- [24] H. Guo, H. Lu, N. Hu, C. Zhang, S. Yang, L. Xie, D. Su, and D. Yu, “Phonetic posteriorgrams based many-to-many singing voice conversion via adversarial training,” *arXiv preprint arXiv:2012.01837*, 2020.
- [25] H. Guo, Z. Zhou, F. Meng, and K. Liu, “Improving adversarial waveform generation based singing voice conversion with harmonic signals,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 6657–6661.
- [26] X. Li, S. Liu, and Y. Shan, “A hierarchical speaker representation framework for one-shot singing voice conversion,” *International Speech Communication Association (INTERSPEECH)*, pp. 4307–4311, 2022.
- [27] J.-T. Wu, J.-Y. Wang, J.-S. R. Jang, and L. Su, “A unified model for zero-shot singing voice conversion and synthesis,” in *International Society for Music Information Retrieval Conference (ISMIR)*, 2022.
- [28] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, “Unsupervised learning of visual features by contrasting cluster assignments,” *Advances in Neural Information Processing Systems*, pp. 9912–9924, 2020.
- [29] M. Assran, M. Caron, I. Misra, P. Bojanowski, F. Bordes, P. Vincent, A. Joulin, M. Rabbat, and N. Ballas, “Masked siamese networks for label-efficient learning,” in *European Conference on Computer Vision*. Springer, 2022, pp. 456–473.
- [30] J. Spijkervet and J. A. Burgoyne, “Contrastive learning of musical representations,” in *International Society for Music Information Retrieval (ISMIR)*, 2021.
- [31] Y. Gong, Y.-A. Chung, and J. Glass, “AST: Audio Spectrogram Transformer,” in *International Speech Communication Association (INTERSPEECH)*, 2021, pp. 571–575.
- [32] H. Yakura, K. Watanabe, and M. Goto, “Self-supervised contrastive learning for singing voices,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 1614–1623, 2022.
- [33] H. Zhao, C. Zhang, B. Zhu, Z. Ma, and K. Zhang, “S3T: Self-supervised pre-training with Swin Transformer for music classification,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 606–610.
- [34] M. Rice, C. J. Steinmetz, G. Fazekas, and J. D. Reiss, “General purpose audio effect removal,” in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2023.
- [35] P. Sobot, “Pedalboard,” 2021, [Online] <https://github.com/spotify/pedalboard>.
- [36] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *International Conference on Learning Representations (ICLR)*, 2015.
- [37] L. Van der Maaten and G. Hinton, “Visualizing data using t-SNE,” *Journal of Machine Learning Research*, vol. 9, no. 11, 2008.
- [38] A. Richard, D. Markovic, I. D. Gebru, S. Krenn, G. A. Butler, F. Torre, and Y. Sheikh, “Neural synthesis of binaural speech from mono audio,” in *Proc. Int. Conf. Learning Representations*, 2021.

MEL-ROFORMER FOR VOCAL SEPARATION AND VOCAL MELODY TRANSCRIPTION

Ju-Chiang Wang Wei-Tsung Lu Jitong Chen

ByteDance, San Jose, CA, USA

ju-chiang.wang@bytedance.com

ABSTRACT

Developing a versatile deep neural network to model music audio is crucial in MIR. This task is challenging due to the intricate spectral variations inherent in music signals, which convey melody, harmonics, and timbres of diverse instruments. In this paper, we introduce Mel-RoFormer, a spectrogram-based model featuring two key designs: a novel Mel-band Projection module at the front-end to enhance the model’s capability to capture informative features across multiple frequency bands, and interleaved RoPE Transformers to explicitly model the frequency and time dimensions as two separate sequences. We apply Mel-RoFormer to tackle two essential MIR tasks: vocal separation and vocal melody transcription, aimed at isolating singing voices from audio mixtures and transcribing their lead melodies, respectively. Despite their shared focus on singing signals, these tasks possess distinct optimization objectives. Instead of training a unified model, we adopt a two-step approach. Initially, we train a vocal separation model, which subsequently serves as a foundation model for fine-tuning for vocal melody transcription. Through extensive experiments conducted on benchmark datasets, we showcase that our models achieve state-of-the-art performance in both vocal separation and melody transcription tasks, underscoring the efficacy and versatility of Mel-RoFormer in modeling complex music audio signals.

1. INTRODUCTION

Modeling musical audio signals with deep neural networks (DNNs) for MIR tasks has emerged as a vibrant and promising area of research [1–5]. Most of such DNN models are built upon the spectrogram, a fundamental frequency-time representation of audio signals. Traditional approaches typically treat the spectrogram as a sequence of spectra over time, with the frequency axis representing the feature dimension. However, recent advancements have encompassed explicit modeling of the frequency dimension as a sequence in their architecture designs [5–9], recognizing its rich semantic information in music audio

signals, including melody, harmonics, and instrument timbres. These architectures have showcased state-of-the-art performance in various MIR tasks such as vocal melody extraction [5], section segmentation [10], instrument transcription [8], and music source separation [11], leveraging the model’s ability to discern spectral patterns effectively.

The Transformer architecture [12] has demonstrated remarkable efficacy not only in Natural Language Processing but also in various MIR tasks, where it excels at modeling sequences to predict high-level musical semantics such as tags, beats, chords, sections, and notes [5, 8–10, 13–15]. However, its potential to accurately predict low-level audio signals remained uncertain. Lu et al. proposed a novel architecture, called BS-RoFormer [11], to tackle the task of music source separation (MSS), which aims to separate audio recordings into musically distinct sources such as vocals, bass, and drums [16, 17]. Inspired by the Band-Split RNN (BSRNN) model [7], BS-RoFormer adopts the interleaved sequence modeling, treating time and frequency dimensions as two separate sequences. Notably, it replaces Recurrent Neural Networks (RNNs) with Transformer encoders, demonstrating exceptional performance. This was evident in its first-place ranking and substantial margin of performance gain over the runner-up in the Music Separation track of the Sound Demixing Challenge 2023 (SDX’23) [18].

Another key attribute contributing to the success of BS-RoFormer is the band-split module at the front-end. Traditional Transformer-based models typically rely on a Convolutional Neural Network (CNN) front-end to extract features from the spectrogram for the succeeding Transformer blocks (e.g., [5, 13, 19]). However, CNNs are not inherently designed to model two spectral events that are far apart in frequency, which could limit the model to characterize detailed spectral patterns. In contrast, the band-split module divides the frequency dimension into a number of subbands and employs multi-layer perceptrons (MLPs) to directly project the raw subband spectrograms into a sequence of band-wise features for the succeeding Transformer to model it along the frequency axis. From another perspective, the band-split mechanism can be seen as a set of learnable band-pass filters, underscoring the importance of designing an effective band-division scheme.

In this paper, we introduce *Mel-RoFormer*, which is a successor of BS-RoFormer with an enhanced band-division scheme that leverages the Mel-scale [20] to improve the model’s generalization ability. The Mel-scale is



© J.-C. Wang, W.-T. Lu, and J. Chen. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).
Attribution: J.-C. Wang, W.-T. Lu, and J. Chen, “Mel-RoFormer for Vocal Separation and Vocal Melody Transcription”, in *Proc. of the 25th Int. Society for Music Information Retrieval Conf.*, San Francisco, United States, 2024.

engineered to mimic the non-linear perception of sound by the human ear, exhibiting higher discrimination at lower frequencies and reduced discrimination at higher frequencies. It has a long history as a reference for designing acoustic features [21] such as MFCC and mel-spectrogram in audio signal processing. We develop the Mel-band mapping based on the Mel-scale, resulting in a band-division that generates overlapping subbands in frequency. In contrast, the band-split scheme in BS-RoFormer is defined empirically and produces non-overlapping subbands.

Mel-RoFormer is applied to address two fundamental MIR tasks: vocal separation and vocal melody transcription, which involve isolating singing voices from audio mixtures and transcribing their lead melodies, respectively. Mel-RoFormer demonstrates superior performance compared to BS-RoFormer and other MSS models in experiments. For vocal melody transcription, we propose a two-step approach instead of training a unified model. We first pretrain a vocal separation model and then fine-tune it for vocal melody transcription. The resulting model achieves state-of-the-art performance across all metrics and exhibits strong robustness in detecting note offsets, which is considered to be the most challenging aspect of the task [22].

Readers can refer to the open-sourced implementation¹ and configurations² for Mel-RoFormer and its variants.

2. RELATED WORK

To address the capabilities of DNN models that pay attention to modeling the frequency dimension for MIR tasks, SpecTNT is one of early successful attempts. Central to SpecTNT’s architecture is the TNT block, where two Transformer encoders are strategically arranged to model along both frequency and time axes. A novel concept introduced by SpecTNT is the Frequency Class Token (FCT), which serves to bridge of the two Transformers, enabling the interchangeability of embeddings across both axes within the TNT block. However, SpecTNT relies on CNNs at the front-end, and the FCT is obtained through aggregating features from the frequency sequence, potentially leading to information loss. Nonetheless, SpecTNT has showcased remarkable performance across various MIR tasks such as beat tracking [14], chord recognition [5], structure segmentation [10], and vocal melody estimation [5]. Its successor, Perceiver TF [8], is designed to enhance efficiency while demonstrating outstanding performance in multitrack instrument/vocal transcription tasks.

Moving on to MSS, a key MIR task that has significantly benefited from DNNs, approaches typically span frequency-domain and time-domain methodologies. The benchmark MUSDB18 dataset [23] offers 4-stem sources including vocals, bass, drums, and others, adhering to the definition established by the 2015 Signal Separation Evaluation Campaign (SiSEC) [24]. Frequency-domain approaches rely on spectrogram-based representations as input, leveraging models such as fully connected neural net-

works [25], CNNs [26–28], and RNNs [29] to achieve separation. Conversely, time-domain approaches such as Wave-U-Net [30], ConvTasNet [31], and Demucs [32] construct their DNNs directly on waveform inputs. Recently, Hybrid Transformer Demucs (HTDemucs) [33] has proposed a novel approach, utilizing a cross-domain Transformer to amalgamate both frequency- and time-domain models, showcasing promising potential in this field. However, none of the mentioned approaches employ Transformers to model the inter-context of frequency and time as two separate sequences.

The output of vocal melody transcription is a sequence of non-overlapping notes, each comprising onset and offset times along with a pitch key, assuming the melody is monophonic. Due to limited training data availability, only a few studies have focused on transcribing note-level outputs from polyphonic music audio, underscoring the significance of pre-training [34] or semi-supervised [35] techniques. Recently, Wang et al. released a human-annotated dataset comprising 500 Chinese songs [36], along with a baseline CNN-based model. Donahue et al. [34] propose leveraging pre-trained representations from Jukebox [37] to enhance melody transcription, primarily focusing on lead instruments such as synthesizers, guitars, piano, and vocals. They curate a dataset of 50 hours of melody transcriptions sourced from crowdsourced annotations. However, clarity regarding the quality and identification of vocal melody annotations within songs remains lacking. In [35], a teacher-student training scheme is introduced to leverage pseudo labels derived from fundamental frequency (F0) estimations of vocals. On the other hand, [38] presents a system that necessitates a vocal separation as a front-end. In our approach, we employ vocal separation as a pre-trained model for fine-tuning a specialized model, which can be more efficient and task-optimal.

3. MODEL

Figure 1 illustrates the diagram of Mel-RoFormer, consisting of three major modules: Mel-band Projection, RoFormer blocks, and Embedding Projection. Subsequent subsections will delve into the Mel-band Projection and Embedding Projection modules, while readers can find further details on the RoFormer blocks in [11], where they are referred to as “RoPE Transformer blocks.”

Mel-RoFormer takes the input of a complex spectrogram X with dimensions $(C \times F \times T)$, where C , F , and T denote the number of channels, frequency bins, and time steps, respectively. This frequency-time representation X is typically obtained via a short-time Fourier transform (STFT), encompassing both real and imaginary parts. In stereo mode, C is defined as $2 \times 2 = 4$, reflecting the presence of two channels for real and imaginary spectrograms. The output of Mel-RoFormer is denoted by Y , with dimensions $(Z \times T)$, where Z and T represent the number of output features and time steps, respectively. One can treat Y as the feature matrix over time. Depending on the downstream tasks, appropriate values for Z can be set, and Y can be rearranged accordingly, as detailed in Section 4.

¹ <https://github.com/lucidrains/BS-RoFormer>

² <https://github.com/ZFTurbo/Music-Source-Separation-Training>

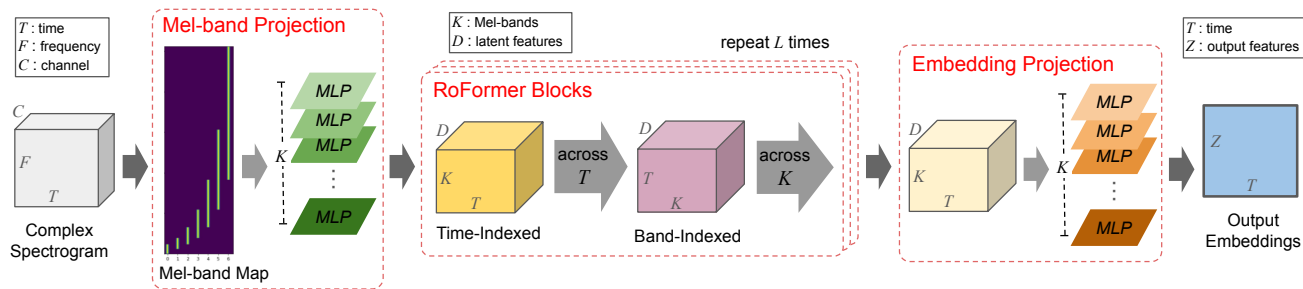


Figure 1. The diagram of Mel-RoFormer, which consists of three major modules: Mel-band Projection, RoFormer Blocks, and Embedding Projection. The input is a Complex Spectrogram, and the output is an Embedding tensor, which can be rearranged into the desired shape.

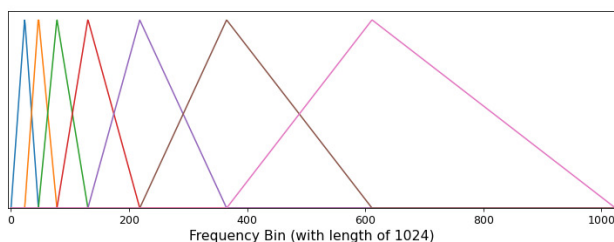


Figure 2. Illustration of Mel filter-bank with 7 bands. In this example, the length of frequency bins is 1024. Here, the frequency bins from 1 to 46 are encompassed by the 0-th Mel-band (i.e. \mathcal{F}_0), those from 24 to 77 are encompassed by the 1-th Mel-band (i.e. \mathcal{F}_1), and so forth.

3.1 Mel-band Projection Module

The Mel-band Projection module comprises a frequency-to-Mel-band mapping and a set of K multi-layer perceptrons (MLPs). Each subband of X is denoted as X_k with dimensions $(C \times |\mathcal{F}_k| \times T)$, where $\mathcal{F}_k \in \{0, 1, \dots, F-1\}$ represents the indices of frequency bins for the k -th subband, and F is the length of frequency bins.

The frequency-to-Mel-band mapping, represented as $\{\mathcal{F}_k\}_{k=0}^{K-1}$, stems from the Mel filter-bank, as depicted in Fig. 2, where triangular-shaped filters are centered at different Mel frequencies on the Mel-scale. The indices of non-zero values of a filter correspond to the frequency bins of the respective Mel-band. Mathematically, the Mel-scale follows a quasi-logarithmic function of acoustic frequency, ensuring that perceptually similar pitch intervals (e.g., octaves) possess equal width across the entire audible range. The width of a Mel-band (i.e., $|\mathcal{F}_k|$) is twice the distance between its center and the center of the preceding Mel-band. Consequently, the latter half of a Mel-band overlaps with its subsequent Mel-band, and so forth, until reaching the final Mel-band. In contrast, due to the band-split design in BS-RoFormer [11], the frequency ranges of different subbands do not overlap. The Mel-band Map depicted in Fig. 1 exemplifies the frequency-to-Mel-band mapping, illustrating a binary relationship between 1024 frequency bins and 7 Mel-bands.

The input X_k is rearranged into a shape of $(C|\mathcal{F}_k| \times T)$ for the MLP layer. The k -th MLP, denoted as Λ_k , com-

prises an RMSNorm layer [39] followed by a linear layer. The linear layer transforms from $C|\mathcal{F}_k|$ dimensions to D dimensions, where D is the number of latent features. The resulting outputs $\{\Lambda_k(X_k)\}_{k=0}^{K-1}$ are stacked to form a shape of $(D \times K \times T)$, serving as input to the subsequent RoFormer blocks. The Mel-band Projection module can be conceptualized as a learnable Mel filter-bank, with the MLP layer functioning as the mechanism to learn the filters. This grants the model greater flexibility in determining the optimal shapes for different filters, without being confined to predefined filter designs such as triangle-shaped ones (see Fig. 2).

3.2 RoFormer Blocks

The RoFormer blocks consist of a stack of L interleaved RoPE Transformer encoders [40]. The interleaved sequence modeling processes the data across time (T) and subband (K) dimensions alternately. In BS-RoFormer [11], the authors observed that Rotary Position Encoding (RoPE) played a crucial role in enhancing Transformer performance compared to using traditional absolute position encoding. It is suggested that RoPE aids in preserving the positional information within the sequence, making it invariant to repetitive processes of rearrangement.

Specifically, the data is first rearranged into a time-indexed shape of $(DK \times T)$, allowing modeling across time. Subsequently, it is rearranged into a band-indexed shape of $(DT \times K)$, facilitating modeling across subbands. The former step treats the data as a time sequence, while the latter treats it as a subband sequence. By repeating this process, information from different time steps and subbands becomes interchangeable, thereby enhancing the model’s ability to generalize.

3.3 Embedding Projection Module

The Embedding Projection module plays a crucial role in generating suitable embeddings necessary for various downstream tasks. It has been observed that utilizing MLPs can lead to more effective mask estimation compared to using plain linear layers in source separation tasks [41]. Our preliminary investigation also suggests that removing this module can result in unstable training.

This module comprises K individual MLPs, denoted as Φ_k , each containing an RMSNorm layer, a linear layer followed by a Tanh activation, and another linear layer followed by a gated linear unit (GLU) layer [42]. The first linear layer transforms from D to $4D$ dimensions, while the subsequent linear layer with GLU transforms from $4D$ to a desired length Z_k . Here, Z_k specifies the number of output features for Φ_k based on its specific purpose, as elaborated in Section 4. All the MLP outputs are concatenated along the feature dimension, resulting in a final output shape of $(Z \times T)$, where $Z = \sum_k Z_k$.

4. DOWNSTREAM TASKS

This section details the application of Mel-RoFormer tailored for vocal separation and vocal melody transcription.

4.1 Vocal Separation

For vocal separation, the Embedding Projection estimates the mask for the complex spectrogram. Each MLP Φ_k is designed to align its output shape with that of the corresponding input Mel-band complex spectrogram, with Z_k set as $C|\mathcal{F}_k|$. The output Y is rearranged into

$$[\hat{\Phi}_0, \hat{\Phi}_1, \dots, \hat{\Phi}_{(K-1)}], \quad (1)$$

with dimensions $(C \times \hat{Z} \times T)$, where $\hat{\Phi}_k$ with a shape of $(|\mathcal{F}_k| \times T)$ is the output corresponding to the k -th Mel-band, and $\hat{Z} = \sum_k |\mathcal{F}_k|$. Because adjacent Mel-bands overlap, the estimated mask values of the overlapping frequency bins are averaged:

$$\hat{M}[c, f, t] = \frac{1}{S_f} \sum_k \hat{\Phi}_k[c, f, t], \quad (2)$$

where \hat{M} represents the estimated mask, c , f , and t are the indices of channel, frequency bin, and time step, respectively, and S_f is the count of the overlapping frequency bins. The estimated mask \hat{M} has the same shape $(C \times F \times T)$ as that of the input X , encompassing both the real and imaginary parts of the complex spectrogram.

We utilize complex Ideal Ratio Masks (cIRMs) [43] as our optimization goal for the vocal separation model. The estimated mask \hat{M} derived from Embedding Projection can serve as the cIRMs. The separated complex spectrogram \hat{Y} is obtained by element-wise multiplication of the cIRM with the input complex spectrogram: $\hat{Y} = \hat{M} \odot X$. Subsequently, an inverse STFT (iSTFT) is applied to \hat{Y} to reconstruct the separated signal \hat{y} in the time-domain.

Let ψ denote the target time-domain signal, and $\Psi^{(w,r)}$ denote the corresponding complex Spectrogram using a window size w and time-resolution r for STFT. We employ the mean absolute error (MAE) loss to train the cIRMs \hat{M} . Specifically, the objective loss encompasses both the time-domain MAE and the multi-resolution complex spectrogram MAE [44]:

$$\mathcal{L} = \|\psi - \hat{y}\| + \sum_{w \in W, r \in R} \|\Psi^{(w,r)} - \hat{Y}^{(w,r)}\|, \quad (3)$$

where the configurations for multi-resolution STFTs cover 5 window sizes with $W = \{4096, 2048, 1024, 512, 256\}$, and 2 resolutions with $R = \{100, 300\}$ frames per second.

4.2 Vocal Melody Transcription

Instead of starting training from scratch, we utilize a pre-trained vocal separation Mel-RoFormer and fine-tune it for vocal melody transcription. Given that the Embedding Projection module in the pre-trained vocal separation model functions as a mask estimator, it might inherently possess biases towards signal-level semantics. Therefore, we opt to replace the pre-trained Embedding Projection with a newly initialized one, with a modification specifically on the output dimension. To ensure equal contribution from each Mel-band to the feature dimension, we set a uniform value of 64 for all Z_k 's. The resulting embeddings take the shape of $(64K \times T)$, with a $64K$ -dimensional feature vector for each time step.

We adopt the "onsets and frames" approach [45], employing two frame-wise predictors: an onset predictor and a frame predictor, both of which receive embeddings from Mel-RoFormer. The onset predictor identifies the onset event of a pitched note, while the frame predictor determines the continuation of a pitched note. This design facilitates a post-processing method where the initiation of a new note is determined only if the onset predictor indicates the start of a pitch, and simultaneously, the frame predictor confirms the presence of an onset for that pitch within the succeeding frames. The onset and frame predictors operate at a time-resolution of 50 frames per second. In cases where the Mel-RoFormer embeddings do not match this time-resolution, we employ 1-D adaptive average pooling over the time dimension.

We employ an MLP layer for the onset predictor, consisting of a linear layer followed by a Rectified Linear Unit (ReLU), dropout with a rate of 0.5, and another linear layer. The linear layer has 512 hidden channels, and the output dimension is set to 60, representing 60 supported pitches. For the frame predictor, a single linear layer is utilized, outputting 61 pitch classes, with one indicating non-pitch. Binary cross-entropy losses of the two predictors are summed as the final loss to train the entire model. The thresholds for the onset and frame predictors are set at 0.45 and 0.25, respectively.

5. EXPERIMENT

Our experiments cover vocal separation and vocal melody transcription. In the vocal separation evaluation, we train and test two types of models: the first on 44.1kHz stereo audio recordings, and the second on 24kHz mono audio recordings. Next, we utilize the model trained on 24kHz mono audio recordings as the pre-trained model for fine-tuning in the vocal melody transcription evaluation.

5.1 Datasets

Table 1 overviews the datasets used in this study. We use four public datasets for evaluation. The data splitting ad-

Dataset	Task	Songs	Split
MUSDB18HQ [23]	sepa	150	train 100, test 50
MoisesDB [46]	sepa	240	train 200, val 40
MIR-ST500 [36]	trans	500	train 330, val 37, test 98
POP909 [47]	trans	909	train 750, val 50, test 109
In-House	sepa	1533	train 1433, val 100

Table 1. Summary of the datasets used in this study. Abbreviations: ‘sepa’: vocal separation, ‘trans’: vocal melody transcription, ‘val’: validation.

heres to the official guidelines of each dataset, except for POP909, where songs with IDs ranging from 801 to 909 are reserved for testing. The ‘Split’ column of Table 1 indicates the numbers of songs allocated for training, validation, and testing. All data for separation tasks are stereo recordings with a sampling rate of 44.1kHz, and stem-level recordings are pre-mixed into four stems: vocals, bass, drums, and other. The audio of transcription data was re-sampled to mono with a 24kHz sampling rate to follow the conventional setting [38]. Although access to some songs in MIR-ST500 was restricted, our test set, comprising 98 songs, closely resembles the original setting of 100 songs.

5.2 Configuration for Vocal Separation

To obtain the frequency-to-Mel-band mapping, we employ the Mel filter-bank implementation in librosa [48], which emulates the behavior of the function in MATLAB Auditory Toolbox [49]. By using `librosa.filters.mel`, we acquire the mapping matrix comprising a triangle filter for each Mel-band. Subsequently, we binarize this matrix by setting all non-zero values to 1, thereby discarding the triangle filters. This process yields the Mel-band Map depicted in Figure 1.

We follow the method outlined in [11] for performing random remixing data augmentation. This strategy involves cross-dataset stem-level combination, resulting in a significantly larger number of examples than the original size of the datasets combined.

Tree evaluation scenarios are considered: ① *musdb18-only*: train a 44.1kHz stereo model only on the MUSDB18-HQ training set; ② *all-data*: all additional data, including MUSDB18HQ, MoisesDB, and In-House, are used to train a 44.1kHz stereo model; ③ *musdb18+moisesdb*: MUSDB18HQ and MoisesDB are resampled to train a 24kHz mono model, serving as the pre-trained model for fine-tuning for the melody transcription task.

Our main baseline is BS-RoFormer [11]. For scenarios ① and ②, we set the parameters as follows: $T=800$ (8-second chunk), $K=60$, $D=384$, $L=12$, and a window size of 2048 and a hop size of 441 for STFT. In scenario ③, two models are trained: *24k-small* and *24k-large*. The small model uses $T=300$ (6-second chunk), $K=32$, $D=128$, $L=12$; while the large model uses $T=300$ (6-second chunk), $K=32$, $D=256$, $L=24$. Both models adopt a window size of 1024 and a hop size of 480 for STFT. All the separation models use the “overlap & average” de-framing method [11] with a hop of half a chunk. These

Model	Vocals	# Param
HDemucs [52] [†]	8.04	-
Sparse HT Demucs [33] [†]	9.37	-
BSRNN [7] [†]	10.01	-
TFC-TDF-UNet-V3 [53] [†]	9.59	-
BS-RoFormer ①	11.49	93.4M
Mel-RoFormer ①	12.08	105M
BS-RoFormer ② [†]	12.82	93.4M
Mel-RoFormer ② [†]	13.29	105M

Tested with resampled 24kHz mono audio

BS-RoFormer (24k-small) ③	10.56	8.0M
Mel-RoFormer (24k-small) ③	11.01	9.1M
BS-RoFormer (24k-large) ③	12.19	48.4M
Mel-RoFormer (24k-large) ③	12.69	50.7M

Symbol † indicates models trained with extra data. Symbols ①, ②, and ③ indicate the three evaluation scenarios.

Table 2. Result (in SDR) on MUSDB18HQ test set.

above mentioned settings remain consistent between Mel-RoFormer and BS-RoFormer.

For training, we utilize the AdamW optimizer [50] with a learning rate (LR) of 0.0005. The LR is reduced by 10% every 40k steps. To optimize GPU memory usage, we employ flash-attention [51] and mixed precision. Specifically, the STFT and iSTFT modules use FP32, while all other components use FP16. Regarding hardware configurations, we employ different setups for each scenario. In scenario ①, 8 Nvidia A-100-80GB GPUs with `batch_size=64` are used, and the training stopped at 400K steps (~40 days). For scenario ②, 16 Nvidia A100-80GB GPUs with `batch_size=128` are utilized, and the training halted at 1M steps (~93 days). In scenario ③, 16 Nvidia V100-32GB GPUs with `batch_size=96` are used, and the training stopped at 500K steps (~31 days).

The reason to use a large number of training steps is driven by the continuous improvement observed in the model’s performance, coupled with the absence of overfitting. This can be attributed to two main factors: the effect of the random remixing augmentation and the inherent capability of the model itself. These factors contribute to the model’s ability to continuously learn and adapt to the training data, resulting in sustained performance improvements without encountering overfitting issues.

5.3 Result for Vocal Separation

Table 2 presents the results, with the signal-to-distortion ratio (SDR) values [54] computed by `museval` [55] as the evaluation metric. The median SDR across the median SDRs over all 1-second chunks of each test song is reported, following prior conventions. Several representative existing models are included for comparison.

From the result, we see that Mel-RoFormer achieve state-of-the-art performance. It is evident that Mel-band Projection significantly enhances vocal separation performance, leading to a consistent improvement over BS-

Model	#Param	COn	COnP	COnPOff
Efficient-b1 [36]	-	.754	.666	.458
JDC _{note} [35]	-	.762	.697	.422
A-VST [56]	-	.783	.707	.538
Perceiver TF [8]	-	-	.777	-
MERT [15] ④	324M	.775	.751	.530
SpecTNT [5] ④	8.4M	.801	.778	.550
Mel-RoF-small ④	14.5M	.807	.786	.609
Mel-RoF-large ④	64.6M	.819	.798	.625
Mel-RoF-small ①	14.5M	.780	.765	.574
Mel-RoF-large ①	64.6M	.790	.776	.594

Symbols ④, ⑤, and ① indicate three evaluation scenarios.

Table 3. Model comparison on MIR-ST500 test set.

Model	COn	COnP	COnPOff
MERT [15] ⑤	.745	.697	.315
SpecTNT [5] ⑤	.797	.775	.371
Mel-RoF-small ⑤	.831	.805	.398
Mel-RoF-large ⑤	.869	.842	.486
Mel-RoF-small ①	.833	.808	.405
Mel-RoF-large ①	.864	.839	.494

Table 4. Model comparison on POP909 test set. Evaluated with a time tolerance of 80 ms.

RoFormer, with an average gain of 0.5 dB across all scenarios. This showcases the effectiveness of the Mel-band mapping scheme in capturing human voices. Qualitative analysis indicates that Mel-RoFormer produces smoother vocal sounds with more consistent loudness. On the other hand, the 24kHz mono model also performs admirably, which bodes well for downstream tasks like vocal melody transcription, as they do not necessitate high-quality audio with high sampling rates. Furthermore, the smaller model with 9.1M parameters achieves over 11 dB, demonstrating its potential for resource-constrained environments.

5.4 Configuration for Vocal Melody Transcription

Three evaluation scenarios are studied: ④ trained on MIR-ST500; ⑤ trained on POP909, and ① trained on a combination of MIR-ST500 and POP909. With $T=300$ and $K=32$ for Mel-RoFormer, the resulting embedding matrix for the onset and frame predictors has a shape of (2048×300) , representing a 6-second input with a frame rate of 50Hz. We also consider a variant that is trained from scratch without a pre-trained model.

For fine-tuning, we use the AdamW optimizer with a LR of 0.001 for the onset and frame predictors and 0.0001 for the Mel-RoFormer module. The LR is reduced by 10% with a patience of 15 epochs, where an epoch comprises 100 steps. The best model is selected based on validation performance for testing.

For baselines, we implement SpecTNT [5] and MERT [15]. SpecTNT features 5-layer TNT blocks and is trained with the random mixing augmentation method outlined in [8]. For MERT, we use the pre-trained weights of “MERT-

v1-330M,” where the model accepts 5-second input with a 24kHz audio sampling rate and produces embeddings with a shape of (1024×375) . All mentioned models are trained or fine-tuned using 8 Nvidia V100-32GB GPUs.

Evaluation metrics include the F-measures of Correct Onset (COn), Correct Onset and Pitch (COnP), and Correct Onset, Pitch, and Offset (COnPOff). These metrics are computed using `mir_eval` [57], with a pitch tolerance of 50 cents and a time tolerance of 50 ms. For POP909, we adjust the time tolerance to 80ms due to the less precise nature of note onsets and offsets in its labeling method [47].

5.5 Result for Vocal Melody Transcription

Tables 3 and 4 display the results on MIR-ST500 and POP909, respectively, featuring a comparison with various existing models. It is worth noting that fine-tuning a pre-trained model typically converges in fewer than 15K steps, while training from scratch requires significantly more steps (e.g., 50k steps) and yields significantly inferior performance compared to fine-tuned models.

Several key observations emerge from the results. Firstly, our proposed model (Mel-RoF-large and Mel-RoF-small) achieves state-of-the-art performance across all metrics on both MIR-ST500 and POP909, showcasing its effectiveness. Particularly noteworthy is the substantial performance improvement in COnPOff over the baselines (e.g., a 7.5 percentage point increase in Mel-RoF-large compared to SpecTNT), highlighting the robustness of Mel-RoFormer in accurately determining full notes, including onset, pitch, and offset. This can be attributed to its capability to extract clean singing voices, thereby minimizing the influence of irrelevant instruments. Comparing our models to MERT underscores the superiority of Mel-RoFormer, owing to its architectural design and pre-training with a relevant task. This emphasizes the importance of explicitly modeling frequency with Transformers and suggests that the separation task can be a valuable objective when training a foundation model.

Cross-comparing MIR-ST500 and POP909, we note that annotations are relatively more consistent in MIR-ST500. In contrast, POP909 exhibits errors primarily at note offsets, along with global time shifts in several songs. Consequently, we accept a larger time tolerance of 80 ms in the evaluation scenario. Particularly, in scenario ①, training with combined datasets improves test performance on POP909 but degrades that of MIR-ST500, consistent with our observations about labeling quality.

6. CONCLUSION

We have presented the Mel-RoFormer model, which integrates the Mel-band Projection scheme to enhance its ability to model musical signals effectively. Our experiments have shown highly promising results in vocal separation and vocal melody transcription. These findings suggest the potential of Mel-RoFormer as a foundation model for various other MIR tasks, including chord recognition and multi-instrument transcription [58].

7. REFERENCES

- [1] S. Dieleman and B. Schrauwen, “End-to-end learning for music audio,” in *ICASSP*, 2014, pp. 6964–6968.
- [2] R. M. Bittner, J. J. Bosch, D. Rubinstein, G. Meseguer-Brocail, and S. Ewert, “A lightweight instrument-agnostic model for polyphonic note transcription and multipitch estimation,” in *ICASSP*, 2022, pp. 781–785.
- [3] K. Choi, G. Fazekas, and M. Sandler, “Automatic tagging using deep convolutional neural networks,” in *ISMIR*, 2016.
- [4] M. Won, S. Chun, O. Nieto, and X. Serra, “Data-driven harmonic filters for audio representation learning,” in *ICASSP*, 2020, pp. 536–540.
- [5] W.-T. Lu, J.-C. Wang, M. Won, K. Choi, and X. Song, “SpecTNT: A time-frequency transformer for music audio,” in *ISMIR*, 2021.
- [6] A. Zadeh, T. Ma, S. Poria, and L.-P. Morency, “Wild-mix dataset and spectro-temporal transformer model for monoaural audio source separation,” *arXiv preprint arXiv:1911.09783*, 2019.
- [7] Y. Luo and J. Yu, “Music Source Separation With Band-Split RNN,” *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 31, pp. 1893–1901, 2023.
- [8] W.-T. Lu, J.-C. Wang, and Y.-N. Hung, “Multitrack music transcription with a time-frequency perceiver,” in *ICASSP*, 2023.
- [9] K. Toyama, T. Akama, Y. Ikemiya, Y. Takida, W.-H. Liao, and Y. Mitsufuji, “Automatic piano transcription with hierarchical frequency-time transformer,” in *ISMIR*, 2023.
- [10] J.-C. Wang, Y.-N. Hung, and J. B. Smith, “To catch a chorus, verse, intro, or anything else: Analyzing a song with structural functions,” in *ICASSP*, 2022, pp. 416–420.
- [11] W.-T. Lu, J.-C. Wang, Q. Kong, and Y.-N. Hung, “Music source separation with Band-Split RoPE Transformer,” *arXiv preprint arXiv:2309.02612*, 2023.
- [12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” *NeurIPS*, vol. 30, 2017.
- [13] M. Won, K. Choi, and X. Serra, “Semi-supervised music tagging transformer,” in *ISMIR*, 2021, pp. 769–776.
- [14] Y.-N. Hung, J.-C. Wang, X. Song, W.-T. Lu, and M. Won, “Modeling beats and downbeats with a time-frequency transformer,” in *ICASSP*, 2022, pp. 401–405.
- [15] L. Yizhi, R. Yuan, G. Zhang, Y. Ma, X. Chen, H. Yin, C. Xiao, C. Lin, A. Ragni, E. Benetos *et al.*, “Mert: Acoustic music understanding model with large-scale self-supervised training,” in *ICLR*, 2023.
- [16] Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimilakis, D. FitzGerald, and B. Pardo, “An overview of lead and accompaniment separation in music,” *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 26, no. 8, pp. 1307–1335, 2018.
- [17] Y. Mitsufuji, G. Fabbro, S. Uhlich, F.-R. Stöter, A. Défossez, M. Kim, W. Choi, C.-Y. Yu, and K.-W. Cheuk, “Music demixing challenge 2021,” *Frontiers in Signal Processing*, 2022.
- [18] G. Fabbro, S. Uhlich, C. Lai, W. Choi, M. Martinez-Ramirez, W. Liao, I. Gadelha, G. Ramos, E. Hsu, H. Rodrigues *et al.*, “The sound demixing challenge 2023—music demixing track,” *arXiv preprint arXiv:2308.06979*, 2023.
- [19] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu *et al.*, “Conformer: Convolution-augmented transformer for speech recognition,” in *INTERSPEECH*, 2020.
- [20] S. S. Stevens, J. Volkman, and E. B. Newman, “A scale for the measurement of the psychological magnitude pitch,” *The journal of the acoustical society of america*, vol. 8, no. 3, pp. 185–190, 1937.
- [21] L. R. Rabiner, R. W. Schafer *et al.*, “Introduction to digital speech processing,” *Foundations and Trends in Signal Processing*, vol. 1, no. 1–2, pp. 1–194, 2007.
- [22] E. Molina, A. M. Barbancho-Perez, L. J. Tardon-Garcia, I. Barbancho-Perez *et al.*, “Evaluation framework for automatic singing transcription,” in *ISMIR*, 2014.
- [23] Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimilakis, and R. Bittner, “The MUSDB18 corpus for music separation,” Dec. 2017.
- [24] A. Liutkus, F.-R. Stöter, Z. Rafii, D. Kitamura, B. Rivet, N. Ito, N. Ono, and J. Fontcave, “The 2016 signal separation evaluation campaign,” in *13th International Conference on Latent Variable Analysis and Signal Separation*, 2017, pp. 323–332.
- [25] E. M. Grais, M. U. Sen, and H. Erdogan, “Deep neural networks for single channel source separation,” in *ICASSP*, 2014, pp. 3734–3738.
- [26] P. Chandna, M. Miron, J. Janer, and E. Gómez, “Monoaural audio source separation using deep convolutional neural networks,” in *Latent Variable Analysis and Signal Separation (LVA/ICA)*, 2017, pp. 258–266.
- [27] Q. Kong, Y. Cao, H. Liu, K. Choi, and Y. Wang, “Decoupling magnitude and phase estimation with deep resnet for music source separation,” in *ISMIR*, 2021.
- [28] A. Jansson, E. Humphrey, N. Montecchio, R. Bittner, A. Kumar, and T. Weyde, “Singing voice separation with deep U-Net convolutional networks,” in *ISMIR*, 2017.

- [29] S. Uhlich, M. Porcu, F. Giron, M. Enenkl, T. Kemp, N. Takahashi, and Y. Mitsufuji, “Improving music source separation based on deep neural networks through data augmentation and network blending,” in *ICASSP*, 2017, pp. 261–265.
- [30] D. Stoller, S. Ewert, and S. Dixon, “Wave-U-Net: A multi-scale neural network for end-to-end audio source separation,” *arXiv preprint arXiv:1806.03185*, 2018.
- [31] Y. Luo and N. Mesgarani, “Conv-TasNet: Surpassing ideal time–frequency magnitude masking for speech separation,” *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 27, no. 8, pp. 1256–1266, 2019.
- [32] A. Défossez, N. Usunier, L. Bottou, and F. Bach, “Music source separation in the waveform domain,” *arXiv preprint arXiv:1911.13254*, 2019.
- [33] S. Rouard, F. Massa, and A. Défossez, “Hybrid transformers for music source separation,” in *ICASSP*, 2023.
- [34] C. Donahue, J. Thickstun, and P. Liang, “Melody transcription via generative pre-training,” in *ISMIR*, 2022.
- [35] S. Kum, J. Lee, K. L. Kim, T. Kim, and J. Nam, “Pseudo-label transfer from frame-level to note-level in a teacher-student framework for singing transcription from polyphonic music,” in *ICASSP*, 2022.
- [36] J.-Y. Wang and J.-S. R. Jang, “On the preparation and validation of a large-scale dataset of singing transcription,” in *ICASSP*, 2021, pp. 276–280.
- [37] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, and I. Sutskever, “Jukebox: A generative model for music,” *arXiv preprint arXiv:2005.00341*, 2020.
- [38] J.-Y. Hsu and L. Su, “Vocano: A note transcription framework for singing voice in polyphonic music.” in *ISMIR*, 2021, pp. 293–300.
- [39] B. Zhang and R. Sennrich, “Root mean square layer normalization,” *NeurIPS*, vol. 32, 2019.
- [40] J. Su, Y. Lu, S. Pan, A. Murthada, B. Wen, and Y. Liu, “Roformer: Enhanced transformer with rotary position embedding,” *arXiv preprint arXiv:2104.09864*, 2021.
- [41] K. Li, X. Hu, and Y. Luo, “On the use of deep mask estimation module for neural source separation systems,” *INTERSPEECH*, 2022.
- [42] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, “Language modeling with gated convolutional networks,” in *ICML*, 2017, pp. 933–941.
- [43] D. Wang and J. Chen, “Supervised speech separation based on deep learning: An overview,” *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 26, no. 10, pp. 1702–1726, 2018.
- [44] E. Gusó, J. Pons, S. Pascual, and J. Serrà, “On loss functions and evaluation metrics for music source separation,” in *ICASSP*, 2022, pp. 306–310.
- [45] C. Hawthorne, E. Elsen, J. Song, A. Roberts, I. Simon, C. Raffel, J. Engel, S. Oore, and D. Eck, “Onsets and frames: Dual-objective piano transcription,” in *ISMIR*, 2018, pp. 50–57.
- [46] I. Pereira, F. Araújo, F. Korzeniowski, and R. Vogl, “Moisesdb: A dataset for source separation beyond 4-stems,” in *ISMIR*, 2023.
- [47] Z. Wang, K. Chen, J. Jiang, Y. Zhang, M. Xu, S. Dai, X. Gu, and G. Xia, “Pop909: A pop-song dataset for music arrangement generation,” in *ISMIR*, 2020.
- [48] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, “librosa: Audio and music signal analysis in python,” in *Proceedings of the 14th python in science conference*, vol. 8, 2015, pp. 18–25.
- [49] M. Slaney, “Auditory toolbox,” *Interval Research Corporation, Tech. Rep.*, vol. 10, no. 1998, p. 1194, 1998.
- [50] I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” in *ICLR*, 2019.
- [51] T. Dao, “Flashattention-2: Faster attention with better parallelism and work partitioning,” *arXiv preprint arXiv:2307.08691*, 2023.
- [52] A. Défossez, “Hybrid spectrogram and waveform source separation,” *arXiv preprint arXiv:2111.03600*, 2021.
- [53] M. Kim and J. H. Lee, “Sound demixing challenge 2023–music demixing track technical report,” *arXiv preprint arXiv:2306.09382*, 2023.
- [54] E. Vincent, R. Gribonval, and C. Févotte, “Performance measurement in blind audio source separation,” *IEEE Trans. Audio Speech Lang. Process.*, vol. 14, no. 4, pp. 1462–1469, 2006.
- [55] F.-R. Stöter, A. Liutkus, and N. Ito, “The 2018 signal separation evaluation campaign,” in *LVA/ICA*, 2018, pp. 293–305.
- [56] X. Gu, W. Zeng, J. Zhang, L. Ou, and Y. Wang, “Deep audio-visual singing voice transcription based on self-supervised learning models,” *arXiv preprint arXiv:2304.12082*, 2023.
- [57] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, D. P. Ellis, and C. C. Raffel, “Mir_eval: A transparent implementation of common mir metrics.” in *ISMIR*, vol. 10, 2014.
- [58] K. W. Cheuk, K. Choi, Q. Kong, B. Li, M. Won, J.-C. Wang, Y.-N. Hung, and D. Herremans, “Jointist: Simultaneous improvement of multi-instrument transcription and music source separation via joint training,” *arXiv preprint arXiv:2302.00286*, 2023.

UNSUPERVISED SYNTHETIC-TO-REAL ADAPTATION FOR OPTICAL MUSIC RECOGNITION

Noelia Luna-Barahona¹ Adrián Roselló¹ María Alfaro-Contreras¹
David Rizo^{1,2} Jorge Calvo-Zaragoza¹

¹ Pattern Recognition and Artificial Intelligence Group, University of Alicante, Spain

² Instituto Superior de Enseñanzas Artísticas de la Comunidad Valenciana, Spain

{noelia.luna, adrian.rosello}@ua.es, malfaro, drizo, jcalvo}@dlsi.ua.es

ABSTRACT

The field of Optical Music Recognition (OMR) focuses on models capable of reading music scores from document images. Despite its growing popularity, OMR is still confined to settings where the target scores are similar in both musical context and visual presentation to the data used for training the model. The common scenario, therefore, involves manually annotating data for each specific case, a process that is not only labor-intensive but also raises concerns regarding practicality. We present a methodology based on training a neural model with synthetic images, thus reducing the difficulty of obtaining labeled data. As sheet music renderings depict regular visual characteristics compared to scores from real collections, we propose an unsupervised neural adaptation approach consisting of loss functions that promote alignment between the features learned by the model and those of the target collection while preventing the model from converging to undesirable solutions. This unsupervised adaptation bypasses the need for extensive retraining, requiring only the unlabeled target images. Our experiments, focused on music written in Mensural notation, demonstrate that the methodology is successful and that synthetic-to-real adaptation is indeed a promising way to create practical OMR systems with little human effort.

1. INTRODUCTION

Encoding and transcribing sheet music by hand is a complex and error-prone task that often requires individuals with specialized knowledge of the music notation at hand. An alternative to this manual digitization is the utilization of advanced artificial intelligence technologies, which enable the automated interpretation of musical documents. This technology is known as Optical Music Recognition (OMR) [1].

OMR has been a subject of research for several years [2], experiencing slow progress initially [3]. However, the recent adoption of advanced machine learning techniques, notably Deep Learning, has catalyzed significant improvements in the field [4]. Current OMR systems, albeit not fully perfected, present a more efficient and accurate alternative to manual transcription efforts [5].

In the context of machine learning, existing literature reports models that achieve satisfactory levels of accuracy when processing collections that share graphic characteristics with the training corpus [6–9]. This situation poses challenges for applying OMR technology to new collections, as it is not always feasible, practical, or resource-efficient to dedicate efforts towards annotating a segment of the target collection for training purposes.

This work explores the potential of creating OMR models to address diverse music collections by leveraging synthetic data for training. Given the vast availability of symbolic music data and score engraving tools, generating synthetic data for training presents itself as a viable and promising approach. However, the significant graphical disparities between renderings and real music collections suggest that a straightforward application of such synthetic data might not suffice. To address this issue, we consider the strategy proposed by Alfaro-Contreras & Calvo-Zaragoza [10], aimed at adapting pre-trained transcription models—in our case, initially trained on synthetic data to accommodate real-world music collections. Some previous works on OMR also implement domain adaptation but in other related tasks such as layout analysis [11] or music-object detection [12].

Our experimentation focuses on early monophonic music written in Mensural notation, as there exists a significant number of collections in this notation, each with specific characteristics. This abundance enables us to conduct a thorough examination, aiming to derive conclusions that are broadly applicable and representative. We will use the same synthetic data (and model) to independently adapt to five different Mensural collections. Our experiments indicate that our approach enables consistent synthetic-to-real adaptation, leading to notable improvements in many settings compared to the baseline. While there is still potential for better adaptation, our method represents a significant step towards developing practical OMR models that do not rely on corpus-specific labeled data.



2. BACKGROUND

The traditional OMR pipeline comprises four stages [13]: (i) *image pre-processing*, which includes tasks such as binarization, distortion correction, or staff separation; (ii) *music symbol detection*, which involves steps such as staff-line removal, connected-component search, and classification; (iii) *notation assembly*, which relates the individual identified components to reconstruct the musical notation; and (iv) *encoding*, which exports the recognized notation to a specific language for storage and further computational processing.

With the rise of Deep Learning, the so-called *end-to-end* formulation has emerged as an alternative to OMR. This approach, which has been dominating the state of the art in other applications such as text or speech recognition [14, 15], is currently considered the reference model in OMR. The related literature includes many successful solutions of this type [16–18], often with some prior pre-processing such as staff segmentation [19, 20].

However, as introduced above, there is still no computational approach for creating a universal OMR system; *i.e.*, one that is capable of dealing with any kind of collection. Instead, in this work, we take a more practical strategy that leverages synthetic data and domain adaptation. Synthetic data, generated through score engraving tools, provides a seemingly infinite resource for training machine learning models without the necessity for laborious manual annotation.

Nevertheless, the utilization of synthetic data presents a critical challenge: while synthetic scores are generated under precise, controlled conditions, real-world music scores exhibit a wide variety of visual characteristics. This variance results in a significant domain gap, where models trained exclusively on synthetic data struggle to generalize. Domain Adaptation (DA) becomes essential to reduce performance degradation by fine-tuning a pre-trained model with unlabeled data from the target domain [21]. While DA has been applied to some stages of the legacy OMR workflow [12, 22], its application to end-to-end approaches remains unexplored. Our contribution is the introduction of an unsupervised synthetic-to-real DA method that employs a specific set of loss functions to adapt pre-trained models using only target staff images.

3. METHODOLOGY

The methodology followed in this work is illustrated in Figure 1. First, a general OMR model is trained in a supervised way using synthetic data. Then, before processing a real collection, for which images but no annotations are available, we apply an unsupervised adaptation approach that modifies the pre-trained model. Then, the adapted model is used to perform OMR on the targeted collection.

The following sections describe the operation of the OMR model and the unsupervised adaptation approach.

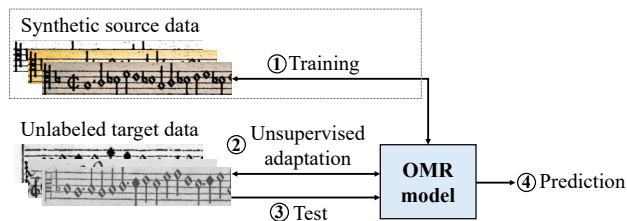


Figure 1: Overview of the unsupervised synthetic-to-real Optical Music Recognition methodology followed in this work.

3.1 Optical Music Recognition model

Our OMR model works at the staff level, assuming that a certain layout analysis has already detected the different staves of the score, as in recent literature [6, 7, 9, 23]. Then, the goal of the model is to retrieve the sequence of music-notation symbols that appear in a given staff.

The state of the art for the aforementioned formulation is to train a Convolutional Recurrent Neural Network (CRNN), using the so-called Connectionist Temporal Classification (CTC) [9, 23]. The convolutional part learns discriminative features from images, while the recurrent block models these features in terms of music-symbol sequences. CTC allows training without explicit information about the location of the symbols in the image [24], which enables an end-to-end learning framework from just pairs of staff images and corresponding transcripts.

Given a staff image \mathbf{x} , the output of the CRNN is a stochastic sequence $\pi_{\mathbf{x}} = (\pi_{x_1}, \dots, \pi_{x_K})$, $\pi_{x_i} \in [0, 1]^{\Sigma}$, where K is the number of frames (columns) processed by the recurrent block and Σ represents the vocabulary of music-notation symbols.¹ $\pi_{x_i}^{\sigma}$ represents the probability of observing music-notation symbol σ in the i -th frame of the input ($\sum_{\sigma \in \Sigma} \pi_{x_i}^{\sigma} = 1$). The whole sequence $\pi_{\mathbf{x}}$ is often referred to as the *posteriorgram* of \mathbf{x} .

For performing OMR, the posteriorgram is converted into an actual sequence of music-notation symbols by following a *greedy policy* based on retrieving the most probable symbol per frame and applying some direct operations to remove repeated symbols and “blank” tokens.

3.2 Unsupervised adaptation

The model explained in the previous section has demonstrated its goodness in scenarios where the training data belongs to the same collection to be processed. However, this is not interesting in most practical cases, especially when the model is trained with synthetic data, as it barely generalizes to real collections. In this section, we explain the considered approach to adapt a pre-trained model to a (real) target collection using only its images.

Specifically, given a mini-batch $b = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N)$ of target staff images, we fine-tune the pre-trained model with the following loss:

¹ The number of frames is usually less than the number of columns of the original image because the convolutional block typically includes pooling operations.

$$\mathcal{L} = \alpha \cdot \mathcal{L}_a(b) + \beta \cdot \mathcal{L}_r(b) \quad (1)$$

The loss involves two terms, weighted by parameters α and β (to be tuned empirically), respectively: i) one that modifies the model’s weights to perform adaptation to a real collection (\mathcal{L}_a), and ii) a regularization term that prevents meaningless convergence for OMR (\mathcal{L}_r). These are formally introduced in the following sections.

For this adaptation stage, we are not allowed to use the synthetic training corpus, despite being available in our particular case. This is because we are interested in the case for which the original training set is not accessible.

3.2.1 Adaptation term

The first mechanism aims to reduce the discrepancy between the pre-trained model and the target collection by aligning extracted features. Specifically, we approximate the distribution of the pre-trained model as a Gaussian distribution $\mathcal{N}_S(\mu_S, \sigma_S^2)$, using the Batch Normalization (BN) statistics stored in the corresponding layers.² During training, BN normalizes layer outputs within each mini-batch, ensuring zero-mean and unit-variance. Exponentially weighted averages of these mean and variance vectors, represented as μ_S and σ_S^2 , respectively, are stored during training so that they can be used in the prediction phase to perform standardization.

To reduce distribution discrepancies between the pre-trained and the target collection, we fine-tune the layers preceding BN by forcing their extracted features to have mean and variance vectors similar to those of the source data. Specifically, when given batch b , we compute the mean μ_b and variance σ_b^2 . The target batch feature distribution is subsequently approximated as $\mathcal{N}_b(\mu_b, \sigma_b^2)$. We then employ the feature-averaged Kullback-Leibler (KL) divergence to align the target batch feature distribution with the pre-trained feature distribution:

$$\mathcal{L}_a(b) = \mathcal{D}_{\text{KL}}(\mathcal{N}_b || \mathcal{N}_S) \quad (2)$$

This is described in the context of a single BN layer, but it can be applied to many of them by calculating the loss for each and then adding them up.

3.2.2 Regularization

The previous mechanism can lead to an informational collapse, where the model consistently extracts the same features, regardless of the input image, to match the expected distribution. This would lead to eventually predicting the same music-notation symbol in all frames, which is useless for OMR.

Furthermore, we want to encourage predictions that exhibit music-symbol diversity. This can be induced by maximizing entropy within each frame’s predictions across the batch with the following loss:³

² Assuming BN layers for this purpose is a soft constraint since most of the considered CRNN architectures for OMR include these.

³ Note that the equation is negating the entropy so that the loss is performing *maximization* during gradient descent.

$$-\sum_{k=1}^K \sum_{\sigma \in \Sigma} \mathcal{H}(\pi_{\mathbf{b}_k}^\sigma) = \sum_{k=1}^K \sum_{\pi \in \Sigma'_S} \sum_{i=1}^{|\mathbf{b}|} \left(\pi_{x_{i_k}}^\sigma \log \pi_{x_{i_k}}^\sigma \right) \quad (3)$$

Specifically, this term penalizes that the same frame in different samples of the batch provides an identical probability distribution over the vocabulary Σ .

Unfortunately, minimizing Eq. 3 might lead to probabilities for a specific frame to be uniformly distributed. In other words, this encourages the model to predict that all music-notation symbols are equiprobable in each frame. However, these distributions should ideally resemble a one-hot distribution, linking each image frame to a single symbol from Σ . To mitigate this, we must further regularize the model to encourage the predictions to behave as one-hot vectors by minimizing the entropy of each frame’s output:

$$\sum_{i=1}^{|\mathbf{b}|} \sum_{k=1}^K \mathcal{H}(\pi_{x_{i_k}}) = -\sum_{i=1}^{|\mathbf{b}|} \sum_{k=1}^K \sum_{\sigma \in \Sigma} \left(\pi_{x_{i_k}}^\sigma \log \pi_{x_{i_k}}^\sigma \right) \quad (4)$$

Therefore, the regularization term of our unsupervised adaptation process becomes:

$$\mathcal{L}_r(b) = \sum_{i=1}^{|\mathbf{b}|} \sum_{k=1}^K \mathcal{H}(\pi_{x_{i_k}}) - \sum_{k=1}^K \sum_{\sigma \in \Sigma} \mathcal{H}(\pi_{\mathbf{b}_k}^\sigma) \quad (5)$$

where predictions are encouraged to behave like the output of an OMR process, while preventing all predictions from providing the same symbol.

4. DATA

This section covers data handling and preparation, encompassing synthetic data generation to pre-train the model, the considered real datasets for the adaptation experiments, and the encoding of the output vocabulary of the OMR.

4.1 Synthetic data generation

We have considered a modified version of the Printed Images of Mensural Staves (PRIMENS) dataset [9]. The PRIMENS dataset is a synthetic corpus designed to emulate low-quality scans of printed mensural sources. It was obtained by transforming compositions by composers such as Agricola, Frye, and Ockeghem, which are accessible through the Josquin Research Project (JRP)⁴. The original JRP files consist of transcriptions in Common Western Modern notation encoded using `**kern` format. To obtain a Mensural notation dataset, Martínez-Sevilla et al. converted the original files to `**mens` format [25]. Given the polyphonic nature of these compositions, they isolated individual monophonic excerpts by segmenting them into randomly chosen measures spanning from 3 to 18. The

⁴ <https://josquin.stanford.edu/>. Last accessed April 12th, 2024.

authors also modified the original clefs accordingly to increase variability and thus expand the dataset size.

The images were generated using the digital engraver Verovio [26], with random values applied to all available options within permitted ranges. Subsequently, these images were also distorted to mimic genuine printed image scans by employing a random sequence of graphical filters through GraphicsMagick Image Processing System. Furthermore, this simulation of real images was further enhanced by blending randomly damaged old paper textures with distorted images. Figure 2a shows a staff example of the PRIMENS dataset.

When analyzing the music-symbol distribution of the original PRIMENS dataset, we found that it lacked bar lines and custodes, two common elements in Mensural corpora. To standardize the vocabulary, we randomly introduced bar lines with a probability of 10% per monophonic excerpt. Custodes were added at the end of each staff, positioned at the most repeated pitch of that region to ensure meaningful vertical staff alignment.

4.2 Real datasets

We have considered five corpora of Mensural music, both handwritten and typeset:

- CAPITAN corpus [27]: a set of 97 manuscript pages dated from the 17th century of liturgical music. An example of a particular staff from this corpus is depicted in Figure 2b.
- Il Lauro Secco (SEILS) corpus [28]: a collection of 151 typeset pages corresponding to an anthology of Italian madrigals of the 16th century. Figure 2c shows a staff example of this set.
- GUATEMALA corpus [29]: a collection of 385 handwritten pages from a polyphonic choir book, part of a larger collection held at the “Archivo Histórico Arquidiocesano de Guatemala”. An example of a particular staff from this corpus is depicted in Figure 2d.
- MOTTECTA corpus [9]: a set of 297 printed pages from a collection of the “Biblioteca Digital Hispánica” dated from the 17th century. Figure 2e shows a staff example of this set.
- MAGNIFICAT corpus [5]: a set of 127 typeset pages corresponding to a Spanish choir book of the 16th century. See Figure 2f for a sample of this corpus.

4.3 Output encoding

OMR primarily deals with image signals, leading OMR systems to prioritize learning graphic concepts over musical ones. This explains why, when training end-to-end OMR models, an internal representation referred to as “agnostic” is used instead of a semantic representation where music symbols are encoded based on their musical significance [28, 30]. This agnostic representation categorizes elements within a collection of musical symbols according

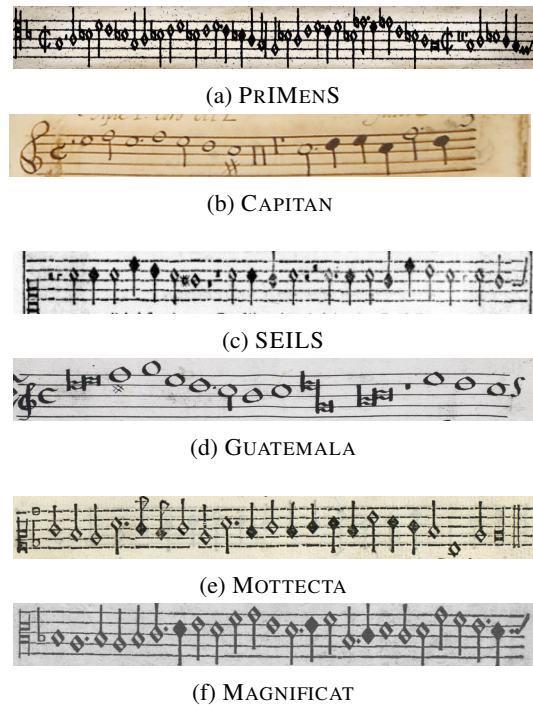


Figure 2: Staff samples of the synthetic data (a) used to train the initial OMR model, which is then adapted to the five Mensural corpora (b-f).

to their form, representing event duration, and their height or vertical position on the staff, denoting pitch. In essence, each symbol is denoted as the 2-tuple $z_i = \langle f_i, h_i \rangle : f_i \in \Sigma_F, h_i \in \Sigma_H$, where Σ_F and Σ_H represent the spaces for the different form and height labels, respectively. This approach effectively describes all symbols, including rests that symbolize silence and can be positioned at various vertical locations.

The concise structure of the agnostic representation not only facilitates faster convergence of OMR models but also enables non-experts to annotate music data, making the subsequent conversion to a semantic representation automatable [31]. However, holistic OMR models do not leverage this dual dimensionality. Instead, they treat each combination of form and height as a single category— $|\Sigma| = |\Sigma_F| \times |\Sigma_H|$. Recent works [8, 23] have shown that splitting the symbols in z_i into their two components and retrieving them sequentially—first, the form and then, the height—leads better recognition rates. Note that the cardinality of the set of symbols in this *split-sequence encoding* is $|\Sigma| = |\Sigma_F| + |\Sigma_H|$, much lower than that of the *standard encoding*, at the expense of doubling the length of the sequence to be predicted. Figure 3 shows a staff sample and its encoding representations in standard and split-sequence encoding.

In this work, we consider both the standard encoding and the split-sequence encoding representations. When using the latter encoding, we adhere to the 2D-greedy decoding method proposed in [8]. This method adjusts the standard CTC greedy decoding to ensure that the output predictions conform to the form-height pattern of the split-



barline:L1, clef.C:L4, note.quarter_up:S0,
 note.wholeBlack:L2, note.half_up:S2
 barline, L1, clef.C, L4, note.quarter_up, S0,
 note.wholeBlack, L2, note.half_up, S2

Figure 3: Staff sample and its encoding representations in standard (above) and split-sequence (below) encoding. Note that L_n and S_n respectively denote the line or space of the staff on which the symbol may be placed, which refers to its height property.

sequence encoding representation.

Note that when using the split-sequence encoding representation, we transition from a cardinality of $\Sigma_F \times \Sigma_H$ to one of $\Sigma_F + \Sigma_H$. This implies fewer different symbols and subsequently enables a greater overlap of vocabularies between the source and target collections. This feature makes it particularly suitable for our synthetic-to-real scenario.

Table 1 provides a summary of the characteristics of each label space for the considered corpora. As for data partitioning, we adhere to the same training, validation, and test splits as outlined in the referenced works.

Table 1: Overview of the corpora used in this work: number of staves, vocabulary size for each label space considered (form and height separately for the split-sequence encoding, and a single token combining these two pieces of information for the standard encoding), and engraving style.

	Staves	Vocabulary			Engraving style
		Form	Height	Combined	
PRIMENS	42 136	37	34	386	Synthetic
CAPITAN	828	62	16	372	Handwritten
SEILS	1 136	37	17	205	Typeset
GUATEMALA	3 263	52	17	315	Handwritten
MOTTECTA	1 847	38	15	228	Typeset
MAGNIFICAT	1 340	42	19	220	Typeset

5. EXPERIMENTAL SET UP

This section describes the evaluation protocol and the implementation details.

5.1 Evaluation metric

We consider the Symbol Error Rate (SER) for assessing the performance of the presented recognition scheme, as in previous works [6–9]. This metric is computed as the average number of elementary editing operations (insertions, deletions, or substitutions) required to match the sequence predicted by the model with that in the ground truth, normalized by the length of the latter. In mathematical terms, this is expressed as:

$$\text{SER} (\%) = \frac{\sum_{i=1}^{|\mathcal{S}|} \text{ED}(\hat{\mathbf{z}}_i, \mathbf{z}_i)}{\sum_{i=1}^{|\mathcal{S}|} |\mathbf{z}_i|} \quad (6)$$

where $\mathcal{S} \subset \mathcal{X} \times \mathcal{Z}$ is a set of test data, $\text{ED} : \mathcal{Z} \times \mathcal{Z} \rightarrow \mathbb{N}_0$ denotes the string edit distance [32], and $\hat{\mathbf{z}}_i$ and \mathbf{z}_i respectively represent the estimated and target sequences. For comparative purposes, we convert all predicted and ground-truth sequences to split-sequence before computing the metric.

5.2 Implementation details

The CRNN scheme is based on that used typically for OMR [7, 9, 27]. Specifically, we used four convolutional layers that applied 64 filters of size 5×5 , 64 filters of size 5×5 , 128 filters of size 3×3 , and 128 filters of size 3×3 , respectively. We considered a Leaky ReLU activation with a negative slope of $\alpha = 0.2$ and max-pooling stages of size and striding factors of 2×1 (except the first convolutional layer, which is 2×2). The produced feature maps were fed into two Bidirectional Long Short-Time Memory layers with 256 hidden units each and a dropout value of 50%, followed by a fully-connected network with $|\Sigma'|$ units that provide a probability for each possible music-notation token.

The evaluation pipeline consisted of two stages: (i) training the source model, and (ii) adapting it to the target dataset using the AMD method. For (i), we used the ADAM optimizer with a batch size of 16 elements and a fixed learning rate of 10^{-3} . We stopped the training using an early stopping strategy with a patience of 20 epochs, retaining the weights that minimize the SER metric in the validation partition. In (ii) we maintained the batch size of 16, the learning rate was selected through a random search ranging from 10^{-3} to 3×10^{-4} , and a maximum of 50 training-adaptation epochs was considered as we fine-tuned an already trained model.

Regarding data pre-processing, we replicated the exact experimental conditions outlined in the aforementioned reference works. Specifically, we resized each staff image to a height of 64 pixels, preserving the aspect ratio (individual samples may vary in width), and converted them to grayscale without any additional pre-processing steps. Additionally, following the approach outlined in the aforementioned works, we incorporated a data augmentation step during the training of the source models.

6. RESULTS

This section presents the results obtained from applying the experimental scheme to the different presented corpora. Specifically, Table 2 depicts the performance of the PRIMENS model before and after adaptation for each real target Mensural corpus in terms of the SER metric.⁵

The most important remark is that the considered synthetic-to-real adaptation framework improves the performance of the synthetic-only scenario across all datasets. The approach does not solve the adaptation challenge completely (the reference value is still far in most cases), but it allows taking the model to more usable levels without

⁵ Code at: <https://github.com/OMR-PRAIG-UA-ES/ISMIR-2024-SYNTHETIC2REAL-OMR>.

Table 2: Results in terms of the SER (%) metric for each real Mensural corpus before and after the unsupervised adaptation of the OMR model trained with the synthetic PRIMENS dataset. For completeness, we also include the in-collection performance, where the real corpus is used for both training and testing. These performances are shaded in gray, serving as an upper-bound reference. Final row reports the relative improvement ($\downarrow\Delta$ %) when adaptation is performed.

	Target corpus									
	CAPITAN		SEILS		GUATEMALA		MOTTECTA		MAGNIFICAT	
	Standard	Split-sequence	Standard	Split-sequence	Standard	Split-sequence	Standard	Split-sequence	Standard	Split-sequence
In-collection (reference)	6.7	6.2	1.8	1.7	1.6	1.6	3.3	2.9	1.5	1.5
Before adaptation	45.9	43.1	23.9	21.3	46.9	53.4	25.8	29.2	12.8	12.8
After adaptation	32.9	32.9	19.3	18.5	21.4	18.1	17.1	16.4	9.1	10.6
Relative improvement	$\downarrow 28\%$	$\downarrow 23\%$	$\downarrow 19\%$	$\downarrow 14\%$	$\downarrow 54\%$	$\downarrow 66\%$	$\downarrow 34\%$	$\downarrow 44\%$	$\downarrow 29\%$	$\downarrow 17\%$

the need to initially annotate data. This is quite useful, for example, in the context of OMR plus post-correction.

The degree of relative improvement varies depending on the specific dataset, ranging from 66% to 14%. In this sense, it is difficult to draw a correlation between the different factors and the degree of improvement. However, it is worth highlighting that the scenarios with a greater margin (for example, GUATEMALA and CAPITAN) lead to a greater absolute improvement. This may indicate that there is a glass ceiling to the performance that can be obtained by training with a synthetic corpus, since in the cases where the result is already relatively successful (e.g. MAGNIFICAT) the improvement is rather limited.

Concerning the output encoding, the split-sequence encoding generally yields better SER figures in the in-collection scenario. However, the differences are marginal in the other two scenarios. Therefore, this does not represent a relevant factor for adaptation.

To provide more insights into the adaptation process, we explored the “relevant” parts of the image that the different OMR models consider to predict the symbols. Gradient-weighted Class Activation Mapping (Grad-CAM) [33] is an interpretability method that uses the gradients of any target prediction to produce a coarse localization map highlighting the important regions in the image for such predictions. Figure 4 shows the activation map over the same test image for the three different scenarios considered: (a) in-collection, (b) before adaptation, and (c) after adaptation. Specifically, we display here the case of processing the real collection GUATEMALA. We can observe how the initially misplaced pixel activations in scenario (b) are corrected to the actual music symbols after adaptation in scenario (c), showing a high degree of similarity to the activation map of the in-collection model of scenario (a).

7. CONCLUSIONS

Existing end-to-end OMR approaches have exhibited remarkable performance in transcribing collections that share graphic characteristics with the training corpus. However, when this condition is not met, allocating resources to manually annotate training data to maintain performance levels becomes impractical and resource-intensive. Our work proposes a possible solution to this challenge. Firstly, we train an initial OMR model with synthetic scores. By doing so, we eliminate the need for hu-

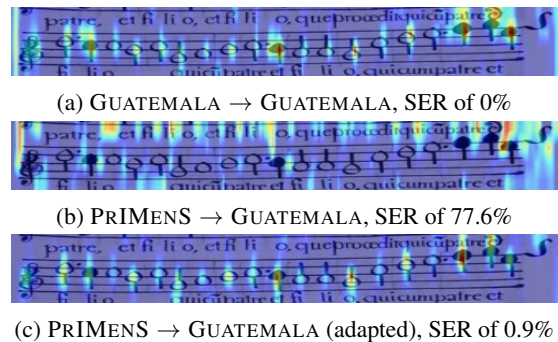


Figure 4: Activation maps over the same GUATEMALA test image using an OMR model trained with (a) GUATEMALA scores, (b) PRIMENS scores, and (c) PRIMENS scores but adapted to GUATEMALA images.

man manual annotation of training data. Subsequently, we tailor this model to the specific characteristics of the target corpus through unsupervised adaptation, using only unlabeled images from the target corpus. This adaptation process employs a loss function to align the learned features of the model with those of the target collection while ensuring the model does not converge to undesirable solutions. Our experiments across five distinct Mensural datasets validate the effectiveness of our synthetic-to-real adaptation as a viable approach to developing universal OMR systems with little human effort. However, there remains room for improvement. Future research avenues may explore leveraging self-labeled samples obtained through the adapted model to further enhance its performance and robustness or exploring few-shot scenarios.

8. ACKNOWLEDGEMENTS

This paper is supported by grant CISEJI/2023/9 from “Programa para el apoyo a personas investigadoras con talento (Plan GenT) de la Generalitat Valenciana”.

9. REFERENCES

- [1] J. Calvo-Zaragoza, J. H. Jr, and A. Pacha, “Understanding Optical Music Recognition,” *ACM Computing Surveys*, vol. 53, no. 4, pp. 1–35, 2020.
- [2] D. Bainbridge and T. Bell, “The challenge of optical

- music recognition,” *Computers and the Humanities*, vol. 35, pp. 95–121, 2001.
- [3] A. Rebelo, I. Fujinaga, F. Paszkiewicz, A. R. Marcal, C. Guedes, and J. S. Cardoso, “Optical music recognition: state-of-the-art and open issues,” *International Journal of Multimedia Information Retrieval*, vol. 1, pp. 173–190, 2012.
- [4] J. Calvo-Zaragoza, J. C. Martínez-Sevilla, C. Peñarubia, and A. Ríos-Vila, “Optical music recognition: Recent advances, current challenges, and future directions,” in *Proceedings in International Conference on Document Analysis and Recognition*, ser. Lecture Notes in Computer Science, M. Coustaty and A. Fornés, Eds., vol. 14193. San José, CA, USA: Springer, 2023, pp. 94–104.
- [5] M. Alfaro-Contreras, D. Rizo, J. M. Inesta, and J. Calvo-Zaragoza, “OMR-assisted transcription: a case study with early prints,” in *Proceedings of the 22nd International Society for Music Information Retrieval Conference*. Online: ISMIR, Nov. 2021, pp. 35–41.
- [6] A. Baró, C. Badal, and A. Fornés, “Handwritten Historical Music Recognition by Sequence-to-Sequence with Attention Mechanism,” in *Proceedings of the 17th International Conference on Frontiers in Handwriting Recognition*. Dortmund, Germany: IEEE, Sep. 2020, pp. 205–210.
- [7] M. Villarreal and J. A. Sánchez, “Handwritten Music Recognition Improvement through Language Model Re-interpretation for Mensural Notation,” in *Proceedings of the 17th International Conference on Frontiers in Handwriting Recognition*. Dortmund, Germany: IEEE, Sep. 2020, pp. 199–204.
- [8] M. Alfaro-Contreras, A. Ríos-Vila, J. J. Valero-Mas, J. M. Iñesta, and J. Calvo-Zaragoza, “Decoupling music notation to improve end-to-end Optical Music Recognition,” *Pattern Recognition Letters*, vol. 158, pp. 157–163, 2022.
- [9] J. C. Martínez-Sevilla, A. Roselló, D. Rizo, and J. Calvo-Zaragoza, “On the Performance of Optical Music Recognition in the Absence of Specific Training Data,” in *Proceedings of the 24th International Society for Music Information Retrieval Conference*. Milan, Italy: ISMIR, Nov. 2023, pp. 319–326.
- [10] M. Alfaro-Contreras and J. Calvo-Zaragoza, “Align, minimize and diversify: A source-free unsupervised domain adaptation method for handwritten text recognition,” *arXiv preprint arXiv:2404.18260*, 2024.
- [11] F. J. Castellanos, A. J. Gallego, J. Calvo-Zaragoza, and I. Fujinaga, “Domain adaptation for staff-region retrieval of music score images,” *International Journal on Document Analysis and Recognition (IJ DAR)*, vol. 25, no. 4, pp. 281–292, 2022.
- [12] L. Tuggener, R. Emberger, A. Ghosh, P. Sager, Y. P. Satyawana, J. Montoya, S. Goldschagg, F. Seibold, U. Gut, P. Ackermann *et al.*, “Real world music object recognition,” *Transactions of the International Society for Music Information Retrieval*, vol. 7, no. 1, pp. 1–14, 2024.
- [13] I. Fujinaga and G. Vigiensoni, “The art of teaching computers: the SIMSSA optical music recognition workflow system,” in *Proceedings of the 327th European Signal Processing Conference*. IEEE, 2019, pp. 1–5.
- [14] A. Chowdhury and L. Vig, “An Efficient End-to-End Neural Model for Handwritten Text Recognition,” in *British Machine Vision Conference*. Newcastle, UK: BMVA Press, Sep. 2018, p. 2018.
- [15] C.-C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, E. Gonina, N. Jaitly, B. Li, J. Chorowski, and M. Bacchiani, “State-of-the-Art Speech Recognition with Sequence-to-Sequence Models,” in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2018, pp. 4774–4778.
- [16] Pau Torras and Arnau Baró and Lei Kang and Alicia Fornés, “On the Integration of Language Models into Sequence to Sequence Architectures for Handwritten Music Recognition,” in *Proceedings of the 22nd International Society for Music Information Retrieval Conference*. Online: ISMIR, Nov. 2021.
- [17] M. Alfaro-Contreras, J. M. Iñesta, and J. Calvo-Zaragoza, “Optical music recognition for homophonic scores with neural networks and synthetic music generation,” *International Journal of Multimedia Information Retrieval*, vol. 12, no. 1, p. 12, 2023.
- [18] A. Ríos-Vila, J. Calvo-Zaragoza, and T. Paquet, “Sheet Music Transformer: End-To-End Optical Music Recognition Beyond Monophonic Transcription,” *arXiv preprint arXiv:2402.07596*, 2024.
- [19] M. Kletz and A. Pacha, “Detecting Staves and Measures in Music Scores with Deep Learning,” in *Proceedings of the 3rd International Workshop on Reading Music Systems*, Alicante, Spain, 2021, pp. 8–12.
- [20] F. J. Castellanos, C. Garrido-Munoz, A. Ríos-Vila, and J. Calvo-Zaragoza, “Region-based layout analysis of music score images,” *Expert Systems with Applications*, vol. 209, p. 118211, 2022.
- [21] W. M. Kouw and M. Loog, “A Review of Domain Adaptation without Target Labels,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 3, pp. 766–785, 2021.
- [22] F. J. Castellanos, A. J. Gallego, J. Calvo-Zaragoza, and I. Fujinaga, “Domain adaptation for staff-region retrieval of music score images,” *International Journal*

- on *Document Analysis and Recognition*, vol. 25, no. 4, pp. 281–292, 2022.
- [23] A. Ríos-Vila, J. Calvo-Zaragoza, and J. M. Iñesta, “Exploring the two-dimensional nature of music notation for score recognition with end-to-end approaches,” in *Proceedings of the 17th International Conference on Frontiers in Handwriting Recognition*. Dortmund, Germany: IEEE, Sep. 2020, pp. 193–198.
- [24] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, “Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks,” in *Proceedings of the 23rd International Conference on Machine Learning*. Pittsburgh, USA: Association for Computing Machinery, Jun. 2006, pp. 369–376.
- [25] D. Rizo, N. Pascual-León, and C. Sapp, “White mensural manual encoding: from humdrum to mei,” *Cuadernos de Investigación Musical*, 2019.
- [26] L. Pugin, R. Zitellini, and P. Roland, “Verovio - A library for Engraving MEI Music Notation into SVG,” in *Proceedings of the 15th International Society for Music Information Retrieval Conference*. Taipei, Taiwan: ISMIR, Jan. 2014.
- [27] J. Calvo-Zaragoza, A. H. Toselli, and E. Vidal, “Handwritten Music Recognition for Mensural notation with convolutional recurrent neural networks,” *Pattern Recognition Letters*, vol. 128, pp. 115–121, 2019.
- [28] E. Parada-Cabaleiro, A. Batliner, and B. W. Schuller, “A Diplomatic Edition of Il Lauro Secco: Ground Truth for OMR of White Mensural Notation,” in *Proceedings of the 20th International Society for Music Information Retrieval Conference*. Delft, The Netherlands: ISMIR, Nov. 2019, pp. 557–564.
- [29] M. E. Thomae, J. E. Cumming, and I. Fujinaga, “Digitization of Choirbooks in Guatemala,” in *Proceedings of the 9th International Conference on Digital Libraries for Musicology*. Prague, Czech Republic: Association for Computing Machinery, Jul. 2022, pp. 19–26.
- [30] J. Calvo-Zaragoza and D. Rizo, “Camera-PrIMuS: Neural End-to-End Optical Music Recognition on Realistic Monophonic Scores,” in *Proceedings of the 19th International Society for Music Information Retrieval Conference*. Paris, France: ISMIR, Sep. 2018, pp. 248–255.
- [31] A. Ríos-Vila, M. Esplà-Gomis, D. Rizo, P. J. Ponce de León, and J. M. Iñesta, “Applying Automatic Translation for Optical Music Recognition’s Encoding Step,” *Applied Sciences*, vol. 11, no. 9, pp. 3890–3912, 2021.
- [32] V. I. Levenshtein, “Binary Codes Capable of Correcting Deletions, Insertions, and Reversals,” *Soviet Physics-Doklady*, vol. 10, no. 8, pp. 707–710, 1966.
- [33] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization,” in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 618–626.

MMT-BERT: CHORD-AWARE SYMBOLIC MUSIC GENERATION BASED ON MULTITRACK MUSIC TRANSFORMER AND MUSICBERT

Jinlong Zhu¹ Keigo Sakurai¹ Ren Togo²
Takahiro Ogawa² Miki Haseyama²

¹ Graduate School of Information Science and Technology, Hokkaido University, Japan

² Faculty of Information Science and Technology, Hokkaido University, Japan

jinlong@lmd.ist.hokudai.ac.jp, mhaseyama@lmd.ist.hokudai.ac.jp

ABSTRACT

We propose a novel symbolic music representation and Generative Adversarial Network (GAN) framework specially designed for symbolic multitrack music generation. The main theme of symbolic music generation primarily encompasses the preprocessing of music data and the implementation of a deep learning framework. Current techniques dedicated to symbolic music generation generally encounter two significant challenges: training data’s lack of information about chords and scales and the requirement of specially designed model architecture adapted to the unique format of symbolic music representation. In this paper, we solve the above problems by introducing new symbolic music representation with MusicLang chord analysis model. We propose our MMT-BERT architecture adapting to the representation. To build a robust multitrack music generator, we fine-tune a pre-trained MusicBERT model to serve as the discriminator, and incorporate relativistic standard loss. This approach, supported by the in-depth understanding of symbolic music encoded within MusicBERT, fortifies the consonance and humanity of music generated by our method. Experimental results demonstrate the effectiveness of our approach which strictly follows the state-of-the-art methods.

1. INTRODUCTION

Music plays an indispensable role in our daily lives, and there is a significant demand for creating new musical contents. Automatic music generation is one of the most intriguing tasks in bringing new music experiences to consumers [1]. The earliest studies in the 1950s focused on a combination of music theory and Markov-chains-based probabilistic models, and realized randomly creating music parts and combining them into a synthesis [2]. Contemporary studies have achieved higher quality and faster music generation by utilizing advanced neural networks

such as Generative Adversarial Networks (GANs), Transformer, and diffusion models [3–5]. Despite significant advancements, previous methods continue to suffer from challenges such as insufficient data extraction and unstable training trajectories. Consequently, there is room for new approaches for more effective music representations and more robust deep learning architectures.

In particular, chords are crucial for conveying emotional and humanistic expressions in music, yet few methods take chords into account in symbolic music representation. Consequently, previous methods are deprived of indispensable information about chords and scales. This lack results in the generation of music that exhibits a diminished degree of humanity. Therefore, previous methods for music generation face limitations in their ability to produce human-like and high quality expressions [6]. A feasible solution to overcome this difficulty is the integration of a chord analysis model [7]. Chord analysis model aids in the extraction of chord data from raw audio, fostering a novel representation method that encompasses chord information [8–11]. With the aid of state-of-the-art chord analysis models, we can generate more harmonious and structured music with more regular chord progressions by automatically extracting and encoding chords from raw audio files. Therefore, it is expected that adopting chord analysis models in creating new symbolic representations of music will enable the generation of music that is closer to human composition.

Another problem arises from the ever-changing format of symbolic music representation, which makes designing the model’s architecture that fits symbolic music generation to be another challenge. GANs are widely applied in the symbolic music generation field because the addition of a discriminator obviously strengthens the fidelity of the overall generative model [12–17]. The performance of GANs is deeply influenced by the architecture of the generator and discriminator. Previous studies have demonstrated the effectiveness of transformer-based generators [6, 7, 18–23]. Whereas, the architecture of the discriminator has been extensively discussed in recent years. Some methods [12, 14, 16, 17] involve constructing a discriminator based on CNN or Transformer, while others [15] utilize pre-trained models adapted to their tasks. Compared to hand-crafted discriminators, using pre-trained models often achieves a fairly good result



© J. Zhu, K. Sakurai, R. Togo, T. Ogawa and M. Haseyama. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** J. Zhu, K. Sakurai, R. Togo, T. Ogawa and M. Haseyama, “MMT-BERT: Chord-aware Symbolic Music Generation Based on Multitrack Music Transformer and MusicBERT”, in *Proc. of the 25th Int. Society for Music Information Retrieval Conf.*, San Francisco, United States, 2024.

because pre-trained models are already trained on large and diverse datasets. Therefore, the application of pre-trained models will allow the GAN to leverage their learning and knowledge, ensuring the training efficiency and stability.

However, there are few choices of pre-trained models designed for symbolic music representation that can be used as the discriminator considering the input format and length limitation. We solve this problem by applying BERT-based scores, which are well correlated with human ranking and can jointly measure quality and diversity [24,25]. Since BERT is trained using a self-supervised loss on bidirectional contexts of all attention layers, it can effectively extract representations [26–28]. Muhamed *et al.* employ a pre-trained Span-BERT model and achieve considerable results on harmonic choices and overall music quality, showing that pre-trained BERT-based models outperform CNN-based discriminator [15]. Hence, employing a pre-trained model as a discriminator can amplify the overall performance of the GAN model.

In this paper, we propose a novel symbolic music generation method using the chord-aware symbolic music representation and MusicBERT-based discriminator. In terms of symbolic musical expression, we introduce the novel symbolic music representation with chord information derived from MusicLang¹, one of the state-of-the-art chord analysis models. By employing symbolic music representation with chord information, our model can achieve the generation of more human-like music that considers chord progressions. For the model architecture, we employ the Multitrack Music Transformer (MMT) [6] as the generator and fine-tune the MusicBERT [29], a symbolic music understanding model pre-trained in large-scale dataset, as the discriminator. Leveraging the superior comprehension capabilities of MusicBERT, we can improve GAN’s performance, thereby facilitating the creation of higher-quality music. Furthermore, we introduce relativistic standard loss to further optimize the stability and consistency of the training process [30]. The use of Relativistic Standard GAN (RS-GAN) has realized great results in the field of image generation. It enables models to account for the fact that half of the data in a mini-batch is fake, leading to more accurate estimations of data realism [14, 31]. Building upon the innovations mentioned above, our model is capable of retrieving substantial information about chords and scales, acquiring knowledge in music theory, and autonomously generating multitrack music of superior quality and enriched with human-like characteristics.

The contributions of this paper are summarized as follows.

- We propose a modified MMT style symbolic music representation including chord and scale information.
- We develop MMT-BERT, an optimized GAN architecture utilizing MMT and MusicBERT, with relativistic standard loss to enhance the stability of the training process and achieve better results.

¹ <https://musiclang.github.io/tokenizer/>

Representation	Multitrack	Instrument control	Compound tokens	Chord awareness
REMI [7]				✓
MMM [21]	✓			
CP [18]			✓	✓
FIGARO [23]	✓			✓
MMT [6]	✓	✓	✓	
MMT-BERT (ours)	✓	✓	✓	✓

Table 1. Comparisons of related representations.

2. RELATED WORKS

2.1 Symbolic Music Representation

To enable computers to properly understand music, research on symbolic music representation has been conducted for many years [32]. Musical Instrument Digital Interface (MIDI) is the most commonly used format for symbolic music representation, containing performance data and control information for musical notes. In the music processing community, many researchers symbolize music with MIDI-like events [33].

Huang *et al.* have proposed REvamped MIDI-derived events (REMI), which adds note duration and bar events, enabling models to generate music with subtle rhythmic repetition [7]. However, the REMI representation often encounters a challenge that the sequence is too long. Building upon the REMI framework, Hsiao *et al.* have proposed Compound Word Transformer (CP) [18]. CP modifies REMI’s approach by transforming one-dimensional sequence tokens into compound words sequence using specific rules. Although this modification significantly shortens the average token sequence length and simplifies the model’s ability to capture musical nuances, CP is hard to generate multitrack music [6]. Dong *et al.* have proposed their multitrack music representation, which represents music with a sequence of sextuple tokens, along with a Transformer-XL-based generation method Multitrack Music Transformer (MMT). This approach utilizes a decoder-only Transformer architecture, adept at processing multi-dimensional inputs and outputs. MMT leverages the advantages of the Transformer to enable the generation of longer multitrack music compositions than previous music generation methods. However, MMT’s representation scheme lacks chord event inclusion, an essential element in musical compositions. In contrast, our symbolic music representation technique builds on the foundation laid by MMT by integrating chord information, enabling our model to produce more harmonically rich compositions.

2.2 Generative Adversarial Network-based Music Generation

Previous studies have employed various GANs to realize symbolic music generation [12, 13, 17]. In early states, Dong *et al.* have proposed MuseGAN, a CNN-based GAN architecture, managing to generate multitrack music pieces [16]. However, CNN-based GANs often suffer from problems such as limited local perception, fixed-size inputs, etc. Muhamed *et al.* solved this problem by introducing their Transformer-GANs model, using a Transformer-

Event type t_j	Quintuple token $\mathbf{x}_i^{t_j}$
start-of-song	(0, 0, 0, 0, 0)
instrument	(0, 0, 0, 0, instrument)
start-of-score	(0, 0, 0, 0, 0)
note	(beat, position, pitch, duration, instrument)
chord	(beat, degree, root, mode, extension)
end-of-song	(0, 0, 0, 0, 0)

Table 2. The elements of the quintuple token $\mathbf{x}_i^{t_j}$ for each event type t_j .

XL-based generator and pre-trained Span-BERT as the discriminator [15]. Transformer-XL introduces the notion of recurrence into the deep self-attention network, enabling the reuse of hidden states from previous segments as memory for the current segment, allowing for the modeling of long-range dependencies. For the discriminator, Span-BERT is utilized to extract sequence embeddings followed by a pooling and linear layer. The bidirectional transformer has a comparable capacity to the transformer-based generator and uses the self-attention mechanism to capture meaningful aspects of the input music sequence. Their research validates the efficacy of employing a Transformer-XL-based generator in conjunction with a BERT-based discriminator [34]. Building on this concept, we developed the MMT-BERT model by utilizing MMT as the generator and Music-BERT as the discriminator.

3. METHODOLOGY

3.1 Proposed Symbolic Music Representation

In our approach, we introduce a novel symbolic music representation that incorporates chords. Table 1 shows differences between the conventional approaches and the proposed symbolic music representation. While most conventional representations omit details about chords, we focus on chord information-aware representation to facilitate the process of generating music that more closely resembles humans. First, we extract music data including chords and notes from MIDI files based on MuspyToolkit [35] and MusicLang. During the extraction process, we recognize a chord once per bar, i.e., every four beats. We exclude songs with a time signature other than 4/4, limit the number of chords in a bar to one, and ignore chord changes within a bar since MusicLang only detects chord changes once per bar. Each time MusicLang detects a chord change, it extracts the scale degree, tonality root, tonality mode, chord octave and extension note of the chord. After extracting chord and note information, we encode a piece of music into a sequence of quintuple tokens $\mathbf{X} = (\mathbf{x}_0, \dots, \mathbf{x}_{N-1})$, where \mathbf{x}_i and N denote the i -th quintuple token and the total number of quintuple tokens, respectively. Here, t represents the following event type: {start-of-song, instrument, start-of-score, note, chord, end-of-song}. The meanings of each event type are shown as follows:

- **Start-of-song:** The beginning of the music piece
- **Instrument:** An instrument used in the music piece

Quintuple tokens	Meaning
$\mathbf{x}_0^{\text{start-of-song}}$: (0, 0, 0, 0, 0)	Start of song
$\mathbf{x}_1^{\text{instrument}}$: (0, 0, 0, 0, 6)	Instrument : harpsichord
$\mathbf{x}_2^{\text{instrument}}$: (0, 0, 0, 0, 40)	Instrument : violin
$\mathbf{x}_3^{\text{instrument}}$: (0, 0, 0, 0, 42)	Instrument : cello
$\mathbf{x}_4^{\text{start-of-score}}$: (0, 0, 0, 0, 0)	Start of score
$\mathbf{x}_5^{\text{chord}}$: (1, 1, 7, 2, 9)	Chord: beat=1, degree=1, root=G, mode=minor, extension=9th
$\mathbf{x}_6^{\text{note}}$: (1, 1, 46, 6, 6)	Note : beat=1, position=1, pitch=A2, duration=6, instrument=harpsichord bar 1
$\mathbf{x}_{12}^{\text{chord}}$: (5, 1, 7, 2, 6)	Chord: beat=5, degree=1, root=G, mode=minor, extension=6th
$\mathbf{x}_{13}^{\text{note}}$: (5, 7, 48, 5, 42)	Note : beat=5, position=7, pitch=G2, duration=5, instrument=cello bar 2
$\mathbf{x}_{N-1}^{\text{end-of-song}}$: (0, 0, 0, 0, 0)	End of song

Figure 1. An example of the proposed representation. Compared to the conventional representation, the proposed representation incorporates an additional chord event (highlighted by red blocks) per bar, thereby aiding the model in understanding the relationship between the notes and chords.

- **Start-of-score:** The beginning of a sequence of musical events, including notes and chords
- **Note:** A note characterized by beat, position, pitch, duration, and instrument
- **Chord:** A chord characterized by beat, scale degree, root note, mode, and extension note
- **End-of-song:** The end of the music piece

The meaning of each element in the quintuple token \mathbf{x}_i^t varies depending on the event type t . The correspondence between the event type t and the meanings of the quintuple token \mathbf{x}_i^t is shown in Table 2. Additionally, it is noted that we apply different embeddings for the different features sharing the same axis. A schematic diagram of the proposed representation is illustrated in Figure 1. In this way, we can obtain a symbolic music representation that incorporates chords that is suitable for input into the aforementioned MMT-BERT architecture.

3.2 MMT-BERT Architecture

The fundamental structure of our MMT-BERT architecture is based on a GAN architecture, employing MMT as the generator and MusicBERT as the discriminator. The overview diagram of MMT-BERT is illustrated in Figure 2. The primary concept of GAN is minimizing the loss to enhance the generator’s ability to deceive the discriminator by producing fake music indistinguishable from real music, while simultaneously maximizing the discriminator’s accuracy in distinguishing between real and fake music. Details of the generator and discriminator will be discussed later.

3.2.1 Generator

As the generator, we employ MMT [6], a Transformer-XL-based model that consists solely of decoders. In MMT, elements in the quintuple token \mathbf{x}_i are individually embedded first, and then concatenated, followed by the addition of

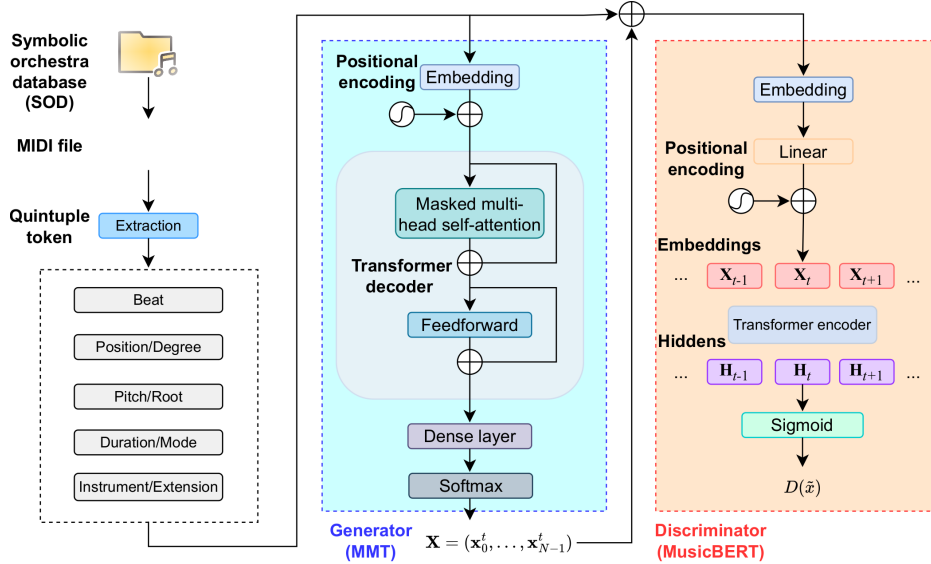


Figure 2. The diagram of the MMT-BERT. The generator is built upon Transformer-XL architecture, and the discriminator is built upon MusicBERT. A MIDI file from the dataset is firstly encoded into a sequence of quintuple tokens before fed into the model. Embeddings of the 6 elements are concatenated by a linear layer and converted into a single vector. Then they are fed into the encoder and decoder layers with the addition of position embeddings.

positional embeddings. Subsequently, this combined input is passed through transformer decoder blocks, which are composed of a masked multi-head self-attention layer and a feedforward layer. The output from the decoder blocks is then processed by a dense layer and a softmax layer, resulting in the generation of new music samples. MMT proves advantageous due to its capability to handle multi-dimensional input and output spaces, aligning perfectly with the requirements of symbolic music representation. Significantly, MMT can retain hidden states from previous segments, thereby eliminating the need for recalculating from scratch with each new segment. These retained states function as a memory aid for the current segment, establishing a recurrent connection between segments.

The application of MMT as the generator allows for instrument-controllable multitrack music generation with extended duration and higher training speed. Such a key feature facilitates the modeling of extensive long-range dependencies.

3.2.2 Discriminator

As the discriminator, we adopted MusicBERT [29], a large-scale Transformer model developed for symbolic music understanding. MusicBERT consists of a Transformer encoder and utilizes a masked language modeling approach where certain tokens in the input music sequence are masked and then predicted by the model output. The original proposed encoding method, called OctupleMIDI process transforms a symbolic music piece into a sequence of octuple tokens, each containing eight basic elements related to a musical note. In order to make MusicBERT act as a discriminator adapted to the proposed representation mentioned in Sec. 3.1, we refine the input and output format of MusicBERT. Quintuple tokens are converted into a single vector through the concatenation of embeddings and

a linear layer. The resulting vector is combined with position embeddings and provided as input to the Transformer encoder. To predict each of the five tokens within the quintuple, separate softmax layers are added to map the hidden states of the Transformer encoder to the vocabulary sizes of the different element types. MusicBERT’s proficiency in comprehending symbolic music as the discriminator integrates with MMT’s generation process, thereby aiding in the stability of the training process and faster convergence.

3.2.3 Relativistic Standard Loss

Inspired by RS-GAN [31], one of the state-of-the-art methods in GANs, we adopt the relativistic standard loss as our objective function. Applying relativistic standard loss prevents the network from becoming overconfident, leading to slower and more careful decisions, allowing the generator more room to adjust its weights and improve the training process [30]. The probability that the given fake data is more realistic than a randomly sampled real data is defined as follows:

$$D(\tilde{x}) = \text{sigmoid}(C(f) - C(r)), \quad (1)$$

where $C(\cdot)$ denotes a non-transformed layer, and \tilde{x} denotes real/fake data pairs $\tilde{x} = (r, f)$. Hence, the loss function of the generator G and the discriminator D are defined as follows:

$$L_G = \mathbb{E}_{(r,f) \sim p(r,f)} [\log(\text{sigmoid}(C(r) - C(f)))] - \sum_i r_i \log f_i, \quad (2)$$

$$L_D = \mathbb{E}_{(r,f) \sim p(r,f)} [\log(\text{sigmoid}(C(f) - C(r)))] \quad (3)$$

where r_i and f_i denote ground truth logits and generated music logits, respectively. It is noted that we add cross entropy to the loss function of generator in order to accelerate

the convergence process of the loss function. By training both the generator and discriminator with the relativistic standard loss to emulate human-like musical compositions, our MMT-BERT model can generate high quality music pieces that incorporate sophisticated chord information.

4. EXPERIMENT

4.1 Experiment Setup

In the experiment, we utilize the Symbolic Orchestral Database (SOD) [36], which comprises 5,864 music pieces encoded as MIDI files along with associated metadata. The dataset is partitioned into training, testing, and validation sets, receiving 80%, 10%, and 10% of the data, respectively. We set a temporal resolution of 12 time steps per quarter note for detailed timing accuracy. The Transformer-XL generator is composed of six decoder layers with 512 dimensions and eight self-attention heads, and the MusicBERT discriminator consists of one encoder layer with two self-attention heads. The maximum length for symbolic music sequences is set at 1024, with a maximum of 256 beats. To optimize the models, we employ the Adagrad optimizer to mitigate issues of gradient explosion and vanishing [37]. Additionally, to enhance the robustness of the data, we augment it by randomly transposing all pitches by $s \sim U(-5, 6)$ ($s \in \mathbb{Z}$) semitones and assign a starting beat. Here, U denotes a uniform distribution.

As comparative methods, we employ three state-of-the-art music generation models: MMM [21], FIGARO [23], and MMT [6]. We validate the performance of our MMT-BERT model by conducting quantitative evaluations using existing metrics and subjective experiments to assess the human-like qualities of the generated music pieces.

4.2 Quantitative Evaluation

Following [6], we evaluate the generated music pieces using four metrics: pitch class entropy similarity (PCES), scale consistency similarity (SCS), groove consistency similarity (GCS), and average length (AL). We consider higher values of PCES, SCS, and GCS as indicators of superior quality, while a higher AL denotes a greater capability to produce long-duration music pieces.

In preparation for calculating PCES, the pitch class entropy (PCE) is defined as follows:

$$\text{PCE} = - \sum_{i=0}^{11} h_i \log_2(h_i), \quad (4)$$

where h_i denotes the number of occurrences of each note name in the 12-dimensional pitch class histogram. As the PCE values increase, the tonality of the generated music pieces exhibits greater instability. However, it is important to recognize that more stable tonality does not necessarily imply higher quality. Subsequently, we calculate PCES between generated music samples and human compositions as follows:

$$\text{PCES} = 1 - \frac{|\text{PCE}_{\text{gen}} - \text{PCE}_{\text{tr}}|}{\text{PCE}_{\text{tr}}}, \quad (5)$$

where PCE_{gen} and PCE_{tr} denotes the PCE value of generated music samples and human compositions, respectively. Moreover, noticing that PCE is intrinsically linked to the volume of data, we truncate the generated musical pieces to the preceding k seconds and calculate their PCES.

The scale consistency (SC) is derived by calculating the proportion of tones that conform to a conventional scale and presenting the value for the most closely aligned scale [38]. SC serves as an indicator of the model’s proficiency in generating musical segments that demonstrate cognizance of chords and scales within the current bar. The SCS between generated music samples and human compositions is defined as follows:

$$\text{SCS} = 1 - \frac{|\text{SC}_{\text{gen}} - \text{SC}_{\text{tr}}|}{\text{SC}_{\text{tr}}}, \quad (6)$$

where SC_{gen} and SC_{tr} denote the SC values of generated music samples and human compositions, respectively.

To calculate GCS, we first define a groove pattern \mathbf{g} as a 64-dimensional binary vector. The groove consistency (GC) between two grooving patterns ($\mathbf{g}^a, \mathbf{g}^b$) is defined as follows:

$$\text{GC} = 1 - \frac{1}{Q} \sum_{i=0}^{Q-1} \text{XOR}(g_i^a, g_i^b), \quad (7)$$

where $\text{XOR}(\cdot, \cdot)$ denotes the exclusive OR operation, and g_i denotes a position in a bar at which there is at least a note onset. Q is the dimensionality of \mathbf{g}^a and \mathbf{g}^b . GC is a measure of music’s rhythmicity. The value of GC stands for the steadiness in rhythm of the generated music pieces. The GCS between generated music samples and human compositions is defined as follows:

$$\text{GCS} = 1 - \frac{|\text{GC}_{\text{gen}} - \text{GC}_{\text{tr}}|}{\text{GC}_{\text{tr}}}, \quad (8)$$

where GC_{gen} and GC_{tr} denote the GC values of generated music samples and human compositions, respectively.

AL denotes the mean duration of the generated music pieces, which collectively illustrates the model’s ability to generate musical sequences with significant length.

The results of the quantitative evaluation are shown in Table 3. To facilitate a fair comparison by standardizing the lengths of music pieces, PCES is assessed over a 15-second span due to the limitations of MMM and FIGARO in producing extended compositions. Experimental results show that MMT-BERT achieves higher performance in PCES, SCS, and GCS compared to the other methods, demonstrating its effectiveness in generating high quality music pieces. This achievement is attributed to its chord awareness and the symbolic music understanding facilitated by MusicBERT. MMT-BERT’s AL is marginally less than that of MMT, and this results from integrating chord events that are not converted to audio during the decoding phase. However, MMT-BERT’s AL significantly surpasses that of MMM and FIGARO, confirming its capability to generate longer compositions. Additionally, the AL of all the music pieces in the SOD we used, which also serve as

	PCES (%)	SCS (%)	GCS (%)	AL (sec)
MMM [21]	92.93±1.22	98.64±0.92	98.28±0.29	38.69
FIGARO [23]	94.33±0.31	98.70±0.22	<u>98.84±0.67</u>	28.69
MMT [6]	95.19±0.45	98.94±0.77	98.44±0.55	100.42
MMT-BERT w/o Chord event	95.57±1.32	98.81±0.23	99.56±0.32	<u>100.25</u>
MMT-BERT w/o MusicBERT	<u>96.22±0.44</u>	<u>99.14±0.29</u>	98.61±0.44	97.43
MMT-BERT (ours)	99.73±0.21	99.64±0.31	99.66±0.25	99.87

Table 3. Quantitative evaluation results. The **boldface** denotes the highest value, and the underlined denotes the second highest value, respectively.

	R	H	C	S	O
MMM [21]	3.83±0.92	3.78±0.87	3.78±0.73	3.67±0.84	3.83±0.79
FIGARO [23]	3.78±1.11	3.78±1.11	3.89±1.02	3.89±1.13	3.83±0.92
MMT [6]	3.22±0.70	3.17±0.98	3.44±1.03	3.33±1.09	3.22±0.78
MMT-BERT (ours)	3.55±0.94	3.55±0.92	3.33±0.98	3.39±0.90	3.44±0.80

Table 4. Subjective evaluation results. Each metric is rated on a five-point scale, with the average score being calculated.

the ground truth, is 99.88 seconds. Evaluation results show that MMT-BERT can produce music of higher quality than MMT, and of longer duration than MMM and FIGARO.

4.3 Impacts of Chord Event and Discriminator

MMT-BERT aims to generate more harmonious, more human-like music pieces through the addition of chord events and adversarial generative learning by employing MusicBERT as its discriminator. To evaluate aspects related to richness and humanness, we have conducted subjective experiment and ablation study.

In the subjective experiment, we asked 18 music amateurs as the following five questions and requested that they rated each on a five-point scale.

- **Richness (R):** Does the music piece have diversity and interestingness?
- **Humanness (H):** Does the music piece sound like it was composed by an expressive human musician?
- **Correctness (C):** Does the music piece contain perceived mistakes in composition or performance?
- **Structureness (S):** Does the music piece exhibit structural patterns such as repeating themes or the development of musical ideas?
- **Overall (O):** What is the general score of the music piece?

As mentioned in Sec. 4.2, FIGARO and MMM employ a music representation that considers percussive sounds and typically generates much shorter pieces. Therefore, the nature of the music pieces generated by these models, FIGARO and MMM, differs significantly from that of MMT-BERT and MMT due to their use of percussive sounds and shorter compositions. To fairly evaluate the human-like quality of the generated music pieces, we compared MMT-BERT with MMT, a state-of-the-art approach whose generated compositions have lengths and musical styles that are

relatively similar to those of MMT-BERT. Additionally, to ensure clarity in subjective evaluation, we included the results for MMM and FIGARO. The results of the subjective evaluation are shown in Table 4. Table 4 indicates that MMT-BERT scores are particularly high in both richness and humanness compared to MMT. This suggests that the application of chord events and MusicBERT contribute to the generation of music pieces that more closely resemble human compositions. On the other hand, regarding correctness, our method did not specifically aim to enhance this metric, which may cause the gap in this value. For the same reason, our method exceeds MMT by a small margin in structureness mainly because of uncertainty. Although there is no clear advantage between MMT and MMT-BERT in correctness and structureness, our method still outperforms MMT in richness and humanness. The overall score also proves the superiority of MMT-BERT, which indicates that chord events and MusicBERT enhance the ability to create music similar to that produced by humans.

The results of the ablation study are shown in Table 3 along with the quantitative evaluation results. It is evident that the addition of chord events improves PCES and SCS. MusicBERT also contributes to the enhancement of PCES and GCS.

5. CONCLUSION

In this paper, we have proposed the chord-aware symbolic music generation approach, named MMT-BERT. By extracting chord information from raw audio files, we have devised a chord-aware symbolic music representation. We also developed a novel RS-GAN architecture based on MMT and MusicBERT. Both experimental evaluations validate the efficacy of our method in producing music pieces of superior quality, enhanced human likeness, and considerable length. In future works, we plan to explore methods that refine musical structure and incorporate information from various musical modalities.

6. ACKNOWLEDGMENTS

This work was partly supported by JSPS KAKENHI Grant Numbers JP21H03456, JP23K11141 and JP23KJ0044.

7. REFERENCES

- [1] W. Liu, "Literature survey of multi-track music generation model based on generative confrontation network in intelligent composition," *The Journal of Supercomputing*, vol. 79, no. 6, pp. 6560–6582, 2023.
- [2] F. P. Brooks, A. L. Hopkins, P. G. Neumann, and W. V. Wright, "An experiment in musical composition," *IRE Transactions on Electronics Computers*, vol. 6, pp. 175–182, 1957.
- [3] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in Neural Information Processing Systems*, vol. 27, pp. 53–65, 2014.
- [4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, pp. 5999–6009, 2017.
- [5] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in Neural Information Processing Systems*, vol. 33, pp. 6840–6851, 2020.
- [6] H.-W. Dong, K. Chen, S. Dubnov, J. McAuley, and T. Berg-Kirkpatrick, "Multitrack music transformer," in *Proc. of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.
- [7] Y.-S. Huang and Y.-H. Yang, "Pop music transformer: Beat-based modeling and generation of expressive pop piano compositions," in *Proc. of the ACM International Conference on Multimedia (ACMMM)*, 2020, pp. 1180–1188.
- [8] F. Takuya, "Realtime chord recognition of musical sound: A system using common lisp music," in *Proc. of the International Computer Music Conference (ICMC)*, 1999, pp. 464–467.
- [9] J. Park, K. Choi, S. Jeon, D. Kim, and J. Park, "A bi-directional transformer for musical chord recognition," in *Proc. of the International Society for Music Information Retrieval Conference (ISMIR)*, 2019, pp. 620–627.
- [10] R. E. Scholz and G. L. Ramalho, "Cochonut: Recognizing complex chords from midi guitar sequences," in *Proc. of the International Society for Music Information Retrieval Conference (ISMIR)*, 2008, pp. 27–32.
- [11] E. Demirel, B. Bozkurt, and X. Serra, "Automatic chord-scale recognition using harmonic pitch class profiles," in *Proc. of the Sound and Music Computing Conference (SMC)*, 2019, pp. 72–79.
- [12] H. Jhamtani and T. Berg-Kirkpatrick, "Modeling self-repetition in music generation using generative adversarial networks," in *Proc. of the Machine Learning for Music Discovery Workshop, ICML*, 2019.
- [13] H. Zhang, L. Xie, and K. Qi, "Implement music generation with GAN: A systematic review," in *Proc. of the International Conference on Computer Engineering and Application (ICCEA)*, 2021, pp. 352–355.
- [14] S. Walter, G. Mougeot, Y. Sun, L. Jiang, K.-M. Chao, and H. Cai, "MidiPGAN: A progressive gan approach to midi generation," in *Proc. of the IEEE International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, 2021, pp. 1166–1171.
- [15] A. Muhamed, L. Li, X. Shi, S. Yaddanapudi, W. Chi, D. Jackson, R. Suresh, Z. C. Lipton, and A. J. Smola, "Symbolic music generation with transformer-gans," in *Proc. of the AAAI Conference on Artificial Intelligence*, 2021, pp. 408–417.
- [16] H.-W. Dong, W.-Y. Hsiao, L.-C. Yang, and Y.-H. Yang, "MuseGAN: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment," in *Proc. of the AAAI Conference on Artificial Intelligence*, 2018.
- [17] H.-W. Dong and Y.-H. Yang, "Convolutional generative adversarial networks with binary neurons for polyphonic music generation," in *Proc. of the International Society for Music Information Retrieval Conference (ISMIR)*, 2018, pp. 190–196.
- [18] W.-Y. Hsiao, J.-Y. Liu, Y.-C. Yeh, and Y.-H. Yang, "Compound word transformer: Learning to compose full-song music over dynamic directed hypergraphs," in *Proc. of the AAAI Conference on Artificial Intelligence*, 2021, pp. 178–186.
- [19] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Défossez, "Simple and controllable music generation," *Advances in Neural Information Processing Systems*, vol. 36, 2024.
- [20] S.-L. Wu and Y.-H. Yang, "The jazz transformer on the front line: Exploring the shortcomings of ai-composed music through quantitative measures," in *Proc. of the International Society for Music Information Retrieval Conference (ISMIR)*, 2020, pp. 142–149.
- [21] J. Ens and P. Pasquier, "MMM: Exploring conditional multi-track music generation with the transformer," *arXiv preprint arXiv:2008.06048*, 2020.
- [22] B. Yu, P. Lu, R. Wang, W. Hu, X. Tan, W. Ye, S. Zhang, T. Qin, and T.-Y. Liu, "Museformer: Transformer with fine-and coarse-grained attention for music generation," *Advances in Neural Information Processing Systems*, vol. 35, pp. 1376–1388, 2022.

- [23] D. von Rütte, L. Biggio, Y. Kilcher, and T. Hofmann, “FIGARO: Generating symbolic music with fine-grained artistic control,” in *Proc. of the International Conference on Learning Representations (ICLR)*, 2023, pp. 1–18.
- [24] J. Salazar, D. Liang, T. Q. Nguyen, and K. Kirchoff, “Masked language model scoring,” in *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*, Jul. 2020, pp. 2699–2712.
- [25] E. Montahaei, D. Alihosseini, and M. S. Baghshah, “Jointly measuring diversity and quality in text generation models,” in *Proc. of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation (NeuralGen)*, 2019, pp. 90–98.
- [26] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. V. Le, and R. Salakhutdinov, “Transformer-XL: Attentive language models beyond a fixed-length context,” in *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019, pp. 2978–2988.
- [27] T. Zhang*, V. Kishore*, F. Wu*, K. Q. Weinberger, and Y. Artzi, “Bertscore: Evaluating text generation with bert,” in *Proc. of the International Conference on Learning Representations (ICLR)*, 2020.
- [28] E. Montahaei, D. Alihosseini, and M. S. Baghshah, “Jointly measuring diversity and quality in text generation models,” in *Proc. of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation (NeuralGen)*, 2019, pp. 90–98.
- [29] M. Zeng, X. Tan, R. Wang, Z. Ju, T. Qin, and T.-Y. Liu, “MusicBERT: Symbolic music understanding with large-scale pre-training,” in *Proc. of the Association for Computational Linguistics: ACL-IJCNLP*, 2021, pp. 791–800.
- [30] A. Jolicoeur-Martineau, “The relativistic discriminator: a key element missing from standard gan,” in *Proc. of the International Conference on Learning Representations (ICLR)*, 2018.
- [31] C. Li, R. Fu, and Y. Liu, “Algorithm for generating tire defect images based on RS-GAN,” in *Proc. of the International Conference on Neural Information Processing (ICNIP)*, 2023, pp. 388–399.
- [32] J.-P. Briot, G. Hadjeres, and F.-D. Pachet, “Deep learning techniques for music generation—a survey,” *arXiv preprint arXiv:1709.01620*, 2017.
- [33] S. Ji, J. Luo, and X. Yang, “A comprehensive survey on deep music generation: Multi-level representations, algorithms, evaluations, and future directions,” *arXiv preprint arXiv:2011.06801*, 2020.
- [34] P. Neves, J. Fornari, and J. Florindo, “Generating music with sentiment using transformer-gans,” in *Proc. of the International Society for Music Information Retrieval (ISMIR)*, 2022, pp. 717–725.
- [35] H.-W. Dong, K. Chen, J. McAuley, and T. Berg-Kirkpatrick, “MusPy: A toolkit for symbolic music generation,” in *Proc. of the International Society for Music Information Retrieval Conference (ISMIR)*, 2020, pp. 101–108.
- [36] L. Crestel, P. Esling, L. Heng, and S. McAdams, “A database linking piano and orchestral midi scores with application to automatic projective orchestration,” in *Proc. of the International Society for Music Information Retrieval Conference (ISMIR)*, 2018, pp. 592–598.
- [37] J. Duchi, E. Hazan, and Y. Singer, “Adaptive sub-gradient methods for online learning and stochastic optimization.” *Journal of machine learning research*, vol. 12, no. 7, 2011.
- [38] O. Mogren, “C-RNN-GAN: A continuous recurrent neural network with adversarial training,” in *Proc. of the Constructive Machine Learning Workshop (CML) at NIPS*, 2016, pp. 1–6.

DISCOGS-VI: A MUSICAL VERSION IDENTIFICATION DATASET BASED ON PUBLIC EDITORIAL METADATA

R. Oguz Araz Xavier Serra Dmitry Bogdanov
Music Technology Group, Universitat Pompeu Fabra, Barcelona
{recepoguz.araz, xavier.serra, dmitry.bogdanov}@upf.edu

ABSTRACT

Current version identification (VI) datasets often lack sufficient size and musical diversity to train robust neural networks (NNs). Additionally, their non-representative clique size distributions prevent realistic system evaluations. To address these challenges, we explore the untapped potential of the rich editorial metadata in the Discogs music database and create a large dataset of musical versions containing about 1,900,000 versions across 348,000 cliques. Utilizing a high-precision search algorithm, we map this dataset to official music uploads on YouTube, resulting in a dataset of approximately 493,000 versions across 98,000 cliques. This dataset offers over nine times the number of cliques and over four times the number of versions than existing datasets. We demonstrate the utility of our dataset by training a baseline NN without extensive model complexities or data augmentations, which achieves competitive results on the SHS100K and Da-TACOS datasets. Our dataset, along with the tools used for its creation, the extracted audio features, and a trained model, are all publicly available online.

1. INTRODUCTION

Artists continue to cover, remix, and reinterpret musical works, creating a rich tapestry of musical versions that celebrate the originals. This proliferation presents a complex challenge: how to accurately identify different versions of a musical work within vast digital catalogs. Version identification (VI) addresses this problem using audio processing methods to find versions of query tracks in music catalogs [1–3]. VI has thus emerged as a crucial solution with significant implications across multiple applications including music discovery, musicological research, and copyright enforcement. From both the artists’ and copyright holders’ perspectives, VI has substantial importance as it offers a tool for financial compensation to many music industry stakeholders.

Recently, multiple datasets, all derived from scraping

the SecondHandSongs¹ website, were proposed for developing VI systems [4–7]. These datasets have facilitated the development of various systems based on convolutional neural networks (CNNs) [5–12]. However, their limited sizes have restricted the feasibility of employing larger architectures, such as transformers, which are increasingly utilized in other music information retrieval (MIR) tasks [13, 14]. Additionally, existing datasets such as Da-TACOS [5] and SHS100K [4] lack comprehensive metadata, such as genre, style, and release year, which can be useful for detailed performance evaluation and sophisticated training approaches. Furthermore, they fall short in presenting sufficient challenges regarding the distribution of clique sizes, genres, styles, and track durations.

This study introduces a significantly larger and more challenging VI dataset. Rather than relying on SecondHandSongs, we use public editorial metadata from the Discogs² database, which has not been explored in the field previously. Discogs is collaboratively maintained by music enthusiasts and professionals who submit detailed metadata about music releases, including artist details, release information, and extensive credit descriptions. These descriptions not only list track artists and writers but also provide aliases, name variations, and artist relationships, offering a rich framework for identifying versions.

Using this metadata, we propose a methodology for identifying a large dataset of versions and mapping this dataset to various music audio collections. The resulting dataset is the largest open-source VI dataset to date. Our contributions can be summarized as follows:

1. A metadata-only dataset, Discogs-VI, containing over 1,900,000 versions of around 348,000 works.
2. A subset of this dataset, Discogs-VI-YT, containing about 493,000 versions of around 98,000 works matched to YouTube URLs of official music uploads. It contains over nine times as many works and over four times as many versions as other datasets.
3. A larger and more challenging test set that contains other publicly available test sets.
4. A pre-trained baseline model, Discogs-VINet.

The dataset³, together with the tools for its creation, the extracted audio features, and the model trained on this data⁴, are publicly available online.

¹ <https://secondhandsongs.com/>

² <https://www.discogs.com/>

³ <https://mtg.github.io/discogs-vi-dataset/>

⁴ <https://github.com/raraz15/Discogs-VINet>

Dataset	Source	Cliques	Versions	MCS	ACS	mCS	A-URL	m-URL	OV	Content
covers80 [15]	private	80	160	2	2	2	-	-	-	Full audio, title, album, artist
YouTubeCovers [16]	YouTube	50	350	7	7	7	-	-	✗	Features (full track)
Da-TACOS [5]	SHS	1,000	13,000	13	13	13	1.0	1.0	✗	Features (full track), metadata
CoversDataset [6]	SHS	26,905	110,794	24	4	3	1.0	1.0	✗	Features (first 3 min)
SHS-100K [4]	SHS	9,999	116,353	387	12	8	1.0	1.0	✗	Title, artist
Discogs-VI-YT	Discogs	98,785	493,049	658	5	2	1.5	1.0	✓	Rich metadata, features (full track)
Discogs-VI	Discogs	348,796	1,911,611	1,837	6	2	-	-	-	Rich metadata

Table 1. Overview of publicly-available VI datasets. Da-TACOS refers to the benchmark subset, for which the 2,000 noise works are not reported as they do not form cliques. SHS refers to the SecondHandSongs website. MCS: maximum clique size; ACS: average clique size; mCS: median clique size; A-URL: average YouTube URLs per version; m-URL: median YouTube URLs per version; OV: use of official YouTube videos only. “-” denotes that the property is not applicable.

2. IDENTIFYING VERSIONS ON DISCOGS

Discogs database metadata has been previously used in other MIR tasks [17–19]. In this section, we describe the proposed methodology to identify versions and cliques using its metadata. The complete Discogs data is shared as monthly data dumps under a Public Domain license, making it easy to access. In our study, we used the July 2024 data dump.

Numerous metadata fields are provided for releases, tracks, and artists, some of which are relevant for VI. We use the track title, track artists, featuring track artists, release artists, track writer artists, and release writer artists metadata. The artist metadata contains unique artist IDs and provides information regarding group memberships, artist aliases, and artist name variations, which we use extensively. In addition, we include genre, style, record label, release format, release date, master release, and release country metadata that can be potentially useful.

2.1 Version finding from metadata

We use two critical pieces of information to establish the version relationship between two tracks: the track title and the track writer artists, indicated by the “Written-By” metadata field. Specifically, we consider two tracks with the same title and a shared writer artist as versions. This is a sufficient but not necessary condition since two tracks with different names can also be versions. Nonetheless, this condition facilitates finding a significant amount of cliques and versions from the database with high precision.

The search for cliques operates on a set of tracks from the database whose track titles are normalized by applying string processing. This includes transliterating Latin characters by removing diacritics, removing leading articles, replacing “&” with “and”, eliminating any text within parentheses, and removing punctuation marks. These steps aim to mitigate potential differences in metadata between different releases and eliminate mix or edit indicators enclosed in parentheses, e.g., “(Radio Edit)”, thus facilitating the process of identifying cliques. Later, such differences are considered for differentiating between versions.

Using the normalized track titles, we partition the set of tracks into disjoint subsets using exact string matching. Then, we further partition these subsets by the common

track writer relation to distinguish different cliques with the same title. To do so, we compile a set of writer artist IDs for every track. Given that an artist on Discogs may represent a group with several members, we extend our collection to contain all associated members and incorporate each artist’s known aliases and name variations. As a result of the two-step partitioning, tracks that have the same normalized title and share a track writer are joined in the same cliques. We opted for the shared writer approach because not all writers are consistently included in credits on some releases.

Once the cliques are formed, we identify different versions by the track or release artists. In cases where track artists metadata is available, it is used; otherwise, the release artists metadata is used. If there are featuring track artists, they are also included. Therefore, a set of tracks belonging to the same clique and performed by the same set of artists is defined as a version. After identifying the versions, we discard the cliques with only one version.

In previous VI datasets, versions are not treated as sets of tracks as in our dataset. This difference arises because Discogs often lists multiple releases for essentially the same version of a track, which may vary only by the year or country of the release. Without direct access to these releases, it is impossible to confirm their differences in advance. Therefore, we treat such tracks as identical versions. Remarkably, our dataset comprehensively includes a variety of version types as systematized in [20], including live versions, remixes, and radio edits, which add valuable diversity and potential utility.

The resulting dataset, Discogs-VI, contains numerous cliques and versions. Statistics about the dataset in comparison to other datasets are provided in Table 1.

2.2 Limitations

Due to the complex processes of composing, performing, and releasing music, along with issues related to incomplete or inaccurate metadata, there are potential issues related to our approach.

Title variability: Versions can have different names, e.g. “Moon Over Naples” is the original version of both “Spanish Eyes” and “Blue Spanish Eyes”. Due to having different names, our algorithm falsely places these versions into different cliques. To address this issue, comple-

mentary data from SecondHandSongs or a large language model with music history knowledge can be used.

Rule-based text matching: Even for a single language, capturing all syntactic variations with simple rules is difficult. Yet, the database contains many languages with different syntaxes. A music named-entity recognition model may help to resolve this issue.

Metadata ambiguity: “You’re My Everything” is credited to “Miles Davis” in some releases while to “Miles Davis and John Coltrane”, and to “The Miles Davis Quintet” in others. These credential differences often arise from practical or legal reasons associated with publishing music. However, we can not know beforehand if they are different versions using only metadata. To reduce duplicate versions, we treat them as the same version.

3. VERSION SEARCH IN YOUTUBE

Owing to its detailed metadata, Discogs-VI can be mapped to music audio catalogs or other metadata sources. For our research purposes, we use YouTube. To match the Discogs metadata of a version to the YouTube metadata of a video, we design a rule-based algorithm.

In the matching process, we only accept videos provided by an official distributor, which can be the artists themselves or third parties such as record labels. This approach is adopted because we expect the official uploads to have more accurate metadata and be more persistent on the platform over time. Consequently, our dataset is the only VI dataset containing official uploads exclusively. In addition, due to this selectivity, our algorithm demonstrates high retrieval accuracy.

Discogs provides YouTube URL annotations for some of the releases associated with versions. However, these annotations are not on the track level and they are rarely provided. For a unified approach, we instead query YouTube for all versions. The queries are created using the Discogs version metadata in the format “artist1, artist2 - track title”, and if featuring artist information is available, we concatenate “(featuring artist3)”. We then store the top five results for each query and apply our metadata-matching algorithm to all stored results, which allows alternative URLs for certain versions.

As a result, we successfully matched 34% of the versions of Discogs-VI to a YouTube URL. Between these matched versions, we were able to download 98% successfully, corresponding to 33% of the total versions. We then discarded the versions that were not downloaded and the cliques without at least two downloaded versions to create the Discogs-VI-YT dataset. It contains 26% of the versions and 28% of the cliques of Discogs-VI.

3.1 Metadata matching algorithm

From Discogs metadata, our algorithm utilizes the track title, track artists, or, if unavailable, the release artists, along with any featuring artists. From YouTube, it uses the video’s category, uploader, artist, description, duration, and title. We process the strings similarly to the method de-

scribed in Section 2.1, except that the punctuation marks and possible texts within parentheses are not deleted to identify different versions.

The algorithm initially checks if the video metadata contains the “Music” category and if the video is an official YouTube upload. We consider a video official under the following conditions: an artist or a label provided the video, which is indicated in the video description; the video uploader is an artist topic channel auto-generated by YouTube; or the Discogs artist name is the same as the video uploader’s. Videos with a duration longer than 20 minutes are discarded to deal with the potential but unlikely issue of tracks sharing their titles with their albums or EPs, which could lead to full-release audio downloads.

If a video metadata passes these controls, we use the title and artist information to decide a match. If two titles are equal, we use the artist information. If the titles do not match exactly, we apply some heuristics to strip the video title from any additional information related to remastering, HD, lyrics, etc., and re-attempt the match. We then compare all possible permutations to deal with video titles in the “artist1, artist2 - track title (featuring artist3)” format, using exact string matching. This approach makes the dataset less noisy at the cost of losing potential matches.

3.2 Limitations

Since search results and the availability of YouTube videos can be affected by geolocation, re-creating the dataset may yield differences.⁵ Moreover, some URLs may become unavailable in the future.⁶ To mitigate this issue, we provide multiple YouTube URLs per version when possible. Therefore, even if the main URL becomes inactive, numerous versions can still be recovered from alternative URLs. Furthermore, since we only include official uploads, the probability of a video disappearing should be lower than in other datasets. These features have not been considered in previous datasets that share YouTube URLs [4, 21].

Another limitation of our methodology is that less than 8% of versions are matched to the same YouTube URLs. Analysis showed that almost all of these versions are members of the same cliques. For the cliques that exhibit this issue, we manually kept one of the duplicate versions.

4. DATASET ANALYSIS

Following the methodologies described in Section 2 and Section 3, we created the Discogs-VI and Discogs-VI-YT datasets, respectively. Table 1 reports their sizes. The large amount of detailed metadata in Discogs-VI shows great potential: combined with an industrial-scale music audio catalog, it can create new possibilities for VI system development. Moreover, Discogs-VI-YT contains more clique and version audio than all the others combined, promising to boost model performance and generalization capability.

The range of clique sizes in our dataset is unparalleled by others in the field. The presence of cliques with many

⁵ We conducted YouTube queries from Barcelona, Spain.

⁶ The URLs were accessed between March 2023 and July 2024.

versions is beneficial for metric learning, as it provides numerous examples within each clique [22]. The average, median, and maximum clique sizes in the dataset indicate that the distribution has a long tail, with the weight concentrated on small clique sizes. Unlike other datasets, this distribution is highly representative of real use cases.

Figure 1 reports the genre distribution of Discogs-VI-YT, demonstrating significant coverage over 13 genres. The distribution of styles, which is included in the project repository, covers 512 styles from Mambo to Tech House. Importantly, such genre metadata opens new possibilities for developing and evaluating VI systems. Previous studies have not delved into genre and style analyses, leaving their effect on performance underexplored. Given that our dataset contains relatively reliable genre and style annotations⁷ such analysis is now possible [17].

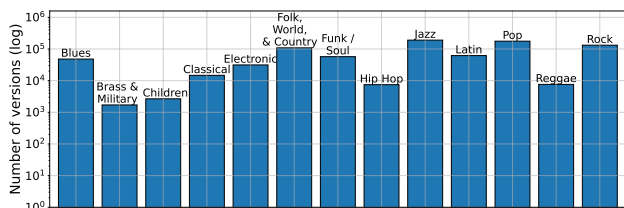


Figure 1. Discogs-VI-YT version genre distribution.

Table 2 compares the total number of artists of several VI datasets. Da-TACOS and SHS100K datasets provide only one artist per version while Discogs-VI offers multiple. For a consistent comparison, we count one artist per Discogs-VI version and do not include the group members. In addition, Da-TACOS noise works are not considered. The number of versions and artists comparisons between SHS100K and Discogs-VI-YT implies that our dataset contains more versions per artist on average.

Dataset	Artists
Da-TACOS	6,375
SHS100K	34,170
Discogs-VI-YT	67,345
Discogs-VI	239,949

Table 2. Number of track artist comparison between selected datasets. One artist per version is reported.

Figure 2 reports the audio duration distribution of Discogs-VI-YT, reflecting a comprehensive music collection. We observed that the long-duration tracks are mostly live versions and jazz or electronic music tracks, which can be notoriously long. Having long tracks increases the difficulty of training VI systems due to requiring effective time aggregation techniques or small embedding dimensions.

4.1 Development and test splits

We split the Discogs-VI-YT dataset into training, validation, and test sets. To increase the compatibility with other

⁷ Discogs genre and style annotations are release-level, however, they serve as a reasonable approximation for individual tracks.

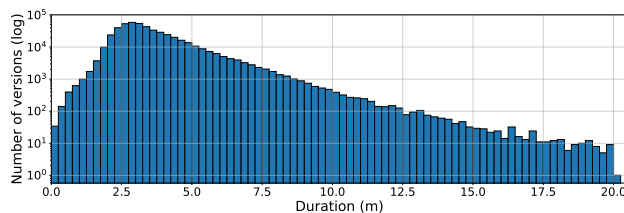


Figure 2. Discogs-VI-YT audio duration distribution.

datasets, the cliques in Discogs-VI that intersect with the Da-TACOS benchmark and SHS100K-Test sets are ensured to be part of our test set. We excluded CoversDataset from this consideration due to its lack of metadata.

To determine the intersection between our dataset and the Da-TACOS benchmark set, we conducted a thorough comparison of track titles and track writers using artist names, aliases, and name variations. We successfully identified 935 out of the 1,000 (93%) Da-TACOS cliques and 1,412 out of the 2,000 (71%) “noise” tracks. Given the detailed artist metadata we employed, it is unlikely that the unidentified works are included in our training set. Moreover, since Da-TACOS selects its “noise” tracks from those lacking alternate versions and our Discogs-VI consists exclusively of tracks with at least two versions, these tracks are also unlikely to be included in our training set. Regarding the SHS100K-Test set, we identified 1,555 out of the 1,692 cliques (90%). The union of the identified cliques from both datasets is reserved for our test set.

We aimed for a 90-10% development-test split; therefore, we sampled new cliques to add to the reserved cliques. While sampling the additional cliques, we did not exclude the SHS100K-Train set to use our dataset without restrictions. The reserved cliques from the Da-TACOS benchmark and SHS100K-Test sets had large enough sizes in our dataset. Moreover, similar to [7], we believe that having small-sized cliques in the test set simulates real use cases better. Therefore, we randomly sampled the additional cliques from sizes two to six. The remaining cliques were assigned to the development set and were further partitioned into training and validation sets following a 90-10% split. Figure 3 shows the clique size distribution of our splits, and Table 3 compares the split sizes of different datasets.

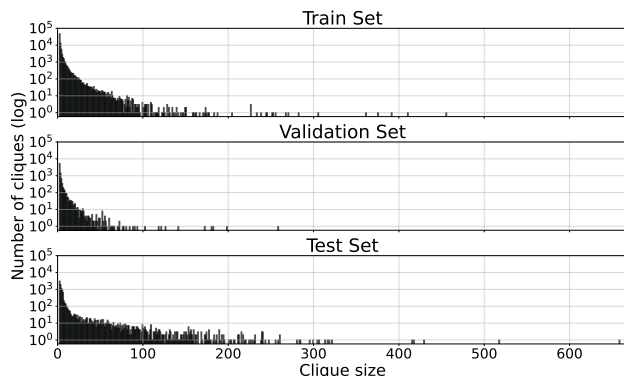


Figure 3. Discogs-VI-YT splits clique size distributions.

Dataset	Split	Cliques	Versions	MCS	ACS	mCS
Da-TACOS	Benchmark	1,000	13,000	13	13	13
	Noise	2,000	2,000	-	-	-
SHS100K	Test	1,692	10,547	162	6	5
	Validation	1,842	10,884	17	6	6
	Train	5,324	87,091	359	16	12
Discogs-VI-YT	Test	9,878	116,197	658	12	3
	Validation	8,890	37,081	258	4	2
	Train	80,017	339,771	455	4	2

Table 3. Dataset partition sizes. MCS: maximum clique size; ACS: average clique size; mCS: median clique size

4.2 Audio representations

We computed the following audio representations commonly used in VI systems: chroma, HPCP [2], and CQT [23]. They are available under request for non-commercial scientific research purposes.

5. BASELINE MODEL

To demonstrate the utility of Discogs-VI-YT we search for a baseline model that uses computationally inexpensive input representations and is feasible for training on a consumer-grade GPU.

TPP-Net [8] and its successor CQT-Net [10] rely on the classification loss for training. Due to the large number of cliques in Discogs-VI-YT, these models are difficult to train on this dataset without modifications. ByteCover [11], ByteCover2 [12], and LyraC-Net [24] are also difficult to train as they employ the classification loss with additional losses and feature complex architectures having significantly more parameters. Additionally, the code and pre-trained weights for these three models are not publicly available. We do not consider ByteCover3 [25] and CoverHunter [26] as they do not target full-track inputs. MOVE [9] and Re-MOVE [3] are not considered due to their reliance on computationally expensive input representations. Ultimately, we selected CQT-Net, primarily due to its adaptability for use with Discogs-VI-YT.

5.1 CQT-Net

The original model is trained with the classification task, where clique IDs of the SHS100K dataset are used as class labels. A multi-length training strategy that presents the model with three different segments from each version is used to reduce possible biases toward input duration. Additionally, tempo change and spectral masking data augmentation techniques are used. During retrieval, the classification head is discarded and the remaining network is used for extracting version embeddings, whose similarity is computed with cosine similarity.

5.2 Discogs-VINet

Training CQT-Net with classification loss is challenging due to the large number of cliques in Discogs-VI-YT. Therefore, we utilize the triplet loss, similar to previous

research [6, 9]. To this end, we remove the classification head from the architecture and change the affine projection layer to a linear projection with 512-dimensional outputs. Additionally, we include an L_2 normalization layer to ensure that embeddings lie on the unit hypersphere. The resulting model contains 5.2 million parameters.

At each training iteration, a mini-batch is created by randomly sampling 48 distinct cliques and two random versions per clique. With this configuration, each sample can only have one positive; hence, the positive mining strategy is equivalent to offline random sampling. For mining negatives, we use online hard-negative mining.

We extract the CQT input representations before training with CQT-Net’s setting. However, we store them with 16-bit precision due to the large storage requirement of our dataset. Unlike CQT-Net’s multi-length training strategy, we use fixed-length inputs where consecutive CQT frames of about 185 seconds are taken randomly. Then the features are mean downsampled with a factor of 20, following the authors. To demonstrate the benefits of using our large dataset, we do not use any data augmentation method during training, such as tempo and key modifications, spectral masking techniques, or audio degradation methods used in previous VI research.

We train Discogs-VINet for 50 epochs, which takes about 25 hours using a single Nvidia RTX2080. We use the AdamW optimizer, setting the initial learning rate to $1e-3$ and adjusting via exponential decay. The triplet loss margin is set to 0.1.

During training, we use our validation set to monitor performance. Every five epochs, we simulate the VI task and save the best model in terms of mean average precision (MAP). However, we evaluate the model at the end of the training on Discogs-VI-YT, Da-TACOS, and SHS100K datasets using MAP and the mean rank of the first relevant item (MR1) metrics.

5.3 Evaluation on Discogs-VI-YT

Due to potential overlaps between the training sets of publicly available VI models and the Discogs-VI-YT test set, we could not benchmark the publicly available models. For instance, as discussed in Section 4.1, there can be shared tracks with the SHS100K-Train set. Similarly, the Da-TACOS training set, which is not publicly available, may share tracks with our test set, rendering comparisons with models trained on this dataset unreliable. Additionally, as discussed in Section 5, training numerous models on Discogs-VI-YT were not possible. We acknowledge these limitations and suggest that benchmarking models is a critical area for future research.

Despite these challenges, we present the scores obtained by Discogs-VINet. Our model obtains a MAP score of 0.443 and an MR1 score of 614.1 on the Discogs-VI-YT test set, which establishes the baseline scores on this dataset. The contrast between the MR1 and MAP values can be attributed to the realistic clique size distribution. As shown in Figure 3, the test set contains numerous cliques with size two. When a query is made with a ver-

Training data	Model	d	Da-TACOS		SHS100K-Test		SHS100K-Test**	
			MAP \uparrow	MR1 \downarrow	MAP \uparrow	MR1 \downarrow	MAP \uparrow	MR1 \downarrow
Da-TACOS	MOVE [9]	4,000	0.495	48 [†]	\times	\times	\times	\times
	MOVE [9]	16,000	0.507	46 [†]	\times	\times	\times	\times
	Re-MOVE [3]	256	0.524	43 [†]	\times	\times	\times	\times
SHS100K-Train	TTP-Net [8]	300	\times	\times	0.465	72	\times	\times
	CQT-Net [10]	300	\times	\times	0.655	55	\times	\times
	ByteCover [11]	2,048	\times	\times	0.836	47	\times	\times
	ByteCover2 [12]	128	\times	\times	0.839	46	\times	\times
	ByteCover2 [12]	1,536	\times	\times	0.863	39	\times	\times
SHS100K-Train*	ByteCover [11]	2,048	0.714	23	\times	\times	\times	\times
	ByteCover2 [12]	128	0.718	23	\times	\times	\times	\times
	ByteCover2 [12]	1,536	0.791	19	\times	\times	\times	\times
SHS100K-Train**	LyraC-Net [24]	1,024	\times	\times	\times	\times	0.765	48
Private	LyraC-Net [24]	1,024	0.813	15	\times	\times	0.884	33
Discogs-VI-YT	Discogs-VINet	512	0.607	24	\times	\times	0.660	61

Table 4. Performance comparison on the Da-TACOS benchmark and SHS100K-Test sets. * denotes that the Da-TACOS benchmark set tracks were removed, ** denotes that the corresponding authors of that model downloaded the available URLs (therefore LyraC-Net [24] and Discogs-VINet are not evaluated on the same data), d denotes the embedding dimension, \times denotes that the result was not available, and [†] denotes the corrected calculations described in Section 5.4.

sion from these cliques, retrieving the only other version in high rankings contributes significantly to the MAP metric.

5.4 Evaluation on Da-TACOS and SHS100K

We tested Discogs-VINet on the Da-TACOS benchmark and SHS100K-Test sets. From the SHS100K-Test set, we could download 8,489 versions (80% of the total). As discussed in Section 4.1, we perform an extensive analysis to ensure that our training set has a minimal intersection with the evaluated sets.

The results are presented in Table 4, relying on results reported in the literature except for MOVE and Re-MOVE, for which we recomputed the results due to a metric calculation problem we discovered. In the public Da-TACOS evaluation script, "noise" works are wrongly boosting the MR1 score instead of being excluded. We corrected this issue, tested the official MOVE and Re-MOVE models, and listed the updated MR1 values.

In Table 4, Discogs-VINet outperforms both MOVE and Re-MOVE on the Da-TACOS benchmark set, which is a significant improvement given the simplicity of our input representation and lack of data augmentations. Unlike such, Discogs-VINet does not depend on pre-trained models for input representation. As a result, it exhibits significantly faster embedding extraction, similar to those reported in [12].

On the SHS100K-Test set, even though we used a slightly smaller subset due to some URLs becoming unavailable, we could not improve over other considered models, except for the TTP-Net and CQT-Net. In particular, CQT-Net, which we modified for our baseline, performed similarly. We posit that these differences may stem from the absence of data augmentation techniques in our methodology or from the classification loss possibly structuring the latent space more effectively than the triplet loss

we implemented. Nonetheless, further experiments are required.

ByteCover, ByteCover2, and LyraC-Net outperform Discogs-VINet by a significant margin. This performance difference can be attributed to several factors: the combined use of classification and triplet losses, as reported in the literature [27], the advantages obtained by training larger architectures, or the absence of data augmentations in our model. However, it is important to note that independent studies have raised concerns about the reproducibility of the published results associated with the ByteCover approach [24, 28].

6. CONCLUSION

We presented a new methodology to create a VI dataset from a previously unused metadata source, Discogs. Using this metadata, we identified a large number of cliques and versions to create the Discogs-VI dataset and matched a large portion of the versions with official YouTube URLs to create its Discogs-VI-YT subset. Our datasets surpass existing datasets by far in size and provide unprecedented metadata detailing genre, style, and artist relationships.

To demonstrate the utility of Discogs-VI-YT, we trained a baseline model, Discogs-VINet, on the training set and evaluated the model performance on the test set, establishing baseline results. Additionally, we assessed Discogs-VINet’s performance on the Da-TACOS benchmark and SHS100K-Test sets, where it demonstrated competitive performance. Notably, our model achieved these results without relying on any data augmentation techniques, multiple training losses, or complex architectural designs.

We leave training large models, using the metadata relations for training and evaluation, and investigating the role of data augmentations as future work.

7. ACKNOWLEDGEMENTS

This work is supported by “IA y Música: Cátedra en Inteligencia Artificial y Música” (TSI-100929-2023-1) funded by the Secretaría de Estado de Digitalización e Inteligencia Artificial and the European Union-Next Generation EU, under the program Cátedras ENIA.

8. REFERENCES

- [1] J. Serrà, M. Zanin, C. Laurier, and M. Sordo, “Unsupervised Detection Of Cover Song Sets: Accuracy Improvement And Original Identification,” in *Proc. of the 10th Int. Soc. for Music Information Retrieval Conf. (ISMIR)*, 2009.
- [2] J. Serrà, E. Gómez, and P. Herrera, “Audio Cover Song Identification and Similarity: Background, Approaches, Evaluation, and Beyond,” in *Advances in Music Information Retrieval*, Z. W. Raś and A. A. Wieczorkowska, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 307–332.
- [3] F. Yesiler, J. Serrà, and E. Gómez, “Less is more: Faster and better music version identification with embedding distillation,” in *Proc. of the 21st Int. Soc. for Music Information Retrieval Conf. (ISMIR)*, 2020.
- [4] X. Xu, X. Chen, and D. Yang, “Key-Invariant Convolutional Neural Network Toward Efficient Cover Song Identification,” in *IEEE Int. Conf. on Multimedia and Expo (ICME)*, 2018.
- [5] F. Yesiler, C. Tralie, A. Correira, D. F. Silva, P. Tovstogan, E. Gómez, and X. Serra, “Da-TACOS: A Dataset for Cover Song Identification and Understanding,” in *Proc. of the 20th Int. Soc. for Music Information Retrieval Conf. (ISMIR)*, 2019.
- [6] G. Doras and G. Peeters, “Cover Detection Using Dominant Melody Embeddings,” in *Proc. of the 20th Int. Soc. for Music Information Retrieval Conf. (ISMIR)*, 2019.
- [7] —, “A Prototypical Triplet Loss for Cover Detection,” in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.
- [8] Z. Yu, X. Xu, X. Chen, and D. Yang, “Temporal Pyramid Pooling Convolutional Neural Network for Cover Song Identification,” in *Proc. of the 28th Int. Joint Conf. on Artificial Intelligence*, 2019.
- [9] F. Yesiler, J. Serrà, and E. Gómez, “Accurate and Scalable Version Identification Using Musically-Motivated Embeddings,” in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.
- [10] Z. Yu, X. Xu, X. Chen, and D. Yang, “Learning a Representation for Cover Song Identification Using Convolutional Neural Network,” in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.
- [11] X. Du, Z. Yu, B. Zhu, X. Chen, and Z. Ma, “Bytecover: Cover Song Identification Via Multi-Loss Training,” in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2021.
- [12] X. Du, K. Chen, Z. Wang, B. Zhu, and Z. Ma, “Bytecover2: Towards Dimensionality Reduction of Latent Embedding for Efficient Cover Song Identification,” in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.
- [13] T. Zeng and F. C. M. Lau, “Training audio transformers for cover song identification,” *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2023, no. 1, p. 31, Aug. 2023.
- [14] P. Alonso-Jiménez, X. Serra, and D. Bogdanov, “Efficient Supervised Training of Audio Transformers for Music Representation Learning,” in *Proc. of the 24th Int. Soc. for Music Information Retrieval Conf. (ISMIR)*, 2023.
- [15] D. P. W. Ellis, “The covers80 cover song data set,” 2007. [Online]. Available: <https://labrosa.ee.columbia.edu/projects/cover songs/cover s80/>
- [16] D. F. Silva and V. M. A. Souza, “Music Shapelets For Fast Cover Song Recognition,” in *Proc. of the 16th Int. Soc. for Music Information Retrieval Conf. (ISMIR)*, 2015.
- [17] D. Bogdanov, A. Porter, H. Schreiber, J. Urbano, and S. Oramas, “The Acousticbrainz Genre Dataset: Multi-Source, Multi-Level, Multi-Label, And Large-Scale,” in *Proc. of the 20th Int. Soc. for Music Information Retrieval Conf. (ISMIR)*, 2019.
- [18] D. Bogdanov and X. Serra, “Quantifying Music Trends And Facts Using Editorial Metadata From The Discogs Database,” in *Proc. of the 18th Int. Soc. for Music Information Retrieval Conf. (ISMIR)*, 2017.
- [19] P. Alonso-Jiménez, X. Serra, and D. Bogdanov, “Music Representation Learning Based on Editorial Metadata from Discogs,” in *Proc. of the 23rd Int. Soc. for Music Information Retrieval Conf. (ISMIR)*, 2022.
- [20] J. Serrà, “Identification of Versions of the Same Musical Composition by Processing Audio Descriptions,” Ph.D. dissertation, Universitat Pompeu Fabra, Spain, 2011.
- [21] L. A. Lanzendörfer, F. Grötschla, E. Funke, and R. Wattenhofer, “DISCO-10M: A large-scale music dataset,” in *Proc. of the 37th Int. Conf. on Neural Information Processing Systems*, 2024.
- [22] F. Schroff, D. Kalenichenko, and J. Philbin, “FaceNet: A unified embedding for face recognition and clustering,” in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015.

- [23] C. Schörkhuber and A. Klapuri, “Constant-Q transform toolbox for music processing,” in *Proc. of the 7th Sound and Music Computing Conf.*, 2010.
- [24] S. Hu, B. Zhang, J. Lu, Y. Jiang, W. Wang, L. Kong, W. Zhao, and T. Jiang, “WideResNet with Joint Representation Learning and Data Augmentation for Cover Song Identification,” in *Proc. Interspeech*, 2022.
- [25] X. Du, Z. Wang, X. Liang, H. Liang, B. Zhu, and Z. Ma, “Bytecover3: Accurate Cover Song Identification On Short Queries,” in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2023.
- [26] F. Liu, D. Tuo, Y. Xu, and X. Han, “CoverHunter: Cover Song Identification with Refined Attention and Alignments,” in *IEEE Int. Conf. on Multimedia and Expo (ICME)*, 2023.
- [27] H. Luo, Y. Gu, X. Liao, S. Lai, and W. Jiang, “Bag of Tricks and a Strong Baseline for Deep Person Re-Identification,” in *IEEE/CVF Conf. on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019.
- [28] K. O’Hanlon, E. Benetos, and S. Dixon, “Detecting Cover Songs with Pitch Class Key-Invariant Networks,” in *IEEE 31st Int. Workshop on Machine Learning for Signal Processing (MLSP)*, 2021.

WHO'S AFRAID OF THE 'ARTYFYSHALL BYRD'? HISTORICAL NOTIONS AND CURRENT CHALLENGES OF MUSICAL ARTIFICIALITY

Nicholas Cornia
Orpheus Instituut

nicholas.cornia@orpheusinstituut.be

Bruno Forment
Orpheus Instituut

bruno.forment@orpheusinstituut.be

ABSTRACT

The meteoric surge of AI-generated music has prompted significant concerns among artists and publishers alike. Some fear that the adoption of AI is poised to result in massive job destruction; others sense it will jeopardize and eventually upend all legal frameworks of intellectual property. AI, however, is not the first instance where humanity has confronted the prospect of machines emulating musical creativity. Already in the Baroque, various modes of musical artificiality were explored, ranging from automata and organ stops mimicking human performance and natural sounds, up to devices for mechanized composition (e.g., Athanasius Kircher, Johann Philip Kirnberger, C.P.E. Bach, Antonio Calegari and Diederich Nickolaus Winkel). Valuable insights emerge from the reconsideration—and digital implementation—of these curiosities through the lens of present-day generative models. It can be argued that the very notion of ‘artificiality’ has presented humanity with long-standing philosophical dilemmas, in addressing the debate on the role of art as a substitute of (divine) nature. By digitally implementing and formalizing some pioneering instances of algorithmically-generated music we wish to illustrate how mechanical devices have played a role in human art and entertainment prior to our digital era.

1. INTRODUCTION

The rise of AI-generated music has sparked considerable concern among both artists and publishers. Some worry that the integration of AI technology may lead to widespread job displacement, while others foresee potential threats to existing legal structures governing intellectual property rights. The very notion of ‘artificiality’ has a decidedly negative ring to most people, evoking feelings of distrust, inauthenticity, and deviations from the ‘natural’ or ‘genuine.’ This can be attributed to the Platonic tradition. In *The Republic* (c. 375 BCE), Book X, Plato famously criticised the act of imitation (*mimesis*) in art and poetry as the ‘copy of a copy,’ merely satisfying the inferior senses and base pleasures, and lacking connections

with truth, virtue, or other higher ideas. The imitator, Plato contended, was a person who “has neither knowledge nor right opinion about whether the things they make are fine or bad.” [1, p. 1206]

But art is of course ‘artificial’ by its very nature, as a cultural expression, and vice versa: all artificiality requires art. Like ‘artifact’ and ‘artifice,’ ‘artificiality’ combines the Latin noun *ars* with the verb *facere* into one expression which means ‘doing art.’ ‘Art,’ consequently, can be understood as something so well-made (or ‘artful’) that it can substitute for the real or natural, which it is inseparably paired with. Artificiality, in this sense, does not need to possess any pejorative connotation; it simply amounts to ‘art’ or ‘artistry’ itself. As man-made contraption, an artifice demands art, being the craftsmanship or ‘science’ required to entice the beholder or listener through its mimicry. The past teaches us important lessons in this regard.

Artificiality, or “Nature’s Changeling,” [2, p. 51] as Margaret Cavendish termed it in *The Blazing World* (1666), has long fascinated humanity for providing an illusion of divine creation. The idea of building an alternative reality, which can be controlled by its human creators, has appealed to artists, scholars, and musicians through the ages. In particular in the long Baroque (c. 1550–1800) ‘artificial’ even denoted anything that was ‘artful.’ When, for example, the English diarist John Evelyn (1620–1706) visited the royal park of Brussels, on 8 October 1641, he marveled at “*artificial* cascades, rocks, grots” and a “grot of more neat and costly materials, full of noble statues, and entertaining us with *artificial* music.” [3, p. 37] In 1635, the French literary critic Jean Chapelain (1595–1674) contended that:

imitation in all poems, must be so perfect that *no difference appears between the thing imitated and that which imitates* [emphasis added], for the principal effect of the latter consists in proposing to the mind, in order to purge it of its unbridled passions, the objects as true and present”. [4, p. 115]

The Italian painter and architect Federico Zuccaro (1539–1609), furthermore, distinguished three types of design: natural (implying the imitation of nature), artificial (being a stylized distortion of nature), and fantastic-artificial (producing images of an entirely imaginary and unusual kind). [5] In sum, the Baroque revelled in artifi-



© N. Cornia and B. Forment. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** N. Cornia and B. Forment, “Who’s afraid of the ‘Artyfyshall Byrd’? Historical notions and current challenges of musical artificiality”, in *Proc. of the 25th Int. Society for Music Information Retrieval Conf.*, San Francisco, United States, 2024.

ciality, hailing the *trompe l'œil*, masquerade, automaton, and other sorts of mimicry as pinnacles of art.¹ [6, p. 10]

The Baroque did not perceive anything deceptive per se about artificiality, as long as not the mimicry itself—the relationship between artifice and nature—and the methods to obtain it were denied. Thus, François Hédelin, abbé d'Aubignac (1604–1676), argued in *La Pratique du théâtre* (1657) that spectators in the theatre knew all too well they were tricked when being “shown a new heaven, a new Earth, and an infinity of wonders that we believe to be present, at the very time we are quite sure we are being deceived.”² Conscious of the fact they were beholding painted canvases, handled by mechanical equipment, they relished the thought of artists producing such wonders. In a similar vein, Francis Bacon (1561–1626) included “all manner of feats of juggling, false apparitions, impostures, and illusions” in Salomon’s house, the utopian research institute evoked in *New Atlantis* (publ. posth., 1627):

[a]nd surely you will easily believe that we, that have so many things truly natural which induce admiration, could in a world of particulars deceive the senses, if we would disguise those things and labour to make them seem more miraculous. *But we do hate all impostures, and lies*; [emphasis added] insomuch as we have severely forbidden it to all our fellows, under pain of ignominy and fines, that they do not show any natural work or thing, adorned or swelling; but only pure as it is, and without all affectation of strangeness. [7, p. 40]

Consequently, the Baroque accepted and even actively endorsed methods of replicating nature as expressions of supreme craftsmanship, but it demanded that the mechanics of those “miraculous” devices be fully acknowledged and revealed.

It was only in the nineteenth century, as ‘authority’ and ‘originality’ emerged as core values of a “new code of artistic morality,” [8, p. 319] that a shift occurred in the understanding of art. This transformation altered the perception of the artwork from a handcrafted, artisanal product—an ‘artifice’—into a cerebral, isolated, and unique expression of genius. To replicate something came to be seen as an act of unoriginality, forgery, or plagiarism, [9] while technologies for mechanical reproduction (including photography, audio recording, and cinematography) were held responsible for the destruction of art’s ‘aura.’ [10] Plato returned with a vengeance.

In what follows, we will revisit the Baroque, and more particularly the devices for mechanised music composi-

¹ German Bazin argued that “Perhaps the most surprising feature of Baroque art,” the art historian and former Louvre curator Germain Bazin argued, is how the artists “who in thought and deed created new worlds could indulge in childish games of make-believe. One might pretend to be Apollo, Rinaldo, the Grand Turk, or even Confucius, but never simply oneself...”

² “on nous montre un nouveau Ciel, une nouvelle Terre, & une infinité de merveilles que nous croyons avoir présentes, dans le temps même que nous sommes bien assurés qu’on nous trompe.”

tion through which it explored artificiality in music. In discussing and digitally implementing a select number of these curiosities, our intention is not necessarily to engage in history for the sake of history itself, but rather to gain *transhistorical* insights into the workings and ethics of generative models in music composition.

By digitally implementing and formalizing some pioneering instances of algorithmically-generated music we wish to illustrate how mechanical devices have played a role in human art and entertainment prior to our digital era.

2. A SELECTED HISTORY OF GENERATIVE MODELS IN MUSIC

Whenever mechanical music is mentioned, one naturally thinks of our latest inventions, of the most highly perfected products of a technical, industrial age. [11]

The opening of 1934 article by Hugo Leichtentritt on mechanical musical instruments is an instructive example of how humanity has regularly confronted itself with cultural changes caused by technological progress, such as the early 20th century media revolution of radio broadcasting, movies and musical recordings. [12] Breakthrough technologies, such as the printing press, musical automata and clockworks, and audio recordings have always transformed artistic practice into new, unforeseen modes of expression. For instance, musical styles such as hip-hop, electronic dance music and *musical collages* such as Luciano Berio’s *Sinfonia* (1968), [13] laid their foundation on the possibility to repeat, transform, assemble and interact with pre-recorded material.

In a similar fashion, watching automata playing music in action, ingeniously designed using programmed cylinders and cogwheels mechanisms, [14] must have been an unimaginable experience for our forerunners, only comparable to our modern wonder for AI tools. These devices were able to entertain their public with musical pieces composed on the spot without any apparent human intervention.

We can even reassess Henry Purcell’s famous “Wonderous Machine” bass aria from *Ode for St Cecilia’s Day* Z. 328, reinterpreting the lyrics through the lens of an impatient Baroque musician (in this case a lute player) confronting themselves with the infinite possibilities of indefatigable mechanical devices:

*Wondrous machine!
To thee the warbling lute,
though used to conquest,
must be forced to yield,
with thee unable to dispute.*

The voice and instrumental accompaniment’s patters seem to emulate the *perpetuum mobile* of mechanically driven musical instruments, the like of which are described in later treatises like Engramelle’s *La Tonotechnie ou l’art de noter les cylindres* (1775) or ambitious implementations such as Diederich Nikolaus Winkel *Componium* (1821), a



Figure 1. Detail from the opening engraved page of Marie-Dominique-Joseph Engramelle *La Tonotechnie ou l'art de noter les cylindres* (1775).



Figure 2. Opening ground bass, accompaniment of the two oboes and singing voice dotted diminutions of respectively measures 1-2, 3-4 and 15-16.

mechanical device able to play an almost endless amount of variations on a pre-programmed piece of music. [15]

But how to translate a highly complex activity, such as music, into an algorithmic procedure? The act of music-making, either planned by a composer or made *ex tempore* by an improviser, arises from selecting musical gestures from an associative knowledge base stored in the musician's long-term memory. [16] For centuries, musicians have built such repositories, organising the vast palette of musical gestures, or schemata, through various systems of classification. [17] Tables, decision trees and voice-leading matrices have helped musicians to create a repertoire of melodic, harmonic and rhythmic patterns reflecting their contemporary musical style and performance practice.

Archetypical musical schemata were represented by rules, such as Thomas Campion's procedure for four-voice harmonisation of a given bass line. [18, p. 1-8] Moreover, Pietro Cerone encyclopaedic work *El melopeo y maestro*, [19] provided an endless series of musical tables and examples, similar in fashion to our modern "training sets" for AI models, that musicians internalised in their long term-memory, ready to be used during improvisation or composition of new pieces. [20]

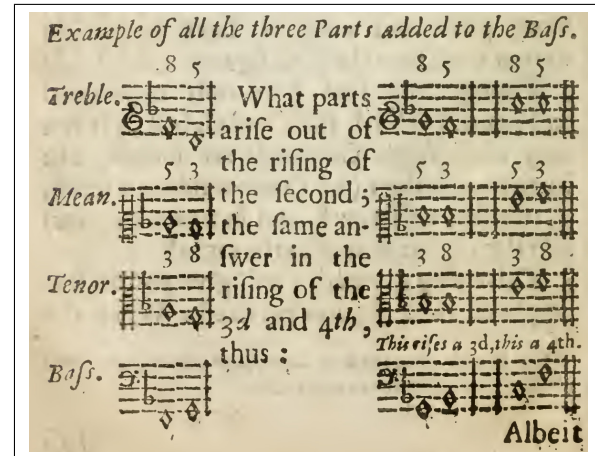


Figure 3. Voice-leading rules four four-voice harmonisation of a given bass melody. The procedure is based on the bass movements and relative consonances between the upper and lower voices.

Several treatises, like Giovanni Battista Chiodino's *Arte Pratica Latina et Volgare di far contrapunto a mente, et a penna* (1610), focussed on contrapuntal patterns that could be used by musicians to harmonise a given melody or bassline, while others, like Francesco Rognoni *Selva de varii passaggi* (1620), provided the students with complex rhythmic patterns for ornamenting melodic lines and cadential formulas not very dissimilar to 20th-century collections like Nicolas Slominsky's *Thesaurus of Scales and Melodic Patterns* (1947) or Jerry Coker's *Patterns for Jazz* (1970).

Of particular interest for our discussion is the 'Arca Musarithmica', a computational device designed by the German polymath Athanasius Kircher (1602–1680) and described in the second volume of his *Musurgia Universalis* (1650). [21] Kircher's compositional tool generates four-voice homophonic and polyphonic harmonisations (respectively named *contrapunctus simplex* and *floridus*) on the basis of a given set of verses and a musical scale, according to the contemporary Renaissance theory of authentic and plagal modes. The machine was designed to generate hymns for Jesuit missionaries working in religious communities outside Europe: thanks to Kircher algorithm, the priests could easily generate music from a liturgical text in the native language of their communities and compose the music according to the "affect" of the verses. [22] As if anticipating Purcell's "Wonderous Machine," the author describes the algorithm as "wondrous music" (*musurgiae mirificae*), referring to the device's capacity to instill wonder (*meraviglia*) in listeners and composers (or operators) alike. ³ To the best of our knowledge, Kircher is one of the first to use an abstract representation of the four-voice counterpoint: he assigned numerals to the scale's relative degrees and provided tables of rhythmic patterns that

³ A recent digital implementation of the *arca* has been made by Andrew A. Cashner of University of Rochester. The code is publically available on GitHub at <https://github.com/andrewacashner/kircher> while a web-based application can be found at <https://arca1650.info>

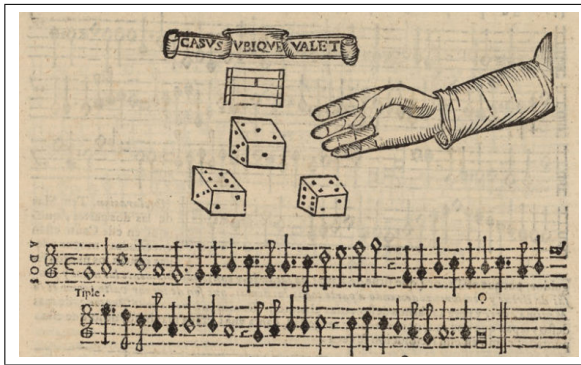


Figure 4. An earlier example of the use of dice and randomness in musical composition is Pietro Cerone *Enigma de la suerte, ò de los dados* appearing in his *El melopeo y maestro* (1613), pag. 1124. [31]

could be independently assembled with each other. A more "serialistic" approach can be found in the Anonymous treatise *Ludus Melothedicus* (1758), where the author created a series of numerical tables, where each number corresponded to a chromatic pitch and a given duration.

A detailed analysis of Kircher's voice-leading patterns reveals that these four-voice harmonies conform to typical 16th century musical schemata and chord progressions. Many of these progressions are based on counterpoint rules encoded in other treatises, like the one of Thomas Campion, the musical examples of Vicente Lusitano *Introductione facilissima, et novissima, ...* (1553), [23] Thomas Morley *A Plain and Easy Introduction to Practical Music* (1597) [24] and Tomás de Santa María *Libro llamado arte de tañer fantasía* (1565) [25]. [26]

Furthermore, the idea of encoding contrapuntal structures in a series of mathematical operations has found several resonances in the works of C.P.E. Bach *Einfall einen doppelten Contrapunct in der Octave von 6 Tacten zu machen* (1757), [27] the stylistic analysis of "Palestrina style" counterpoint of Serge Taneiev *Convertible Counterpoint in the Strict Style* (first publication, 1909) [28] and in the theories of melody, harmony and rhythm of 18th century mathematician Leonard Euler. [29]

A step ahead of pure deterministic rules, was the introduction of randomness in the compositional process, reducing the musician's agency on the generated artifact. The development of combinatorics, with its first application to music theory can be found in the early 17th century works of Kircher and Marin Mersenne, musicians explored the possibility of generating music from a series of exemplars through a randomized process, usually implemented by appending musical fragments according to numerical tables and dice rolling. To the best of our knowledge, the first published "dice game" (*würfelspiel* in German) is Johann Philipp Kirnberger *Der allezeit fertige polonoisen- und menüettencomponist* (1757), providing random tables to generate popular dance music (a polonaise and a menuet) for two violins and harpsichord accompaniment. [30]

In the coming decades, many other musicians imitated



Figure 5. Ingestion of lyrics into the generated music in Calegari's *Gioco Pitagorico Musicale*.

Kirnberger's curious experiment to generate music in an algorithmic fashion, leaving the user's agency to pure chance. The best known of these "dice games" is probably *Anleitung so viel Walzer oder Schleifer mit zwei Würfeln zu componiren*, attributed to W.A. Mozart and published by Nikolaus Simrock around 1790. Its fame is so great that a digital implementation of the compositional device had already been developed by David Caplin in 1955. [32,33]

Of particular interest is Antonio Calegari's *Gioco Pitagorico Musicale* (1801), which provides a framework for including lyrics to the generated airs and duets. In the title the author states clearly that the game is designed "for people without any knowledge of music",⁴ willing to amuse themselves at home with a seamless infinite combination of songs in the then current operatic style. [34]

A similar statement, made in the introduction of Andrea Mangeruva's *Nuovo Metodo per Comporre Migliaja di Walser* (1839), where the author designed a complicated randomised procedure based on modular arithmetic, encourages the use of the book for domestic music-making and amusement but warns the reader about the "seriousness" of his device: according to Mangeruva a "mechanical musician" (*un musico meccanico*) cannot aspire to "true music" (*la vera musica*), making an analogy between rules and procedures of prosody with the art of poetry. [35, p. 4] Unfortunately, Mangeruva's treatise is nothing more than a plagiarism of a 1811 French publication *Barême musical, ou l'Art de composer la musique sans en connaître les principes* attributed to Italian composer Gioseffo Catrufo. [30]

Many of these publications address a specific facet of music-making, namely amusement and entertainment. Is not by chance that these "dice games" were mainly used to generate popular music, in the form of songs and dances. Furthermore, we have noticed how many publishers have attributed their publications to famous composers, such in the case of the *Gioco Filharmonico*, attributed to Joseph Haydn by Luigi Marescalchi in 1793. Misattributions, rearrangement and even unauthorized reprints have been surprisingly common in the genre, as previously stated in the

⁴ "Col quale potrà Ognuno, anco senza sapere di Musica, formarsi una Serie quasi infinita di piccole Ariette"



Figure 6. Musical table from Andrea Mangeruva *Nuovo Metodo*.

instance of Mangeruva's "borrowing" from *Barème musical*.

Concerning the algorithms, they are mostly based on a series of basic musical variations of a piece, like a menuet or countrydance, composed beforehand by the author. Afterwards, a series of puzzles, enigmas, and randomizations are used as expedients to deceive the user, keeping the illusion that the procedure must be of some kind of magic. In several manuscript sources of early "dice games" the term *cabala* is often used,⁵ alluding to the duality between modern science, in the nascent theory of probability, and proto-scientific disciplines like alchemy and astrology. [36]

3. DIGITAL IMPLEMENTATIONS

A series of digital implementations of the treatises described in our paper are publically available on our GitHub repository.⁶ Alongside the Python code, we are providing the digital images of the discussed treatises and a small dataset of musical examples in LilyPond⁷, MIDI and PDF format, both from the generated music as for the input exemplars. The transcriptions of each musical fragment could be used as ground truth for Optical Music Recognition tasks involving the transcription of individual measures, both for printed as for handwritten music. [37] Furthermore, this unique musical corpus might be used in future research as baselines for evaluating generative models emulating 18th century Western classical music. A detailed list of pre-digital generative models for music can be found on the aforementioned *Artyfyshall Byrd* GitHub.

4. CONCLUSIONS

The present article wishes to present the current discussion on Artificial Intelligence and music from an historical perspective. The desire to artificially emulate nature

⁵ Several 18th century musical dice games refer explicitly to the Jewish kabbalah in their title and content, such as Johann P. Kirnberger *Cabala per componendi minuetti*, Bernardo Ottani *Tavola per la Cabala* and the anonymous *Musicalische Cabala* preserved in the National Library of France. For a detailed list of treatise visit our GitHub repository.

⁶ <https://github.com/NicholasCorniaOrpheus/Artyfyshall-Bird>

⁷ <https://lilypond.org/>

is a fascinating feature of human beings, and can find its roots in history, as well as in myths like Pygmalion, described in Ovid's *Metamorphoses* (c. 8 CE), Book X. [38, p. 128-148] With the technological developments of the Modern Period we have increasingly refined our craft to a point where the differences between the 'artificial' and the 'natural', between the 'authentic' and the 'forged', are almost impossible to discern. [39] On the other hand, the challenges afforded by technology and its artificial devices encourage us to reconsider the meaning of creativity and the role of art in our culture. [40] New technologies pose a "challenge to the imagination" for composers and performers, [41] extending the boundaries of human's creative effort. This statement is still valuable to our modern "wondrous" times, where the dreams of Leonard Euler [42] and Ada Lovelace⁸ [14] to mathematically encode every facet of music so that a machine could generate new pieces have become a tangible reality. Studing what it meant for our forerunnes to interact with the wonders of *musurgiae mirificae* can help us frame the current issue from a historical, dialectical perspective.

5. ACKNOWLEDGMENTS

We are particulary grateful to the library staff of the Conservatory of Venice Benedetto Marcello (Paolo da Col and Silvia Urbani) for the digital images of Mangeruva and Calegari treatises. We are also thankful to the Abjad developers for providing a useful Python library compatible with the Lilypond music encoding syntax [43].

6. REFERENCES

- [1] Plato, *Plato: Complete Works*, J. M. Cooper and D. S. Hutchinson, Eds. Hackett Publishing, 1997.
- [2] M. Cavendish, *The Blazing World and Other Writings*. Mint Editions, 2021.
- [3] J. Evelyn, *The Diary of John Evelyn*. Dent; Dutton, 1952, vol. I.
- [4] J. Chapelain, *Lettre Sur Les Vingt-Quatre Heures*, A. C. Hunter, Ed. Librairie Droz, 1936.
- [5] F. Zuccaro, *L'Idée de' pittori, scultori ed architetti*. Disserolio, 1607.
- [6] G. Bazin, *The Baroque: Principles, Styles, Modes, Themes*. Thames and Hudson, 1968.
- [7] F. Bacon, *New Atlantis*, G. B. Wegemer, Ed. CMTS Publishers at the University of Dallas, 2020.
- [8] R. G. Collingwood, *The Principles of Art*, 2nd ed. Oxford University Press, 1958, vol. 11.

⁸ Once the fundamental features of sound can be encoded in expressions interpretable by a machine, then "the engine might compose elaborate and scientific pieces of music of any degree of complexity or extent."

- [9] M. Wreen, “Is Madam? Nay, It Seems!” in *The Forger’s Art: Forgery and the Philosophy of Art*, D. Dutton, Ed. University of California Press Berkeley, 1983, pp. 188–224.
- [10] W. Benjamin *et al.*, *Das Kunstwerk Im Zeitalter Seiner Technischen Reproduzierbarkeit*. Suhrkamp Frankfurt, 1936, vol. 2.
- [11] H. Leichtentritt, “Mechanical Music in Olden Times,” *The Musical Quarterly*, vol. 20, no. 1, pp. 15–26, 1934.
- [12] R. Moore, “Digital reproducibility and the culture industry: Popular music and the adorno-benjamin debate,” *Fast Capitalism*, vol. 9, no. 1, pp. 75–88, 2012.
- [13] C. C. Losada, “Between Modernism and Postmodernism: Strands of Continuity in Collage Compositions by Rochberg, Berio, and Zimmermann,” *Music Theory Spectrum*, vol. 31, no. 1, pp. 57–100, 2009.
- [14] E. Bowles, “Musicke’s Handmaiden: Or Technology in the Service of the Arts,” in *The Computer and Music*. Cornell Ithaca, NY, 1970, pp. 3–23.
- [15] T. Riley, “Composing for the Machine,” *European Romantic Review*, vol. 20, no. 3, pp. 367–379, 2009.
- [16] J. Pressing, “Psychological Constraints on Improvisational Expertise and Communication,” *In the Course of Performance: Studies in the World of Musical Improvisation*, pp. 47–67, 1998.
- [17] A. M. B. Berger, *Medieval Music and the Art of Memory*. Univ of California Press, 2005.
- [18] T. Champion, *A New Way of Making Foure Parts in Counterpoint*. John Playford, 1671.
- [19] R. Hannas, “Cerone, Philosopher and Teacher,” *The Musical Quarterly*, vol. 21, no. 4, pp. 408–422, 1935.
- [20] P. Schubert, “Counterpoint Pedagogy in the Renaissance,” in *The Cambridge History of Western Music Theory*. Cambridge University Press, 2002, pp. 503–533.
- [21] J. Z. McKay, “Musical curiosities in athanasius kircher’s antiquarian visions,” *Music in Art*, vol. 40, no. 1-2, pp. 157–172, 2015.
- [22] M. Chierotti, “Comporre senza conoscere la musica: Athanasius Kircher e la Musurgia Mirifica. Un singolare esempio di scienza musicale nell’età barocca,” *La Nuova Rivista Musicale Italiana*, vol. 28, pp. 382–410, 1994.
- [23] P. Canguilhem and A. Stalarow, “Singing Upon the Book According to Vicente Lusitano,” *Early Music History*, vol. 30, pp. 55–103, 2011.
- [24] J. Grimshaw, “Morley’s rule for first-species canon,” *Early music*, vol. 34, no. 4, pp. 661–668, 2006.
- [25] M. A. Roig-Francolí, “Modal Paradigms in Mid-Sixteenth-Century Spanish Instrumental Composition: Theory and Practice in Antonio de Cabezón and Tomás de Santa María,” *Journal of Music Theory*, pp. 249–291, 1994.
- [26] R. C. Wegman, J. Menke, and P. Schubert, *Improvising Early Music: The History of Musical Improvisation from the Late Middle Ages to the Early Baroque*. Leuven University Press, 2014.
- [27] E. E. Helm, “Six Random Measures of C. P. E. Bach,” *Journal of Music Theory*, vol. 10, no. 1, pp. 139–151, 1966.
- [28] S. I. Taneiev, *Convertible Counterpoint in the Strict Style*. Bruce Humphries, 1962.
- [29] P. Pesic, “Euler’s musical mathematics,” *The mathematical intelligencer*, vol. 35, no. 2, pp. 35–43, 2013.
- [30] S. A. Hedges, “Dice Music in the Eighteenth Century,” *Music & Letters*, vol. 59, no. 2, pp. 180–187, 1978.
- [31] K. Schiltz, “« casus ubique valet? » josquin, cerone et les dés dans la musique de la renaissance.” Centre d’Études Supérieures de la Renaissance, 2007-11-26, pp. 1–14.
- [32] C. Ariza, “Two Pioneering Projects from the Early History of Computer-Aided Algorithmic Composition,” *Computer Music Journal*, vol. 35, no. 3, pp. 40–56, 2011.
- [33] L. A. Hiller, “Music Composed with Computers — A Historical Survey,” in *The Computer and Music*. Cornell University Press, 1970, pp. 42–96.
- [34] R. Bortolozzo, “Il gioco pitagorico musicale di Antonio Calegari,” phdthesis, Università degli studi di Venezia, 1997.
- [35] A. Mangeruva, *Nuovo Metodo per Comporre Migliaja di Walser*. Roberti, 1839.
- [36] P. Forshaw, “Oratorium—Auditorium—Laboratorium: Early modern improvisations on Cabala, music, and alchemy,” *Aries*, vol. 10, no. 2, pp. 169–195, 2010.
- [37] J. Calvo-Zaragoza, J. H. Jr, and A. Pacha, “Understanding optical music recognition,” vol. 53, no. 4, pp. 1–35, publisher: ACM New York, NY, USA.
- [38] Ovid, *Ovid’s Methaphores Books 6-10*, W. S. Anderson, Ed. University of Oklahoma Press, 1997.
- [39] M. P. Battin, “Exact replication in the visual arts,” *The Journal of Aesthetics and Art Criticism*, vol. 38, no. 2, pp. 153–158, 1979.
- [40] S. Colton, G. A. Wiggins *et al.*, “Computational creativity: The final frontier?” in *Ecai*, vol. 12. Montpellier, 2012, pp. 21–26.

- [41] H. B. Lincoln, "Preface," in *The Computer and Music*, H. B. Lincoln, Ed. Cornell University Press, 1970, pp. xi–xii.
- [42] R. M. Grant, "Leonhard Euler's unfinished theory of rhythm," *Journal of Music Theory*, vol. 57, no. 2, pp. 245–286, 2013.
- [43] T. Baca, J. W. Oberholtzer, J. Trevino, and V. Adán, "Abjad: An open-source software system for formalized score control," in *Proceedings of the First International Conference on Technologies for Music Notation and Representation*, 2015.

END-TO-END AUTOMATIC SINGING SKILL EVALUATION USING CROSS-ATTENTION AND DATA AUGMENTATION FOR SOLO SINGING AND SINGING WITH ACCOMPANIMENT

Yaolong Ju Chun Yat Wu Betty Cortiñas Lorenzo
Jing Yang Jiajun Deng Fan Fan Simon Lui
Huawei Technologies Co., Ltd., China

{yaolongju, wu.chun.yat, cortinas.lorenzo.betty
yangjing201, deng.jiajun, fanfan1, luisiuhang}@huawei.com

ABSTRACT

Automatic singing skill evaluation (ASSE) systems are predominantly designed for solo singing, and the scenario of singing with accompaniment is largely unaddressed. In this paper, we propose an end-to-end ASSE system that effectively processes both solo singing and singing with accompaniment using data augmentation, where a comparative study is conducted on four different data augmentation approaches. Additionally, we incorporate bi-directional cross-attention (BiCA) for feature fusion which, compared to simple concatenation, can better exploit the inter-relationships between different features. Results on the 10KSinging dataset show that data augmentation and BiCA boost performance individually. When combined, they contribute to further improvements, with a Pearson correlation coefficient of 0.769 for solo singing and 0.709 for singing with accompaniment. This represents relative improvements of 36.8% and 26.2% compared to the baseline model score of 0.562, respectively.

1. INTRODUCTION

In recent years, the widespread use of digital media has changed the way users interact with music, giving rise to new applications like streaming services and online karaoke platforms [1, 2]. As numerous singing content is published daily by these applications, it becomes very expensive and practically unscalable to retrieve high-quality content manually. One such scenario is the discovery of vocal talent in the vast online platforms, where automatic singing skill evaluation (ASSE) systems can be used to examine and rate all the singing content, so that the top-tier can be distributed for more views, subscribers, and ultimately more profits.

Despite the potential commercial values, ASSE is a difficult task that encompasses both subjective preferences and multi-dimensional objective features (e.g., intonation accuracy, rhythm accuracy, range, and dynamics) that professional judges also consider when evaluating vocal performances [3]. Over the years, different ASSE systems have been proposed. Depending on whether a reference melody is taken as the ground truth, these ASSE systems can be classified as reference-dependent [4–9] or reference-independent approaches [10–18]. Recent research on ASSE has been mainly focused on reference-independent deep learning-based approaches, where CNN-based architectures are often used to extract useful patterns from input spectrograms [11, 14, 15, 17, 18]. Other features including pitch histograms [11, 14, 15, 17] and singer timbre embeddings [17, 18] are also used, and these features are usually fused via concatenation. Although this is a simple way of feature fusion, the more advanced techniques that could uncover deeper relationships between these features are still unexplored in ASSE.

Another limitation of the current ASSE research stems from the lack of open-source datasets and high-quality annotations. For example, among the three recent datasets: neither the Smule DAMP dataset [14] nor the YJ-16K dataset [18] is open-sourced, and although Lyra-SA [19] is available after filling out an application form¹, the authors claimed that singing skill annotations are still immature and therefore not sufficiently curated for research purposes yet. Other ASSE datasets including self-made recordings [9, 20] or collections from singing platforms [7, 12] are also non-public. The lack of publicly available datasets is one of the major impediments that significantly hinder the advancement of ASSE research.

Finally, most ASSE systems require solo singing as input, leaving the scenario of singing with accompaniment largely unexplored [7–9, 11, 12, 14, 15]. On the other hand, [17] proposed an ASSE system that can process singing with accompaniment, but it is achieved by employing a singing voice separation tool [21] as a pre-processing step to remove the accompaniment, which not only results in a more complicated and computationally expensive system, also the model input is still essentially solo



© Y. Ju, C. Y. Wu, B. C. Lorenzo, J. Yang, J. Deng, F. Fan, and S. Lui. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Y. Ju, C. Y. Wu, B. C. Lorenzo, J. Yang, J. Deng, F. Fan, and S. Lui, “End-to-end automatic singing skill evaluation using cross-attention and data augmentation for solo singing and singing with accompaniment”, in *Proc. of the 25th Int. Society for Music Information Retrieval Conf.*, San Francisco, United States, 2024.

¹ Available at: <https://lyracobar.y.qq.com/singvoicedataset.html>

singing. In this paper, we propose a new ASSE system capable of processing both solo singing and singing with accompaniment in an end-to-end manner, thereby eliminating the need for a singing voice separation tool. This is achieved through data augmentation during training, where we present the same singing clip in three distinct versions: solo singing, singing with its original accompaniment, and singing remixed with a different accompaniment. For the remixed version, we explore and compare four different approaches, which will be detailed in Section 2.2.2. Furthermore, we explore feature fusion techniques beyond simple concatenation, since such methods can better integrate and amalgamate diverse data sources with greater efficacy [22, 23]. In particular, we adopt a Bi-directional Cross-Attention (BiCA) mechanism during feature fusion, given its effectiveness in capturing the reciprocal knowledge exchange between the source and target features in both directions [24]. The contributions of this paper are:

- We propose an ASSE system that processes both solo singing and singing with accompaniment. The system functions in an efficient end-to-end manner, thereby eliminating the need for a singing voice separation tool required by the baseline model [17].
- We adopt a BiCA mechanism during feature fusion, which better exploits the inter-relationships between different features and facilitates their reciprocal knowledge exchange, compared to simple feature concatenation (see Section 2.2.1).
- We explore data augmentation for ASSE and compare four different approaches: the existing Shuffle-And-Remix [25], the proposed Same-Song Remix, the proposed Key-Match Remix, and All Remix that combines the data augmented from the above three methods (see Section 2.2.2).
- Results show that BiCA and data augmentation boost performance individually (see Section 3.4). The combination of both results in further improvements, with a Pearson correlation coefficient of 0.769 for solo singing and 0.709 for singing with accompaniment on the 10KSinging dataset. This represents relative improvements of 36.8% and 26.2% compared to the baseline score of 0.562 [17], respectively.

2. METHODOLOGY

2.1 The Baseline Model

In the ASSE literature, singing skills can be presented as a ranking [11], a category [8, 9, 12, 18] (e.g., awesome, mediocre, or inferior), or a numerical score [14, 15, 17, 20] (e.g., 60 out of 100). We consider numerical scores for singing skills since they can be mapped into discrete categories or sorted as a ranking, which can be used in different scenarios. Within this range, the existing literature on ASSE is quite limited, and we consider [17] the baseline

model for our study, due to its superior performances to the recent ASSE system [14].

The pipeline of the baseline model is shown in Fig. 1(a): it begins by extracting solo singing from the input using an existing singing voice separation tool [21], then the Constant-Q Transform (CQT) is computed and processed by a Convolutional Recurrent Neural Network (CRNN) with an attention mechanism. Following this, the 200-dimensional output from the CRNN and attention is fused with the 120-dimensional pitch histogram² and the 512-dimensional X-vector³ using concatenation. Finally, the combined features are subsequently fed into a streamlined pair of dense layers to output the predicted singing rating.⁴

Compared to [14], three improvements were made in [17]: (1) the attention mechanism was added to the CRNN structure to further explore the useful, long-term relationships in the feature space; (2) X-vector [26] was added as additional features to depict the singing voice timbre, representing the control, resonance, and power that can be essential in singing skill evaluations; (3) the network structure was also finetuned to accommodate the first two improvements, where an extra dense layer was added to optimize the performance. Furthermore, they presented the 10KSinging dataset, which includes the singing skill ratings for 9,756 songs from 93 Chinese male singers and 97 Chinese female singers, and it was further divided into training, validation, and testing sets with 8,000, 756, and 1,000 songs, respectively. Each song from 10KSinging has two versions: the original singing with accompaniment version and the solo singing version, where the accompaniment of the latter was removed using a singing voice separation tool [21]. They used both versions to train their proposed ASSE model and found the solo singing version achieved better performances. Therefore, they considered singing voice separation an integral part of the pipeline, extracting solo singing as the input to their ASSE model shown in Fig. 1(a). As a result, a 62.4% relative improvement was achieved on Pearson correlation coefficient compared to [14] (0.562 VS. 0.346) on the 10KSinging dataset, serving as a solid baseline in this paper.

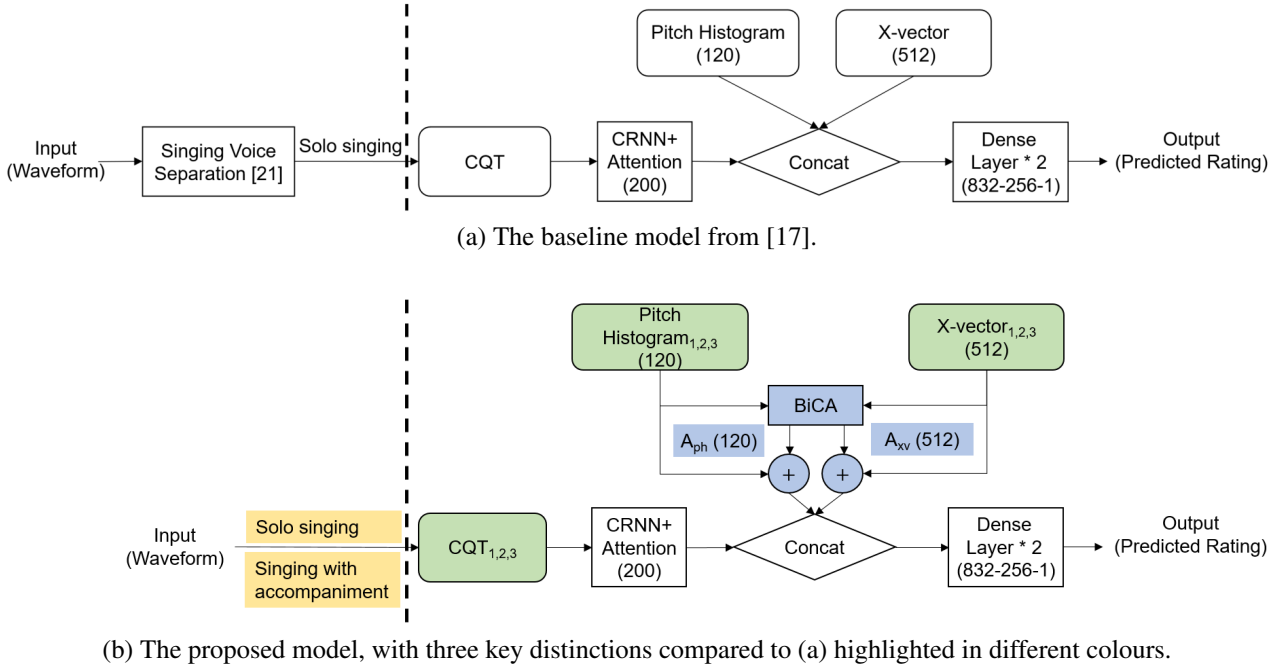
2.2 The proposed Model

Our proposed ASSE model is shown in Fig. 1(b), which highlights the three improvements compared to the baseline model Fig. 1(a) in different colours. The yellow part indicates that the proposed model can process both solo singing and singing with accompaniment, while the baseline model requires singing voice separation to extract solo singing from the input; the blue part represents the use of bi-directional cross-attention between the pitch histogram

² Originally proposed in [14], pitch histogram is a global representation of pitch distribution for music, where all octave-equivalent pitches are folded, resulting a range of 12 pitch classes. The distance between two adjacent pitch classes is represented with 10 bins.

³ According to [26], X-vector distinguishes different kinds of voice timbre. It has been applied to areas including speaker/emotion recognition [27], and singer identification [28].

⁴ The reader can refer to [14] for more specifics regarding the CRNN structure and the pitch histogram generation, and [17] for details on the attention mechanism integrated into the CRNN and X-vector extraction.



(b) The proposed model, with three key distinctions compared to (a) highlighted in different colours.

Figure 1. Illustration of the baseline model (a) and the proposed model (b). The diagram on the right of the dashed line indicates the architecture of the two ASSE systems, where (a) requires singing voice separation as pre-processing [21] to extract solo singing while (b) can process both solo singing and singing with accompaniment (yellow) in a more efficient, end-to-end manner. The proposed model features Bi-directional Cross-Attention (BiCA, blue) and data augmentation (green) using the three distinct training sets of 10KSinging, namely: subindex 1 as singing with accompaniment, subindex 2 as solo singing, and subindex 3 as singing and accompaniment remix discussed in Section 2.2.2. The number in parentheses represents the number of dimensions, while A_{ph} and A_{xv} denote the attention output for the pitch histogram and X-vector, respectively. Both A_{ph} and A_{xv} enter a sum operation with their respective input feature via residual connections.

and X-vector features (see Section 2.2.1); the green part represents data augmentation, where three distinct sets of 10KSinging: singing with accompaniment, solo singing, and singing and accompaniment remix are used during the training process (see Section 2.2.2).

2.2.1 Bi-directional Cross-Attention

As discussed above, the baseline model [17] includes a self-attention mechanism in CRNN to capture the long-term relationships from the input CQT spectral representation. This is based on the scaled dot-product attention layer proposed by Vaswani et al [29]:

$$\begin{aligned} \text{Attn}(Q, K, V) &= \text{softmax}\left(\frac{Q \times K^T}{\sqrt{D}}\right) \times V \\ &= \text{softmax}(S) \times V, \end{aligned}$$

where Q , K , V , S , and D denote the query, key, value, similarity matrix, and dimension of the attention layer, respectively.

In addition to the self-attention mechanism adopted by the baseline model, we further improve our approach by applying cross-attention to the remaining two features: pitch histogram and X-vector, since there can be correlations between singers' pitch accuracy and timbre quality that are beneficial for ASSE. In the cross-attention mechanism, the query Q_t is derived from the target t , with the key

K_s and the value V_s derived from the source s . The attention output $\text{Attn}_t(Q_t, K_s, V_s)$ is then added to the target t via a residual connection, leaving the source s unmodified. This means if we aim to apply cross-attention to both pitch histogram and X-vector features, we need to do it twice: one using pitch histogram as target (t), X-vector as source (s) and vice versa for the other one. To reduce the excessive computational demands in this case, we adopted Bi-directional Cross-Attention (BiCA) [24] that contains a reciprocal attention mechanism, where a shared query-key (QK) matrix [30] is applied to update both the target t and the source s in parallel. Concretely, the similarity matrices of S_t and S_s in BiCA can be calculated as:

$$S_t = \frac{(QK)_t \times (QK)_s^T}{\sqrt{D}} = S_s^T,$$

where $(QK)_t$ and $(QK)_s$ are the shared query-key matrices projected from t and s , respectively. As a result, the attention features of t and s can be respectively obtained by multiplying the corresponding similarity matrix with the value matrix projected from both t and s :

$$\text{Attn}_t = \text{softmax}(S_t) \times V_s; \quad \text{Attn}_s = \text{softmax}(S_s) \times V_t$$

Finally, we perform a residual connection in both t and s to add the corresponding attention features Attn_t and Attn_s . Overall, we enhance the learning of inter-relationships between pitch histogram and X-vector by

implementing the cross-attention mechanism, specifically BiCA⁵ to our approach illustrated in Fig. 1(b). This way, our proposed model is capable of maintaining the effectiveness of cross-attention while being computationally efficient [24].

2.2.2 Data Augmentation for ASSE

As discussed in Section 1, the lack of data can hinder the research and development of ASSE models, and we aim to mitigate this problem by adopting data augmentation. For this purpose, we use the 10KSinging dataset from [17], which contains 9,756 songs in two versions: singing with accompaniment and solo singing. It is further divided into training, validation, and testing sets with 8,000, 756, and 1,000 songs, respectively. We combine the two versions (singing with accompaniment and solo singing) of the training sets and develop a third one called “singing and accompaniment remix”, with an additional 8,000 songs generated by remixing the solo singing with a different accompaniment to create more data. For this purpose, we compare three different remixing approaches:

- **Shuffle-And-Remix** [25]: this existing approach remixes each solo singing with a randomly selected accompaniment from another song. Note that with this approach, the singing and accompaniment may not be in the same musical key, and combining the two will introduce differences in musical key irrelevant to ASSE and may interfere with the training process. Therefore, we propose two new remixing techniques that ensure the same key between singing and accompaniment as follows.
- **Same-Song Remix**: instead of using a different song, we can shift the accompaniment track of the same song by a random duration between 5 to 15 seconds ahead or behind the singing track and remix both. This creates a unique alignment where the vocals and music are out of their original synchronization but still ensures both are in the same musical key.
- **Key-Match Remix**: we use the Madmom key detection algorithm [31] to iterate all the accompaniments and eliminate the ones that are in a different key than the solo singing. Among the remaining candidates, we randomly pick one accompaniment, and remix it with the solo singing.

As a result, we have an augmented training set of 24,000 songs in total, where singing with accompaniment, solo singing, and singing and accompaniment remix all contribute 8,000 songs, indicated respectively as subindex 1, 2, 3 in Fig. 1(b). Furthermore, we can extend the set of subindex 3 by combining the augmented data from all three remixing approaches above (8000×3 songs) and propose a fourth approach: **All Remix**, with 40,000 songs in total.

⁵ We use an open-source implementation of BiCA available at: <https://github.com/lucidrains/bidirectional-cross-attention>.

3. EXPERIMENTS

As indicated in [17], each song of the 10KSinging dataset is associated with an overall, normalized rating between 0 and 1, and the goal of our ASSE model is to predict a regressed value close to the ground truth rating. Although the work presented in this paper is not open-source for proprietary restrictions, most of the essential components are open-source as follows: the code base and the fundamental structure of CRNN, including the pitch histogram calculation can be found at Github⁶; the annotations for 10KSinging, the attention mechanism appended after CRNN, and the X-vector calculation can be found at [17], and we will respectively explain our experimental settings and relevant implementation details in Section 3.1 and Section 3.2 for the ease of reproducing our work.

3.1 Experimental Settings

We first investigate the effect of data augmentation in five settings: no data augmentation, Shuffle-And-Remix (SAR), Same-Song Remix (SSR), Key-Match Remix (KMR), and All Remix (ALL) proposed in Section 2.2.2. In each data augmentation setting, we can either use the baseline architecture (Fig. 1(a)) or adopt BiCA (Fig. 1(b)), resulting in a total of 10 experiments. In each experiment, we present the performance on the 1000-song test set from the two versions of 10KSinging: singing with accompaniment (“w/ acc”) and solo singing (“w/o acc”). As shown in Table 1, the first two experiments involve no data augmentation and each has two distinct ASSE models that are trained using the two versions of 10KSinging (“w/ acc” and “w/o acc”), same as [17].

For the remaining eight experiments involving data augmentation, each uses an augmented training set of 10KSinging, which contains songs from the following three sets: singing with accompaniment, solo singing, and singing and accompaniment remix introduced in Section 2.2.2. Unlike the first two experiments, each of the eight experiments has only one ASSE model, which is evaluated in both “w/ acc” and “w/o acc” test sets. Altogether, 12 models are trained in total to explore the effects of data augmentation and BiCA.

3.2 Implementation Details

We adopt the same parameters for generating CQT and pitch histograms as described in [17], namely 96-bin CQT and 120-bin pitch histogram. For training, we use Mean Squared Error (MSE) as the loss function, where the epoch with the lowest MSE on the validation set is chosen as the best-performing model, both in the “w/ acc” and “w/o acc” settings. We use the Adam optimizer and a learning rate of 0.0001. The number of epochs is set to 250 with a batch size of 4. All other parameters remained consistent with those outlined in [17], except for a few adjustments, which are detailed below.

⁶ Implementation can be found at: <https://github.com/AME430/Towards-Training-Explainable-Singing-Quality-Assessment-Network-with-Augmented-Data>.

Data Aug	BiCA	MSE(↓)		MAE(↓)		Bad P%(↓)		Pearson(↑)	
		w/o acc	w/ acc	w/o acc	w/ acc	w/o acc	w/ acc	w/o acc	w/ acc
No (8,000)	No [17]	0.0042	0.0046	0.0495	0.0524	10.1%	11.8%	0.562	0.497
No (8,000)	Yes	0.0038	0.0043	0.0459	0.0499	8.7%	10.5%	0.623	0.539
SAR (24,000)	No	0.0041	0.0044	0.0495	0.0519	9.3%	9.4%	0.561	0.522
SAR (24,000)	Yes	0.0031	0.0033	0.0386	0.0415	6.5%	7.3%	0.697	0.670
SSR (24,000)	No	0.0041	0.0044	0.0497	0.0515	9.6%	10.2%	0.555	0.514
SSR (24,000)	Yes	0.0029	0.0033	0.0385	0.0416	5.8%	7.0%	0.714	0.673
KMR (24,000)	No	0.0041	0.0044	0.0496	0.0521	8.9%	9.8%	0.562	0.517
KMR (24,000)	Yes	0.0028	0.0031	0.0375	0.0413	6.2%	6.9%	0.730	0.687
ALL (40,000)	No	0.0039	0.0040	0.0471	0.0478	9.4%	10.1%	0.593	0.576
ALL (40,000)	Yes	0.0025	0.0030	0.0351	0.0387	4.7%	6.1%	0.769	0.709

Table 1. The ASSE results of Mean Squared Error (MSE), Mean Absolute Error (MAE), Bad Case Proportion (Bad P%), and Pearson correlation coefficient (Pearson) on the singing with accompaniment test set (w/ acc, 1,000 songs) and the solo singing test set (w/o acc, 1,000 songs) from the 10KSinging dataset [17]. SAR, SSR, KMR, and ALL refer to the four different data augmentation methods introduced in Section 2.2.2: Shuffle-and-Remix, Same-Song Remix, Key-Match Remix, and All Remix, respectively, where the number in parenthesis indicates the number of songs used as training data. For experimental purposes, no data augmentation, SAR, SSR, KMR, and ALL is respectively applied to the model architecture without and with Bi-Directional Cross-Attention (BiCA, illustrated in Fig. 1(b)) to demonstrate the individual and reciprocal effects of data augmentation and BiCA. The downward and upward arrows on the evaluation metrics respectively represent the desirable lower or higher values for better performances. The best results are highlighted in bold, which concentrate on the ASSE model employing both All Remix data augmentation and BiCA (ALL-Yes), is therefore our proposed method in this paper.

We use the sigmoid activation function following the final dense layer to constrain the output range between 0 to 1. Also, the Exponential Linear Unit (ELU) activation function is introduced within the dense layer. These adjustments can facilitate the model’s ability to learn a more accurate distribution of the output score.

3.3 Evaluation Metrics

Although correlation coefficients are often used as the evaluation metric in ASSE [5, 13–15, 17, 32], we aim to incorporate additional metrics to demonstrate the performances of ASSE models more comprehensively. Overall, four evaluation metrics are considered:

- Mean squared error (MSE) (↓): same as the loss function introduced in Section 3.2.
- Mean absolute error (MAE) (↓): it shows how much the predicted rating deviates from the ground truth in the linear scale.
- Bad case proportion (↓): same as [17], the predicted rating will be considered a bad case if its MAE is no less than 0.1.
- Pearson correlation coefficient (↑): it demonstrates the degree of correlation between the predicted rating and the ground truth, within the range of $[-1, 1]$.

3.4 Results and Discussions

The results are shown in Table 1, where we use acronyms to represent each experiment. For example, No-No indi-

cates the experiment without data augmentation nor BiCA, and KMR-Yes indicates the experiment using both KMR augmentation and BiCA, etc.

3.4.1 Results on BiCA

We first investigate the effects of BiCA by comparing the models with and without BiCA under five different data augmentation settings (No-No VS. No-Yes; SAR-No VS. SAR-Yes; SSR-No VS. SSR-Yes; KMR-No VS. KMR-Yes; ALL-No VS. ALL-Yes), finding that using BiCA results in consistent performance improvements in all cases. This demonstrates that the employment of BiCA effectively helps the ASSE models capture useful inter-relationships between pitch histogram and X-vector and facilitate their reciprocal knowledge exchange, leading to better results. This is reasonable since there can be correlations between singers’ pitch accuracy and timbre quality that are beneficial for ASSE. For example, singers with excellent singing skills tend to have great pitch accuracy (indicated by pitch histogram) and timbre quality (indicated by X-vector), and vice versa for mediocre or inferior singers.

3.4.2 Results on Data Augmentation

We then compare the four data augmentation approaches: SAR, SSR, KMR, and ALL to no data augmentation. When using the baseline architecture (Fig. 1(a) without BiCA), results show overall marginal improvements in almost all cases (SAR-No VS. No-No; SSR-No VS. No-No; KMR-No VS. No-No; ALL-No VS. No-No). When

BiCA is applied, the improvement is much more apparent (SAR-Yes VS. No-Yes; SSR-Yes VS. No-Yes; KMR-Yes VS. No-Yes; ALL-Yes VS. No-Yes). These results demonstrate the effectiveness of data augmentation and its reciprocal advantage with BiCA. As discussed in Section 3.4.1, it seems that data augmentation provides more samples for BiCA to further exploit correlations between pitch histogram and X-vector, which can be beneficial for evaluating singing skills in ASSE.

Of particular interest, we notice that KMR achieves superior results among all three data augmentation approaches (KMR-Yes VS. SSR-Yes VS. SAR-Yes). This could be that KMR combines the advantages of SAR and SSR, where the former mixes the singing with a different accompaniment and the latter ensures the same key between singing and accompaniment, leading to better performances. Despite their differences, we can combine the augmented data from SAR, SSR, and KMR as All Remix (see Section 2.2.2) for even more training data, resulting in the best results overall (SAR-Yes VS. SSR-Yes VS. KMR-Yes VS. ALL-Yes).

3.4.3 Results on Solo Singing and Singing with Accompaniment

Additionally, we compare the performances presented in solo singing (w/o acc) and singing with accompaniment (w/ acc) scenarios. Results show that the ASSE models consistently perform better in solo singing, which is understandable considering singing with accompaniment contains irrelevant accompaniment information that can interfere with the training of ASSE models. Although we can follow [17] to add a singing voice separation step (see Fig. 1(a)) to remove accompaniment for better performances, we choose to keep the end-to-end nature of our ASSE system (see Fig. 1(b)) and consider the performance gap between solo singing and singing with accompaniment for our proposed ASSE model (ALL-Yes) non-essential, since both are performing better than the baseline No-No in the solo singing (w/o acc) condition.

3.4.4 Overall Results

Finally, once we combine data augmentation with BiCA, our proposed ALL-Yes model yields notably better results than the baseline [17] (No-No) across all metrics: reaching relative improvements of 40.5% on MSE (0.0025 VS. 0.0042, likewise for the following ones), 29.1% on MAE, 53.5% on Bad P %, and 36.8% on Pearson in solo singing (w/o acc); 34.8% on MSE (0.0030 VS. 0.0046, likewise for the following ones), 26.1% on MAE, 48.3% on Bad P %, and 42.7% on Pearson in singing with accompaniment (w/ acc) scenario.

As discussed in Section 3.1, there are two models under No-No, one trained for w/o acc and the other trained for w/ acc conditions. [17] then proposed the former in the paper due to its superior performance. However, it can only process solo singing data and requires a singing voice separation tool to remove accompaniment from the input. In comparison, our proposed ALL-Yes model can process

both solo training and singing with accompaniment inputs, and this is what we refer to as an end-to-end ASSE model, which does not require singing voice separation and also yields notably better performances, with Pearson correlation coefficients of 0.769 in w/o acc and 0.709 in w/ acc, compared to the baseline model of 0.562.

4. CONCLUSIONS

In this paper, we introduce a new ASSE system using data augmentation and compare four specific augmentation approaches: the existing Shuffle-And-Remix [25], the novel Same-Song Remix, Key-Match Remix, and All Remix we propose. Results show that our All Remix approach achieves the best performances, and our system can process both solo singing and singing with accompaniment in an end-to-end manner, thereby eliminating the need for a singing voice separation tool required by the baseline model [17]. We also introduce a Bi-directional Cross-Attention mechanism (BiCA) as a feature fusion method to ASSE for the first time, which discovers useful inter-relationships between pitch histogram and X-vector and results in consistent performance improvements in our experiments.

With the combination of BiCA and All Remix data augmentation approach, we not only achieve notable improvements in ASSE performances compared to the baseline [17], we also develop a versatile model capable of processing both solo and instrumentally accompanied vocal performances. To the best of our knowledge, such encompassing ASSE models have not been proposed in existing literature before.

5. FUTURE WORK

Looking ahead, we will continue this research by incorporating future open-source ASSE datasets proposed in the literature. Indeed, we could only make use of the 10K Singing dataset due to the lack of open-source datasets in this domain. Our future work also includes exploring alternative features that could potentially improve the performances of our ASSE models. For instance, we will consider large-scale music models that employ self-supervised learning, since features extracted by those models such as Jukebox [33] and MERT [34] have recently been proven effective and even established new SOTA performances in various music-related tasks. Therefore, these features will be incorporated into our model to exert their potential. Finally, since data augmentation combining solo singing with different versions of accompaniment results in consistent performance improvements, we will explore more data augmentation methods for solo singing by, for example, adding noise, adjusting gain, and applying high/low-pass filters that have been employed in other MIR-related tasks [35] for better performances.

6. REFERENCES

- [1] S. Jones, “Music and the internet,” *The handbook of internet studies*, pp. 440–451, 2011.
- [2] C. Shin and Y. Lim, “Design and implementation of impromptu mobile social karaoke for digital cultural spaces in the new normal era,” *Applied Sciences*, vol. 13, no. 22, 2023.
- [3] A. M. Studiorum, “Subjective evaluation of common singing skills using the rank ordering method,” in *Proceedings of the Ninth International Conference on Music Perception and Cognition*, 2006, pp. 1507–1512.
- [4] W.-H. Tsai and H.-C. Lee, “An automated singing evaluation method for karaoke systems,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011, pp. 2428–2431.
- [5] C. Gupta, H. Li, and Y. Wang, “A technical framework for automatic perceptual evaluation of singing quality,” *APSIPA Transactions on Signal and Information Processing*, vol. 7, no. e10, 2018.
- [6] —, “Perceptual evaluation of singing quality,” in *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2017, pp. 577–586.
- [7] H. Zhang, Y. Jiang, T. Jiang, and P. Hu, “Learn by referencing: Towards deep metric learning for singing assessment,” in *Proceedings of the 22nd International Society for Music Information Retrieval Conference (ISMIR)*, 2021, pp. 825–832.
- [8] J. Böhm, F. Eyben, M. Schmitt, H. Kosch, and B. Schuller, “Seeking the superstar: Automatic assessment of perceived singing quality,” in *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, 2017, pp. 1560–1569.
- [9] Bari, Bozkurt, O. Baysal, and D. Yüret, “A dataset and baseline system for singing voice assessment,” in *Proceedings of the International Symposium on Computer Music Multidisciplinary Research (CMMR)*, 2017, pp. 25–28.
- [10] T. Nakano, M. Goto, and Y. Hiraga, “An automatic singing skill evaluation method for unknown melodies using pitch interval accuracy and vibrato features,” in *Proceedings of the 9th International Conference on Spoken Language Processing (INTERSPEECH)*, vol. 4, 2006, pp. 1706–1709.
- [11] C. Gupta, H. Li, and Y. Wang, “Automatic leaderboard: Evaluation of singing quality without a standard reference,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 13–26, 2020.
- [12] N. Zhang, T. Jiang, F. Deng, and Y. Li, “Automatic singing evaluation without reference melody using bi-dense neural network,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 466–470.
- [13] C. Gupta, L. Huang, and H. Li, “Automatic rank-ordering of singing vocals with twin-neural network,” in *Proceedings of the 21st International Society for Music Information Retrieval Conference (ISMIR)*, 2020, pp. 416–423.
- [14] L. Huang, C. Gupta, and H. Li, “Spectral features and pitch histogram for automatic singing quality evaluation with crnn,” in *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2020, pp. 492–499.
- [15] J. Li, C. Gupta, and H. Li, “Training explainable singing quality assessment network with augmented data,” in *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2021, pp. 904–911.
- [16] C. Gupta, J. Li, and H. Li, “Towards reference-independent rhythm assessment of solo singing,” in *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2021, pp. 912–919.
- [17] Y. Ju, C. Xu, Y. Guo, J. Li, and S. Lui, “Improving automatic singing skill evaluation with timbral features, attention, and singing voice separation,” in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, 2023, pp. 612–617.
- [18] X. Sun, Y. Gao, H. Lin, and H. Liu, “Tg-critic: A timbre-guided model for reference-independent singing evaluation,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.
- [19] T. M. L. L. team from Tencent Music Entertainment Group, “Lyra-SA Dataset,” <https://lyracobar.y.qq.com/singvoicedataset.html>, 2023, [Online; accessed 21-March-2023].
- [20] W.-H. Tsai and H.-C. Lee, “Automatic evaluation of karaoke singing based on pitch, volume, and rhythm features,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 4, pp. 1233–1243, 2012.
- [21] C. Li, Y. Li, X. Du, Y. Ju, S. Hu, and Z. Wu, “VocEmb4SVS: Improving singing voice separation with vocal embeddings,” in *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2022, pp. 234–239.
- [22] Y. Yang, J. Song, Z. Huang, Z. Ma, N. Sebe, and A. G. Hauptmann, “Multi-feature fusion via hierarchical regression for multimedia analysis,” *IEEE Transactions on Multimedia*, vol. 15, no. 3, pp. 572–581, 2013.

- [23] J. Chen, Y. He, and J. Wang, “Multi-feature fusion based fast video flame detection,” *Building and Environment*, vol. 45, no. 5, pp. 1113–1122, 2010.
- [24] M. Hiller, K. A. Ehinger, and T. Drummond, “Perceiving longer sequences with bi-directional cross-attention transformers,” *arXiv preprint arXiv:2402.12138*, 2024.
- [25] T.-H. Hsieh, K.-H. Cheng, Z.-C. Fan, Y.-C. Yang, and Y.-H. Yang, “Addressing the confounds of accompaniments in singer identification,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 1–5.
- [26] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, “X-vectors: Robust DNN embeddings for speaker recognition,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5329–5333.
- [27] R. Pappagari, T. Wang, J. Villalba, N. Chen, and N. Dehak, “X-vectors meet emotions: A study on dependencies between emotion and speaker recognition,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7169–7173.
- [28] X. Zhang, J. Wang, N. Cheng, and J. Xiao, “Singer identification for metaverse with timbral and middle-level perceptual features,” in *2022 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2022, pp. 1–7.
- [29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., 2017, pp. 6000–6010.
- [30] N. Kitaev, Łukasz Kaiser, and A. Levskaya, “Reformer: The efficient transformer,” in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020.
- [31] F. Korzeniowski and G. Widmer, “Genre-agnostic key classification with convolutional neural networks,” in *Proceedings of the 19th International Society for Music Information Retrieval Conference (ISMIR)*, 2018, pp. 264–270.
- [32] C. Gupta, H. Li, and Y. Wang, “Automatic evaluation of singing quality without a reference,” in *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2018, pp. 990–997.
- [33] R. Castellon, C. Donahue, and P. Liang, “Codified audio language modeling learns useful representations for music information retrieval,” in *Proceedings of the 22nd International Society for Music Information Retrieval Conference (ISMIR)*, 2021, pp. 88–96.
- [34] Y. Li, R. Yuan, G. Zhang, Y. Ma, X. Chen, H. Yin, C. Lin, A. Ragni, E. Benetos, N. Gyenge, R. Dannenberg, R. Liu, W. Chen, G. Xia, Y. Shi, W. Huang, Y. Guo, and J. Fu, “MERT: Acoustic music understanding model with large-scale self-supervised training,” *arXiv preprint arXiv:2306.00107*, 2023.
- [35] J.-C. Wang, Y.-N. Hung, and J. B. Smith, “To catch a chorus, verse, intro, or anything else: Analyzing a song with structural functions,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 416–420.

Papers – Session IV

CLUSTER AND SEPARATE: A GNN APPROACH TO VOICE AND STAFF PREDICTION FOR SCORE ENGRAVING

Francesco Foscarin^{*1,2} Emmanouil Karystinaios^{*1} Eita Nakamura³ Gerhard Widmer^{1,2}

¹ Johannes Kepler University, Linz, Austria

² LIT AI Lab, Linz Institute of Technology, Austria

³ Kyushu University, Japan

firstname.lastname@jku.at

ABSTRACT

This paper approaches the problem of separating the notes from a quantized symbolic music piece (e.g., a MIDI file) into multiple voices and staves. This is a fundamental part of the larger task of music score engraving (or score typesetting), which aims to produce readable musical scores for human performers. We focus on piano music and support homophonic voices, i.e., voices that can contain chords, and cross-staff voices, which are notably difficult tasks that have often been overlooked in previous research. We propose an end-to-end system based on graph neural networks that clusters notes that belong to the same chord and connects them with edges if they are part of a voice. Our results show clear and consistent improvements over a previous approach on two datasets of different styles. To aid the qualitative analysis of our results, we support the export in symbolic music formats and provide a direct visualization of our outputs graph over the musical score. All code and pre-trained models are available at https://github.com/CPJKU/piano_svsep.

1. INTRODUCTION

The musical score is an important tool for musicians due to its ability to convey musical information in a compact graphical form. Compared to other music representations that may be easier to define and process for machines, for example, MIDI files, the musical score is characterized by how efficiently trained musicians can read it.

An important factor that affects the readability of a musical score for instruments that can produce more than one note simultaneously, is the separation of notes into different *voices* (see Figure 1). This division may follow what a listener perceives as independent auditory streams [1], which can also be reflected in a clearer visual rendition of a musical score [2]. A similar point can be made for the division

into multiple staves (generally 2) for instruments with a large pitch range, such as piano, organ, harp, or marimba. We will consider in this paper piano music.

The term voice is frequently used to describe a sequence of musical notes that do not overlap, which we call a *monophonic voice*. However, this definition may be insufficient when considering polyphonic instruments. Voices could contain *chords*, which are groups of *synchronous notes* (i.e., notes with the same onset and offset) and are perceived as a single entity [3]. We name a voice that can contain chords a *homophonic voice*. Note that partially overlapping notes cannot be part of a homophonic voice.

Music encoded in MIDI (or similar) formats, even when containing quantized notes, time signature, or bar information, often does not contain voice and staff information. The same can be said for the output of music generation [4], transcription [5], or arranging [6] systems. Therefore, such music cannot be effectively converted into a musical score, to be efficiently read and played by human musicians.¹ The tasks of producing voice and staff information from unstructured symbolic music input are called *voice separation* (or voice segregation in some papers [3]) and *staff separation*, respectively.

Most of the existing approaches to voice separation have focused only on music with monophonic voices [7–12], which is not sufficient for our goal of engraving² piano music. The task of homophonic voice separation is much harder to solve: the presence of chords within voices makes the space of solutions grow much bigger; and the choice of the “true voice separation” can be ambiguous, with multiple valid alternatives among which experts may disagree.

The existing approaches to homophonic voice separation can be divided into two groups: the first [1, 3, 5, 13] use dynamic programming algorithms based on a set of heuristics, which makes for systems that are controllable and interpretable, but also hard to develop and tune. Such systems are often prone to fail on exceptions and corner cases that are present in musical pieces. The second group of approaches [14–17] applies deep learning models to predict a voice label for each note. Such an approach creates

* Equal contribution.



¹ Voice and staff separation are only two of the multiple elements, such as pitch spelling, rhythmic grouping, and tuplet creations, which need to be targeted by a score engraving system, but we will only focus on the former two in this paper.

² “score engraving” and “score typesetting” are used interchangeably.



Figure 1. Comparing different voice/staff assignments for two bars from C. Debussy’s *Estampes - Pagodes*. (top) original; voices can be inferred from the beam grouping and (horizontal lines connecting notes), rests, and stem sharing, and are colored for clarity. (bottom) hard-to-read rendition with voice and staff assigned according to heuristics we propose as a baseline.

two fundamental issues: i) the necessity of setting a maximum number of voice labels, and ii) a (highly) unbalanced ratio of occurrence of some voice labels. Moreover, all these approaches assume that a voice cannot move between the two staves, which is not true for complex piano pieces.

In this work, we propose a novel system for homophonic voice separation that can efficiently and effectively assign notes to voices and staves for polyphonic music engraving. Efficiency is ensured by a graph neural network (GNN) encoder, which can create input embeddings with a relatively small number of parameters. Effectiveness is targeted by approaching voice prediction not as a note labeling, but as an edge prediction problem [12], which solves the maximum voice number and the label imbalance problems presented above. Our system predicts staff and voice separately and does not make any assumption on the number of voices; therefore it can deal with cross-staff voices and complex corner cases. We avoid the problem of ground truth ambiguity since we focus specifically on voice separation for musical score engraving, therefore we can extract the (unique) ground truth directly from digitized musical scores.

We evaluate our system on two piano datasets of different difficulty levels, one containing popular, the other classical music. A comparison with a baseline and the approach of Shibata et al. [5] shows a consistent improvement in performance on both datasets. Finally, we develop a visualization tool to display the input and output of our system directly on the musical score, and discuss some predictions and comments on homophonic voice separation.

2. RELATED WORK

The most influential work for this paper is the monophonic voice separation system by Karystinaios et al. [12]. Similarly, we consider voice separation a edge prediction task and use a similar score-to-graph routine and the same GNN

encoder. Differently from that work, we consider homophonic voices and staves and, therefore, we extend the model formulation, the deep learning architecture, and the postprocessing routine to deal with this information.

Shibata et al. [5] developed a voice and staff separation technique applied after music transcription to quantized MIDI files. It works in two stages: first, an HMM separates the notes of the two hands (which will then be used as staff), and then a dynamic programming algorithm that maximizes the adherence to a set of heuristics is applied to separate voices in the two hands independently. We compare against this method since it is the most recent approach focusing specifically on homophonic voice separation.

There are some approaches based on neural networks [14–17], but they never perform this task in isolation, but rather in combination with other tasks such as symbolic music transcription, full scorification, and automatic arrangement. This means that they can only train on a much smaller dataset and a comparison would not be fair.

All the approaches mentioned before, except [12], perform voice separation as a label prediction task, which is problematic, as discussed in the introduction, due to the label imbalance and choice of the maximum number of voices. The former is particularly problematic for the neural network approaches.

3. METHODOLOGY

Our system inputs data in the form of a set of quantized notes (e.g., coming from a quantized MIDI or a digitized musical score), each characterized by pitch, onset, and offset. This information is modeled as a graph, which we call *input graph*, and then passed through a GNN model to predict an *output graph* containing information about voices, staves, and chord groupings. We remind the reader that in our ‘homophonic voice’ scenario, chords are groups of synchronous notes that belong to the same voice.

3.1 Input Graph

From the set of quantized notes representing a musical piece we create a directed heterogeneous graph [18] $G_{in} = (V, E_{in}, \mathcal{R}_{in})$ where each node $v \in V$ corresponds to one and only one note, and the edges $e \in E_{in}$ of type $r \in \mathcal{R}_{in}$ model temporal relations between notes [12]. \mathcal{R}_{in} includes 4 types of relations: onset, during, follow, and silence, corresponding, respectively, to two notes starting at the same time, a note starting while the other is sounding, a note starting when the other ends, and a note starting after a time when no note is sounding. We also create the inverse edges for during, follows, and silence relations. Each node corresponds to a vector of features: one of the 12 note pitch classes³ (C, C#, D, etc.), the octave in $[1, \dots, 7]$, the note duration, encoded as a float value $d \in [0, 1]$ computed as the ratio of the note and bar durations, passed through a tanh function to limit its value and boost resolution for shorter

³ We don’t consider tonal pitch classes [19] since they are not specified in MIDI files which we assume to be our input.

notes, as proposed in [12]. We don't consider grace notes in our system, and we remove them from the input notes.

3.2 Output Graph

The output graph $G_{out} = (V, E_{out}, \mathcal{R}_{out})$ has the same set V of nodes as the input graph, but a staff number in $\{0, 1\}$ is assigned to every node. There are two edge types in E_{out} : chord and voice, i.e. $\mathcal{R}_{out} = \{\text{"chord"}, \text{"voice"}\}$.

Voice edges [8, 12] are an alternative in the literature to the more straightforward approach of predicting a voice number for every note; the usage of voice edges has the advantage of enabling a system to work with a non-specified number of voices, and avoiding the label imbalance problem for high voice numbers. Voice edges are directed edges that connect consecutive notes (without considering rests) in the same voice. Formally, let $u_1, u_2 \in V$ be two notes in the same voice then $(u_1, \text{"voice"}, u_2) \in E_{out}$ if and only if $\text{offset}(u_1) \leq \text{onset}(u_2)$ and $\nexists u_3 \in V$ within the same voice such that $\text{offset}(u_1) \leq \text{onset}(u_3) < \text{onset}(u_2)$.

The previous definition also holds in our setting with homophonic voices. Let us extend the definition of *chord* (a set of synchronous notes) to include the limit case of a single note. Two chords are consecutive if any two notes, respectively, from the first and second chords are consecutive. In the case of two consecutive chords with m and n notes in the same voice, there will be $m * n$ voice edges.

Chord edges are undirected and connect all notes that belong to the same chord without self-loops, so for a n -note chord, there are $n(n - 1)$ edges. They serve to unambiguously identify which notes together form a single chord.

The same output graph can be created from an already properly engraved score. To obtain the graph we only need to draw the true voice edges between consecutive notes in the same voice within a bar and for chord edges we draw the chord ground truth between synchronous notes with the same voice number assignment. This graph can subsequently serve as the ground truth during training.

3.3 Problem Simplification

In this section, we apply some obvious musical constraints to reduce computation and memory usage in our pipeline, without impacting the results. Let us first focus on *chord edge* prediction. Given the simple constraint that all notes of a chord must start and end simultaneously, we can restrict the chord edge prediction process to only consider pairs of synchronous notes (same onset and offset values) as candidates. We do this by creating a set of *chord edge candidates* Λ_c which are calculated automatically and associated with our input graph. Only notes connected by such candidate edges will be considered in the chord prediction part of the model (see next section).

The same reasoning can be applied to the voice edges, by creating a set of *voice edge candidates* Λ_v such that $\forall u_1, u_2 \in V, (u_1, \text{"voice"}, u_2) \in \Lambda_v$ only when $\text{offset}(u_1) > \text{onset}(u_2)$. Another step can be taken towards reducing the number of candidates in the set Λ_v by incorporating some musical engraving considerations.

The separation of notes in multiple voices does not have to be consistent in the whole score, but only within each bar, to produce the intended visual representation. There are no graphical elements that show if two notes in different bars are or are not in the same voice⁴. Music engraving software does not force users to use consistent voices across bars. This can be often observed in digitized musical scores where music motives that belong to the same voice, are assigned different voices in different bars. Such observations have motivated projects such as the Symbolic Multitrack Contrapuntal Music Archive [20] that explicitly encode a "global" voice number.

Since cross-bar consistency is not necessary for our goal of engraving (and is often wrongly annotated in our data) we limit the *voice edge candidates* Λ_v to contain only pairs of notes that occur within the same bar. This design choice is also reflected in our evaluation, i.e. we do not evaluate how the voices propagate across bars, but only within each bar. Note that this process is different from processing each bar independently since our network (detailed in the next section) considers music content across bars.

3.4 Model

We design an end-to-end model (see Figure 2) that receives an input graph as described in Section 3.1 and produces an output graph as in Section 3.2. The model is organized as an encoder-decoder architecture.

The encoder receives an input graph created from a quantized MIDI score and passes it through three stacked Graph Convolutional Network (GCN) blocks to produce a node embedding for each note. We use the heterogeneous version of the Sage convolutional block [18] with a hidden size of 256; the update function for each node u is described by:

$$\begin{aligned} \mathbf{h}_{\mathcal{N}(u)}^{(l+1)} &= \sum (\{\mathbf{h}_v^l, \forall v \in \mathcal{N}(u)\}) \\ \mathbf{h}_u^{(l+1)} &= \sigma(\mathbf{W} \cdot \text{concat}(\mathbf{h}_u^l, \mathbf{h}_{\mathcal{N}(u)}^{l+1})) \end{aligned} \quad (1)$$

where $\mathcal{N}(u)$ are the neighbors of node u , σ is a non-linear activation function, \mathbf{W} is a learnable weight matrix.

The decoder consists of three parts that all use the same node embedding as input: i) a staff predictor; ii) a voice edge predictor; and iii) a chord clustering (i.e., a chord edge predictor). The *staff predictor* is a 2-layer Multi-Layer Perceptron (MLP) classifier that produces probabilities for each graph node (i.e., each note) to belong to the first or second staff. The *voice edge predictor* receives the embeddings of pairs of notes connected by edge candidates and produces a probability for each pair to be in the same voice. It works by concatenating the pairs of note embeddings and applying a 2-layer MLP. The final decoder part, *chord clustering*, receives the embeddings of pairs of notes connected by chord edge candidates (i.e., pairs of synchronous notes) and produces the probability for a pair to be merged into a chord. This is achieved by computing the cosine similarity between the elements of the pair. This process forces the

⁴ This may change for cross-bar beamings, but they are very rarely used in standard music notation (there are no occurrences in our datasets) and therefore we do not consider them in this work.

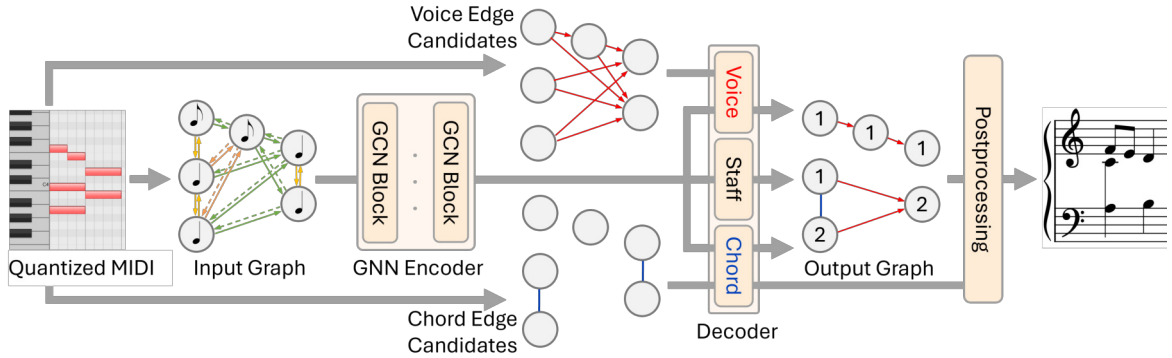


Figure 2. Our Architecture. For simplification, we display the output graph as having “hard” voice predictions, while these are probabilities over voice candidates.

node embeddings created by the decoder to be similar to each other for notes of the same chord, which helps the voice predictor produce consistent voice edge probabilities for notes of the same chord. We apply a threshold to pass from probabilities to decisions on which notes to cluster.

The complete model contains $\sim 3M$ parameters and we train it end-to-end with the (unweighted) sum of three *Binary Cross Entropy* loss functions, one for each task.

3.5 Postprocessing

A straightforward approach to deciding whether to connect two notes with a voice edge would be to threshold the predicted voice edge probabilities. However, even when using edge and chord candidates, we could still produce three kinds of invalid output: (1) multiple voices merging into one voice, (2) one voice splitting into multiple voices, and (3) notes in the same chord that are not in the same voice. To eliminate these issues, we add a postprocessing phase that accompanies our model and guarantees a valid output according to music engraving rules.

The first step, which we call *chord pooling*, merges all nodes that belong to the same chord to a single new “virtual node”. This is done by looking for the connected components considering only chord edges in the output graph, then *pooling* in a single node all original nodes in each connected component, creating a new node which has as incoming and outgoing voice edges all edges entering and exiting the original nodes, respectively. If multiple edges collapse in one edge (e.g. in the case of two consecutive chords in the same voice), the new edge has a probability that is the average of the corresponding edge probabilities.

After the first step, we are left with monophonic streams, which could still exhibit problems (1) and (2). We can solve this with the technique proposed in [12] for monophonic voices, i.e. by framing the voice assignment problem as a linear assignment problem [21] over the adjacency matrix obtained by the updated edge candidates Λ'_v . We follow the linear assignment step by unpooling or unmerging the nodes that were previously pooled, in this way, obtaining the original nodes again. During unpooling, the incoming edges and outgoing edges of the “virtual nodes” are reassigned to each original node, thus resolving problem (3).

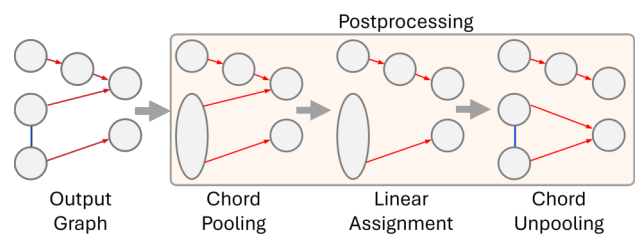


Figure 3. Output graph postprocessing. We do not display the predicted staff labels.

The complete postprocessing method is depicted in Figure 3. It is worth noting that the staff labels are not considered during the postprocessing phase, and we copy them unchanged to the postprocessed output graph.

3.6 Evaluation

We evaluate the predicted voice assignments with the metric proposed by Hiramatsu et al. [15], which formalizes the metric of McLeod and Steedman [22]. This is a version of the F1-score for voice separation [8] which is adapted to work on homophonic voices, by reducing the importance of notes if they are part of a chord. This is important since chords create many voice edges (e.g., two 4-note chords in the same voice are connected by 16 edges), which could potentially overshadow the importance of edges in monophonic voices (or voices with fewer/smaller chords).

Formally, the homophonic voice F1-score $F1$ is calculated as:

$$P = \frac{\sum_{i < j} a_{ij} \hat{a}_{ij} / \hat{w}_i}{\sum_{i < j} \hat{a}_{ij} / \hat{w}_i}, \quad R = \frac{\sum_{i < j} a_{ij} \hat{a}_{ij} / w_i}{\sum_{i < j} a_{ij} / w_i} \quad (2)$$

$$F1 = \frac{2PR}{P + R}$$

where $i < j$, in the sum, considers all pair of notes i, j such that $\text{offset}(i) < \text{onset}(j)$; a_{ij} , \hat{a}_{ij} are equal to 1 or 0 if a voice edge exists or not in the ground truth and predictions, respectively; and w_i and \hat{w}_i are the number of notes that are chorded together with the note i in the ground truth and predictions, respectively. Unlike [15], we consider

only notes j in the same bar of i , for the reasons presented in Section 3.3. We evaluate the staff prediction part of our model with binary accuracy, and we assess chord prediction with the F1 score computed on the chord edges.

3.7 From Network Prediction to Readable Output

The computation of voice and staff numbers is sufficient for the system evaluation, but not for producing a usable tool, which we are interested in in this paper. The missing step, to be described in this section, is the integration of the network predictions into a readable musical score. To achieve this integration we need to undertake two essential steps: beam together notes within the same voice, and infill rests to "fill holes" within each voice.

For the first step, we proceed according to the rules of engraving [2]. We beam two consecutive notes (or chords) in the same voices if their duration is less than a quarter note (excluding ties) unless they belong to different beats. Following the music notation convention we consider the compound time signatures, i.e., $\frac{6}{x}, \frac{9}{x}, \frac{12}{x}$ to have, respectively, 2, 3, and 4 beats. When confronted with tied notes, the algorithm prioritizes producing notations with the fewest number of notes, an heuristic which promotes easier-to-read notation [23].

The second step consists of introducing rests so that each voice fills the entire bar and can be correctly displayed. Some rests could be set as invisible to improve the graphical output when their presence and duration are easy to assume from other score elements, but we display all of them for simplicity. As for the notes, we choose the rest types (with eventual dots) to minimize the number of rests in the score.

The two steps described above cover common cases and produce a complete score in MEI format [24]. However, the score export is still a prototype, since developing one that is robust against all corner cases is an extremely complex task, and is outside the scope of this paper. Since score output problems may obscure the output of our system, we also develop a graph visualization tool. Both the input and output graphs (including the candidate edges) can be displayed on top of the musical score in an interactive web-based interface based on Verovio [25]. Some examples of the output graph visualization are in Figure 4.

4. EXPERIMENTS

We train our model with the ADAM optimizer with a learning rate of 0.001 and a weight decay of $5 * 10^{-4}$ for 100 epochs. For a quantitative evaluation, we compare our results with those of a baseline algorithm and the method proposed by Shibata et al. [5], on two rather diverse datasets.

Our baseline algorithm assigns all notes under C4 (middle C) to the second staff and the rest to the first. Then it groups all synchronous notes (per staff) as chords. Finally, it uses the time and pitch distances between the candidate pairs of notes as weights to be minimized during the linear assignment process (the same as we use in our postprocessing) which creates the voice edges.

4.1 Datasets

We use two piano datasets of different styles and difficulties to evaluate our system under diverse conditions. The ability to handle complex corner cases should not reduce the performance on easier (and more common) pieces.

The *J-Pop* dataset contains pop piano scores introduced by [5]. Most of the scores consist of accompaniment chords on the lower staff and some simple melodic lines on the upper staff. The dataset contains 811 scores; we randomly sampled 159 (roughly 20%) of these for testing and used the rest for training and validation.

The *DCML Romantic Corpus* is more challenging. It was created by [26] and contains piano pieces from the 17th to 20th centuries with some virtuosic quality. It includes characteristics such as cross-staff beaming, a high number of voices, challenging voicing, etc. Similarly to the pop dataset we randomly sample 77 out of the 393 scores (approx. 20%) and use the rest for training and validation.

The *J-Pop* dataset is available in MusicXML format, while the *DCML Romantic Corpus* is in Musescore file format. We use Musescore to convert DCML files to MusicXML and load them with the Python library Partitura [27] to extract the note list.

4.2 Results

Our model aims to be generic across a variety of music, therefore we train a single model on the joined training set of pop and classical scores, not two individual ones. The rules that govern the handling of voices may be fundamentally different in the two datasets, but we assign to the model the task of handling these differences. This approach ensures better future scalability on bigger and more diverse datasets. We compute the metrics separately on the test part of our two datasets.

Table 1 reports results for three versions of our graph-based model: the complete model from Section 3, a variant without postprocessing, and a variant without chord prediction and postprocessing (our postprocessing technique cannot be run without the chord prediction task, since it pools nodes that belong to the same predicted chord). The method of Shibata requires the specification of the number of voices per staff. For compactness, we report only the results with one voice per staff (2 voices total); the results degrade by increasing the number of voices.

Our results show that even our system without pooling and without postprocessing obtains consistently better results than both Shibata et al. [5] and our baseline. Interestingly, the chord prediction task improves the Voice F1 results even when the post-processing is not used; this confirms the benefits of multi-task training, and of enforcing notes of the same chord to have similar representations in the hidden space, with cosine similarity, to predict coherent voice edges. However, we observe a reduction in staff accuracy, probably for the same reason, since the same hidden representation is also used to predict chords, making it harder (though not impossible) to split notes of the same chord in different staves. When the full system is used, there are further improvements in Voice F1.

Dataset	J-pop Dataset			DCML Romantic Corpus		
	Staff Acc	Chord F1	Voice F1	Staff Acc	Chord F1	Voice F1
Baseline	89.9	86.9	85.4	80.7	65.2	78.2
Shibata et al. [5]	92.8	-	92.2	88.5	-	84.9
GNN wo Chord wo Post	96.5 ± 0.1	-	95.2 ± 1.9	91.5 ± 0.1	-	87.2 ± 3.3
GNN wo Post	96.3 ± 0.1	94.9 ± 0.1	95.7 ± 0.4	91.0 ± 0.1	79.5 ± 0.4	88.9 ± 0.4
GNN	96.3 ± 0.1	94.9 ± 0.1	96.6 ± 0.1	91.0 ± 0.1	79.5 ± 0.4	89.9 ± 0.2

Table 1. Metrics for our the J-Pop and DCML test sets. “GNN” denotes our method, without postprocessing (“GNN wo Post”), and without both postprocessing and chord prediction parts (“GNN wo Chord wo Post”). All GNN model runs are repeated 5 times: ± refers to the standard deviation of results across runs.

We are also evaluate our system on the bar-level and study performances for music excerpts of varying difficulties. We compute the voice F1 score for each bar and average them based on the number of voices in the ground truth. We compare with Shibata et al. [5] with 1 & 2 voices per staff (vps). Table 2 shows the results for the DCML Romantic Corpus. Both our model and [5] perform best with 2 voices, the most common number in our dataset. Interestingly, Shibata et al. approach with 2 vps never outperforms vps 1, not even when the target number of voices is 3 or 4, a situation that vps 1 cannot handle. This can be explained by the fact that Shibata et al. parameters were tuned on a simpler dataset, and accepting more voices creates more errors than benefits. Setting vps > 2 consistently degraded the performances, probably also for similar reasons.

#Voices	#Bars	GNN	[5] 1vps	[5] 2vps
1	322	96.6	88.3	87.9
2	4576	94.1	89.3	88.1
3	2464	89.0	84.2	81.5
4	719	81.6	80.5	75.1
5	99	81.6	76.7	73.7
6	17	78.4	68.9	61.6

Table 2. Voice F1 score aggregated by bars with the same number of voices in the ground truth, on the DCML Corpus. Shibata et al. [5] is used with 1 and 2 voices per staff (vps).

4.3 Qualitative Analysis

Let us take a closer look into the predictions of our deep-learning approach (GNN) on the excerpt of Figure 4 produced by our visualization tool. Our approach captures correctly the cross-staff voice in the first two bars, while such a situation causes performance degradation for all other voice separation approaches that don’t support it. We observe some disagreements with the original score in Measure 3: our model predicts a single chord (instead of splitting across the staff) containing all the synchronous syncopated quarter notes, and also mispredicts the staff for the first D#4 note. A more in-depth study of why this happens is not trivial, as neural networks are not interpretable. This is a drawback compared to heuristic-based separation techniques.

Synchronous notes with the same pitch are problematic.

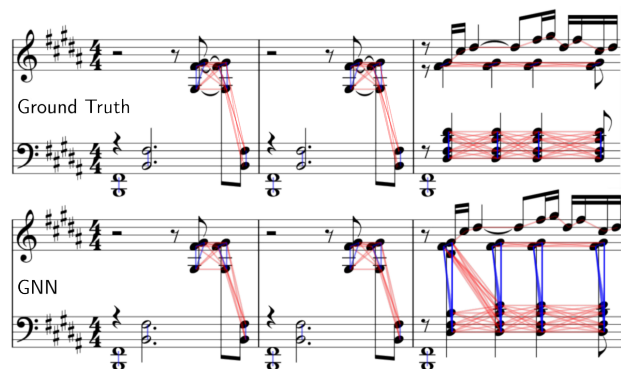


Figure 4. Comparison of voice and staff assignment between the original score (Ground Truth) and our method (GNN) on the first bars of C. Debussy’s Estampes-Pagodes. Voice edges are drawn in red and chord edges in blue.

Our system can predict different voices for these notes, while Shibata et al. always predict them as a chord in the same voice, and this reduces the performances for pieces that contain a lot of them, like Schumann Kinderszenen Op.15. For fairness, we should note that we should expect the output of a music transcription system to only contain one of these notes, instead of multiple like in our current input. An enhancement of our system would then be able to receive a single note as input, assign multiple voices to it (with multiple incoming and outgoing edges) and then split it into multiple notes. Another current limitation of our system is the missing support for grace notes, which in the actual version are ignored and removed from the input.

5. CONCLUSION AND FUTURE WORK

This paper presented a novel graph-based method for homophonic voice separation and staff prediction in symbolic piano music. Our experiments highlight our system’s effectiveness compared to previous approaches. Notably, we obtained consistent improvements over two datasets of different styles with a single model.

Future work will focus on integrating grace notes and the possibility of multiple voices converging on a single note. We aim to create a framework that produces complete engravings from quantized MIDI, including the prediction of clef changes, beams, pitch spelling, and key signatures.

6. ACKNOWLEDGEMENTS

This work is supported by the European Research Council (ERC) under the EU's Horizon 2020 research & innovation programme, grant agreement No. 101019375 (*Whither Music?*), and the Federal State of Upper Austria (LIT AI Lab).

7. REFERENCES

- [1] E. Cambouropoulos, "Voice and stream: Perceptual and computational modeling of voice separation," *Music Perception*, vol. 26, no. 1, pp. 75–94, 2008.
- [2] E. Gould, *Behind bars: the definitive guide to music notation*. Faber Music Ltd, 2016.
- [3] D. Makris, I. Karydis, and E. Cambouropoulos, "VISA3: Refining the voice integration/segregation algorithm," in *Proceedings of the Sound and Music Computing Conference*, 2016.
- [4] Y.-S. Huang and Y.-H. Yang, "Pop music transformer: Beat-based modeling and generation of expressive pop piano compositions," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, p. 1180–1188.
- [5] K. Shibata, E. Nakamura, and K. Yoshii, "Non-local musical statistics as guides for audio-to-score piano transcription," *Information Sciences*, vol. 566, pp. 262–280, 2021.
- [6] M. Terao, E. Nakamura, and K. Yoshii, "Neural band-to-piano score arrangement with stepless difficulty control," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.
- [7] E. Chew and X. Wu, "Separating voices in polyphonic music: A contig mapping approach," in *Proceedings of the International Symposium on Computer Music Modeling and Retrieval*. Springer, 2004.
- [8] B. Duane and B. Pardo, "Streaming from midi using constraint satisfaction optimization and sequence alignment," in *Proceedings of the International Computer Music Conference (ICMC)*, 2009.
- [9] P. Gray and R. C. Bunescu, "A neural greedy model for voice separation in symbolic music," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2016.
- [10] A. McLeod and M. Steedman, "HMM-based voice separation of midi performance," *Journal of New Music Research*, vol. 45, no. 1, pp. 17–26, 2016.
- [11] Y.-W. Hsiao and L. Su, "Learning note-to-note affinity for voice segregation and melody line identification of symbolic music data," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2021.
- [12] E. Karystinaios, F. Foscarin, and G. Widmer, "Musical voice separation as link prediction: Modeling a musical perception task as a multi-trajectory tracking problem," in *International Joint Conference on Artificial Intelligence (IJCAI)*, 2023.
- [13] J. Kilian and H. H. Hoos, "Voice separation—a local optimization approach," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*. Citeseer, 2002.
- [14] M. Suzuki, "Score transformer: Generating musical score from note-level representation," in *Proceedings of the 3rd ACM International Conference on Multimedia in Asia*, 2021, pp. 1–7.
- [15] Y. Hiramatsu, E. Nakamura, and K. Yoshii, "Joint estimation of note values and voices for audio-to-score piano transcription," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2021, pp. 278–284.
- [16] L. Liu, Q. Kong, G. Morfi, E. Benetos *et al.*, "Performance midi-to-score conversion by neural beat tracking," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2022.
- [17] J. Zhao, G. Xia, and Y. Wang, "Q&a: Query-based representation learning for multi-track symbolic music re-arrangement," in *International Joint Conference on Artificial Intelligence (IJCAI)*, 2023.
- [18] W. L. Hamilton, R. Ying, and J. Leskovec, "Representation learning on graphs: Methods and applications," *IEEE Data Engineering Bulletin*, vol. 40, no. 3, pp. 52–74, 2017.
- [19] F. Foscarin, N. Audebert, and R. Fournier S'niehotta, "PKSpell: Data-driven pitch spelling and key signature estimation," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2021.
- [20] A. Aljanaki, S. Kalonaris, G. Micchi, and E. Nichols, "MCMA: A symbolic multitrack contrapuntal music archive," *Empirical Musicology Review*, vol. 16, no. 1, pp. 99–105, 2021.
- [21] R. E. Burkard and E. Cela, "Linear assignment problems and extensions," in *Handbook of combinatorial optimization*. Springer, 1999, pp. 75–149.
- [22] A. McLeod and M. Steedman, "Evaluating automatic polyphonic music transcription," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2018, pp. 42–49.
- [23] F. Foscarin, F. Jacquemard, and P. Rigaux, "Modeling and learning rhythm structure," in *Sound and Music Computing Conference (SMC)*, 2019.

- [24] P. Roland, “The music encoding initiative (mei),” in *Proceedings of the First International Conference on Musical Applications Using XML*, vol. 1060. Citeseer, 2002, pp. 55–59.
- [25] L. Pugin, R. Zitellini, and P. Roland, “Verovio: A library for engraving mei music notation into svg,” in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2014.
- [26] J. Hentschel, Y. Rammos, F. C. Moss, M. Neuwirth, and M. Rohrmeier, “An annotated corpus of tonal piano music from the long 19th century,” *Empirical Musicology Review*, vol. 18, no. 1, pp. 84–95, 2023.
- [27] C. E. Cancino-Chacón, S. D. Peter, E. Karystinaios, F. Foscari, M. Grachten, and G. Widmer, “Partitura: A python package for symbolic music processing,” in *Proceedings of the Music Encoding Conference (MEC)*, Halifax, Canada, 2022.

FROM AUDIO ENCODERS TO PIANO JUDGES: BENCHMARKING PERFORMANCE UNDERSTANDING FOR SOLO PIANO

Huan Zhang Jinhua Liang Simon Dixon
Centre for Digital Music, Queen Mary University of London, UK

huan.zhang@qmul.ac.uk

ABSTRACT

Our study investigates an approach for understanding musical performances through the lens of audio encoding models, focusing on the domain of solo Western classical piano music. Compared to composition-level attribute understanding such as key or genre, we identify a knowledge gap in performance-level music understanding, and address three critical tasks: expertise ranking, difficulty estimation, and piano technique detection, introducing a comprehensive Pianism-Labeling Dataset (PLD) for this purpose. We leverage pre-trained audio encoders, specifically Jukebox, Audio-MAE, MERT, and DAC, demonstrating varied capabilities in tackling downstream tasks, to explore whether domain-specific fine-tuning enhances capability in capturing performance nuances. Our best approach achieved 93.6% accuracy in expertise ranking, 33.7% in difficulty estimation, and 46.7% in technique detection, with Audio-MAE as the overall most effective encoder. Finally, we conducted a case study on Chopin Piano Competition data using trained models for expertise ranking, which highlights the challenge of accurately assessing top-tier performances.

1. INTRODUCTION

Traditional music understanding tasks focus on composition-level attributes: key, tempo, genre and instrumentation are widely explored [1–3]. These attributes are not only tagged individually via end-to-end approaches but have also been the focus of foundation models and various musical representations aimed at learning them in a unified manner [4, 5], facilitating cross-modal understanding [6, 7].

However, a large portion of human music activity is focused not on the composed songs or pieces themselves, but on the process of learning and performing them [8]. Despite its great importance to the vast community of students, teachers and musicians, the ability to understand performance nuances (challenging techniques, skill varieties, stylistic differences, difficulty grading, etc.) has not been grasped by machines. Sporadic experiments [9] of

these tasks are conducted, often on small-scale [10, 11] or proprietary [12–14] datasets. Performance understanding, in contrast to the more recognized composition-level music understanding, suffers from scarcity of data [15, 16], ambiguity of tasks [17], and the inherent complexity of modelling and representing expressive elements in performances [18, 19].

Meanwhile, unified representations and foundation models have advanced several fields by providing robust and versatile frameworks [6, 20], demonstrating their potential to overcome challenges related to data scarcity and task specificity. Building on this precedent and their applications to compositional-level understanding [21], we extend the capacities of pre-trained audio encoders such as MERT [22] and MULE [5] into the performance-understanding realm, investigating the shared knowledge between composition- and performance-level understanding: Do pre-trained audio encoders capture performance nuances? Can they categorize performance-related attributes? If not, how can we improve their performance?

This work is a first step in filling the gaps within the performance understanding realm. Applying domain-adaptation to pre-trained audio encoders, we work towards a *piano judge* that specializes in ranking performers’ skill level, determining the given repertoire’s difficulty and core techniques, thus pursuing a human piano teacher’s capability and paving the way to performance understanding in an educational context. Our contributions¹ include:

1. We benchmark three tasks in the realm of audio performance understanding: expertise ranking, difficulty estimation, and solo piano technique detection.
2. We leverage four audio representation learners (Jukebox, Audio-MAE, MERT, DAC) and compare their capabilities in tackling the downstream tasks.
3. We release the Pianism-Labeling Dataset (PLD) with detailed labeling curated for the three tasks, the first large-scale dataset (136 hrs in total) that aims to address performance understanding.
4. We fine-tune DAC [23] and AudioMAE [24] by domain-adaptation with solo piano, and compare their performances with pre-trained versions.
5. We conduct a case study on Chopin Piano Competition data (*ICPC-2015*), exploring how a trained expertise ranking model can be transferred to rank candidates in the most prestigious competition of the pianistic scene.



© H. Zhang, J. Liang, S. Dixon. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** H. Zhang, J. Liang, S. Dixon, “From Audio Encoders to Piano Judges: Benchmarking Performance Understanding for Solo Piano”, in *Proc. of the 25th Int. Society for Music Information Retrieval Conf.*, San Francisco, USA, 2024.

¹ Code available at: <https://github.com/anusfoil/PianoJudges>

2. RELATED WORK

2.1 Performance and education focused understanding

The exploration of performance through recordings provides rich resources for music understanding. Automatic performance analysis (APA) [17] delves into dimensions from dynamics [25, 26] to timing [27, 28], forming the basis for tasks such as performer identification and automatic music performance assessment (MPA) [15]. The former seeks to attribute performances to their respective musicians based on stylistic and technical signatures [29, 30], and the latter aims to evaluate the quality and expression of performances. MPA approaches can be further divided by the level of proficiency. For novice players, the emphasis is on technical accuracy, ensuring correct notes and rhythm via score alignment [16, 31] or detecting conspicuous mistake regions [14] in a score-free context. Advanced performers are assessed by their expression and musicality, usually in the form of predicting rating scores on multiple dimensions [12, 32]. Recently, the release of feedback-based assessment data [33, 34] offers the possibility to conduct multimodal MPA in a more personalized manner.

On the other hand, the analysis of performances in an educational context emphasizes the identification of challenges and learning opportunities within the repertoire: expert-annotated difficulty level is predicted from symbolic scores [35] via a machine learning classification approach that merges musicologically-inspired score features. At a more granular level, we would like to identify instrument-specific techniques that demand practice. For example, techniques such as *acciaccatura* and *portamento* on Chinese bamboo flute can be identified from spectro-temporal patterns [36], but similar problems have yet to be explored on piano because of the homogeneity of piano sound. Other learning aid information such as fingering [9, 37] and bowing [38] can also be predicted. In this work, we focus on expertise, difficulty and technique estimation by extracting relevant information from performance audio. This work is the first of its kind in piano technique detection and expertise ranking, and the first to use an audio representation approach for difficulty estimation [9, 39].

2.2 Leveraging audio representations for downstream tasks

The surge of learning audio representations was originally motivated by generative models such as AudioLM [40] and MusicLM [41]. Jukebox [42] is a generative model trained on 1.2M songs. Subsequent work [4, 43] has shown that Jukebox’s representations can be effective features for task-specific linear classifiers. Jukebox embeddings have also been employed in multimodal learning [6] of music captioning and reasoning tasks. MERT [22] uses masked language modelling (MLM) style acoustic self-supervised pre-training. With a music teacher and an acoustic teacher, MERT demonstrates good performance in downstream music understanding tasks and extends its music understanding ability into question answering and captioning [44] by generating music representations to aid language models.

Audio-MAE [24] is a vanilla 12-layer transformer that learns to reconstruct randomly-masked spectrogram patches. The output feature map from the penultimate block of an Audio-MAE encoder has been used to encode fine-grained patterns in audio [45]. Different from previous approaches, Descript-Audio-Codec (DAC) [23] is a neural audio compression autoencoder that compresses high-dimensional signals into lower dimensional discrete tokens. DAC has been proven useful in a generative context [46], but there have been few attempts to explore it with downstream understanding tasks [47].

The four aforementioned audio representations are chosen for our investigation. Since they are constructed from different theoretical approaches (quantized codecs vs. continuous spectrograms) and trained on different data (general audio vs. music), this variety presents an opportunity to evaluate the extent to which the encoded information contributes to performance understanding.

3. METHODOLOGY

3.1 Downstream problem definitions

3.1.1 Expertise ranking

We formulate our assessment into a ranking problem: given audio performances p_1 and p_2 , which one has the higher expertise? We define three coarse levels of expertise (beginner, advanced and virtuoso), represented by integers 0, 1 and 2, respectively, and define a function Q which maps a performance to one of these levels. Instead of directly predicting the absolute expertise level Q , we learn a 2-way or 4-way ranking function between each pair of performances from different levels, R_2 or R_4 , as below:

$$R_2 = \begin{cases} 0 & Q(p_1) < Q(p_2) \\ 1 & Q(p_1) > Q(p_2) \end{cases} \quad R_4 = \begin{cases} 0 & Q(p_2) - Q(p_1) = 2 \\ 1 & Q(p_2) - Q(p_1) = 1 \\ 2 & Q(p_1) - Q(p_2) = 1 \\ 3 & Q(p_1) - Q(p_2) = 2 \end{cases} \quad (1)$$

The motivation is to teach the model a relative notion of expertise, instead of an absolute level or category of the performance quality. In real life and competition settings (as will be discussed in Sec 5.1.1), we are more interested in the comparative skill level among a set of candidates.

3.1.2 Difficulty estimation

Following the literature [9, 18, 35] on difficulty level prediction, we formulate the problem as a classification task with 9 difficulty classes, given the dataset described in Section 4.2, which has 9 levels of difficulty annotation. Given that the difficulty annotation is subjective and boundaries between levels are fuzzy, we also report the results of 3-class estimation by merging the level groups, as in [35].

3.1.3 Technique identification

Given a piece, a piano teacher can immediately identify the most challenging passage(s) that would require students hours of practice to master: intense octave runs, fast flowing scales, repeating notes that require finger iteration, etc.

Encoder	C	F (Hz)	Dim
Jukebox	2048	345	64
MERT	-	75	1024
Audio-MAE	-	51.2	768
DAC	9×1024	87	1024
Spectrogram	-	150	128

Table 1. Specifications of the audio encoders as well as the spectrogram baseline: C is the codebook size, F is the frame rate in Hz and Dim is the hidden dimension of the embedding (mel-bins for spectrogram).

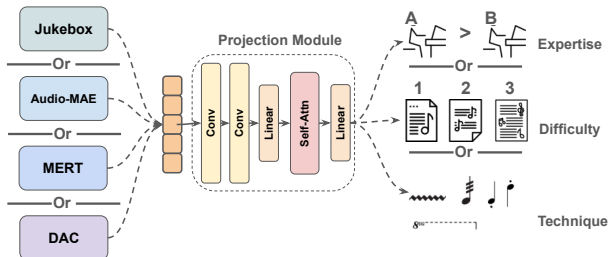


Figure 1. Overview of our tasks and experiment pipeline. For the expertise ranking, two audio embeddings are concatenated in the time dimension.

In the technique-specific dataset (Section 4.3), we include 7 common techniques and formulate a multi-label classification task for technique identification. Given that our labels are relatively sparse, we also experiment with the case of single-label prediction in which predicting any one of the multiple labels is considered correct.

3.2 Audio embeddings and encoder fine-tuning

An overview of the used audio embeddings is given in Table 1. For Jukebox, we employ the 345 Hz sample rate encoding, and for Audio-MAE, the 768-dimensional embedding is taken from the ViT-B Transformer encoder. Additionally, we considered a spectrogram baseline, as a low-level representation to compare with the trained embeddings. We use 128 mel bins, an FFT of 400 samples, and hop size of 160 samples, resulting in a spectrogram with frame rate of 150 Hz and 128 dimensions that feeds into the prediction head module like the trained embeddings.

We examine whether fine-tuning the two generic-audio-trained encoders DAC and Audio-MAE with domain-specific data results in a performance boost. The two encoders are fine-tuned using their original self-supervision objective on around 2k hours of solo piano recordings, from datasets of MAESTRO [48], ATEPP [49], SMD [50], Mazurkas² as well as the novel PLD data introduced in this work. For DAC, the fine-tuning lasts for 25k iterations while the Audio-MAE is fine-tuned for 64 epochs.

3.3 Experiments

For all encoders, we first compute 10-second segment audio embeddings (or spectrograms), and include a maximum of 5

²<http://www.charm.rhul.ac.uk/index.html>

	task type	classes	tracks	len. (s)
Expertise	Multi-class	2 or 4	1694	167.4
<i>ICPC-2015</i>	Multi-class	2	137	1827.0
Difficulty	Multi-class	3 or 9	737	269.8
Techniques	Multi-label	7	222	45.5

Table 2. Dataset statistics (number of classes, number of tracks and average duration) for each task.

minutes (30 segments) of audio as input with padding. The concatenated embedding of each audio track is of shape $(30, F \times 10, D)$ where F is the frame rate and D is the embedding dimension as shown in Table 1.

Given the audio embedding, we transform it through a prediction head module that consists of two 2D convolutional layers ($n_{kernel} = 7, n_{stride} = 5$), one linear layer to align different input dimensions, and one self-attention layer ($n_{heads} = 2, d = 128$), followed by a final linear layer that projects to the desired classes of each task. A full pipeline of the experiments is shown in Figure 1. As is standard practice [4, 43], we maintain a straightforward projection module design, aiming to minimize its influence on the probing performance.

Regarding each individual task, we run a grid search on the hyper-parameters for learning rate, weight decay, batch size, etc. The details for the final training parameters for each task are documented in the project page³. All fine-tuning and training are conducted on one NVIDIA A5000.

4. PIANISM-LABELING DATASET

The pianism-labeling dataset (PLD) includes audio and annotations for three notions that are centrally relevant to pianism and piano education: **expertise**, piece **technique** and **difficulty**, where dataset statistics are specified in Table 2. All of the labeling, metadata correspondence, as well as examples are available on the project page.

4.1 Expertise

We curated a collection of solo piano recordings from YouTube, each annotated with an expertise level. Their categorization was based on information gleaned from the YouTube channels’ descriptions, which provided insights into the background of the recordings. This categorization process was validated by two college-level piano students to ensure accuracy.

- **Beginner** (562): Amateur level, featuring mostly adult self-taught learners’ practice recordings.
- **Advanced** (570): Performances of music students and junior competition recordings.
- **Virtuoso** (562): Famous pianists’ recordings sourced from the ATEPP [49] dataset. To balance with the other groups, we randomly select a subset of 562 of the 11K recordings.

The repertoire of selected performances is mainly focused on the Western classical repertoire, with some rearranged folk and pop songs at the Beginner level. Indeed,

³<https://bit.ly/3SYzozY>

it is challenging to align the performed repertoire across levels since the complexity of played pieces increases with the expertise: e.g. Beginners’ pieces are shorter (av. 128.9 s) than Advanced (201.1 s) or Virtuoso (171.8 s) tracks.

In experiments, the three levels are first individually split into train and test subsets and then paired up randomly. Each recording only shows up once in the pairs to prevent leakage, which results in 2694 pairs in training.

4.1.1 ICPC-2015

In this task we aim to assess whether the learnt comparative ranking objective can be applied to the professional domain, the International Chopin Piano Competition, with data gathered from the 2015 edition (*ICPC-2015 dataset*)⁴. We employ only the preliminary round performances to ensure limits on the length and instrumentation (i.e. solo), and assume that the overall better players clearly demonstrate their skills in the preliminary round performance. Out of 160 candidates, 137 recordings are successfully retrieved.

We compile the data into ranking pairs similar to Section 4.1, by first assigning a score $S(c)$ for each candidate based on their progression into the following rounds. For every round that the candidate passes into, the score is incremented with 1 point. For candidates a, b with their respective scores $S(a), S(b)$, all preliminary round recordings are formed into pairs with ranking as in Eq. 1. As shown in Table 2, the preliminary round recordings have an average duration of 30 minutes. Thus, we obtain paired ranking results for each pair of 5-minute segments (30 segments in total) and use majority voting to obtain the final rank among two recordings.

4.2 Difficulty

We employ the *Can I Play It?* (CIPI) [35] dataset for our task of difficulty prediction. Given that the original dataset is sourced in symbolic MusicXML, we obtain the performance audio from YouTube by querying the metadata followed by manual correction to enforce piece alignment. Note that the performances are sourced from different levels of playing rather than virtuoso recordings only, with the aim of learning a more general view of audio difficulty. In the CIPI dataset, difficulty labels are annotated by Henle Verlag⁵, a renowned publisher in the music education community. The ratings range from 1-9 and span 29 composers. Note that we split the movements from sonata or other multi-movement compositions, resulting in the 737 audio tracks shown in Table 2 compared to 637 compositions in the original metadata. We also use the same train-test split as the original dataset.

4.3 Techniques

The technique dataset contains 222 recordings with an average duration of 45 seconds, demonstrating one or more canonical piano techniques from seven categories taken from piano practice literature [51]. The excerpts are taken

from etude books like Beyer or Czerny, or passages from performance repertoire (e.g. dense octave run from *Chopin op.25 no.10*). Besides YouTube sourcing, 41 out of 222 recordings are recorded by the authors, if the specific passages containing the techniques are not publicly available in any recording. The categories of techniques are:

- **Scales** (48): Pure scale run across octaves. Can be both hands or one hand.
- **Arpeggios** (40): Pure arpeggio run across octaves, or music passages that are accompanied with arpeggios, or melody that is constructed on arpeggiated chords.
- **Ornaments** (31): Including grace notes, trills, *mordents*, *acciaccatura*. Note that we do not balance these subclasses, and the most common ornament in our samples is grace note.
- **Repeated notes** (35): Musical passages that feature a series of repeated single notes.
- **Double notes** (36): Musical passages that feature sequences of simultaneous intervals (mostly thirds, but also fourths and sixths), where the intervals are performed with one hand.
- **Octaves** (35): We differentiate octaves from double notes because of their sheer importance in piano repertoire, as well as their distinctive sonority.
- **Staccato** (41): Musical passages that are predominately performed by *staccato* articulation.

We formulate the prediction task as multi-label classification since a musical passage is often associated with multiple techniques. Among the 222 recordings, we have 40 labeled with two techniques and two recordings with three techniques. Note that besides scales and arpeggios, few other techniques exist in their pure form (e.g. an entire music passage of trills). Thus we aim to identify the most prominent technique present in the recording.

5. RESULTS

5.1 Expertise Ranking

We train the projection module in 2-way and 4-way ranking as described in Section 3.1.1, and show results in Table 3 (left). For 2-way ranking, we achieve up to 93.56% accuracy, indicating a clear distinction between recordings of varying levels of expertise in most cases. Audio-MAE outperforms the other three audio encoders while Jukebox embeddings contain the least information for discerning the level of playing. The result of 4-way prediction is similar with Audio-MAE performing the best with 84% accuracy, indicating a good capability to distinguish larger expertise differences (beginner vs. virtuoso) from smaller ones. The baseline spectrogram achieves much lower metrics on both classifications, indicating that the pre-trained encoders capture more relevant nuances of musical performance. However, we are also aware that the three levels of data differ not only on performance but also on repertoire and recording environment. The effect of fine-tuning with solo piano domain data is not salient in this task: the fine-tuned Audio-MAE achieved roughly the same performance while DAC actually declined.

⁴ <https://github.com/cyrta/ICPC2015-dataset>

⁵ <https://www.henle.de/>

	Expertise Ranking				Difficulty Estimation				Technique Identification				
	2-way		4-way		9-way		3-way		Multi			Single	
	Acc	F1	Acc	F1	Acc ₀	Acc ₁	Acc ₀	F1	mAP	AUC	Acc	Acc	F1
<i>pre-trained</i>													
Spec	75.90	74.73	52.34	49.94	32.98	59.17	67.21	66.75	57.49	71.13	73.02	46.67	39.26
Jukebox	84.51	83.79	60.41	56.75	33.41	55.36	60.49	58.27	49.33	59.79	73.33	25.44	23.53
Audio-MAE	93.48	92.84	84.21	81.20	31.60	66.09	79.03	75.11	60.69	67.51	77.46	42.22	39.81
MERT	89.48	88.73	82.12	78.81	26.55	62.28	73.13	71.38	55.76	69.05	79.37	37.78	35.85
DAC	86.84	87.91	77.77	76.24	27.61	59.86	69.64	69.87	48.50	57.87	78.73	24.44	23.61
<i>fine-tuned</i>													
Audio-MAE	93.56	90.22	82.26	77.82	33.67	60.21	77.73	75.84	61.81	67.61	79.05	35.56	33.73
DAC	82.87	81.83	78.41	76.23	28.63	61.59	64.45	62.34	50.77	59.80	79.68	26.67	25.66

Table 3. From left to right: results of 2-way and 4-way expertise ranking, 9-way and 3-way difficulty estimation, multi-label and single-label technique prediction. Best results are highlighted in bold.

5.1.1 Discussion: How far are we from predicting the Chopin Competition winner?

From the trained 2-way expertise ranking module, we apply the *ICPC-2015* pairs as a testing set as described in Section 4.1.1. Ideally, the model should discern the three levels of piano expertise by identifying specific nuances in performance that distinguish, for example, virtuosi from advanced students. Such insights could then be applicable to evaluating competition-level performances. In Table 4, “fitting” indicates that we first fit the trained model on half of the candidates’ pairs for 5 epochs and test on the other half. Without fitting, we only evaluate on these same testing pairs using the model trained in Section 5.1.

	w/o. fitting		w. fitting	
	Acc	F1	Acc	F1
<i>pre-trained</i>				
Spec	52.91	52.79	49.27	49.04
Jukebox	46.63	44.92	48.27	47.13
Audio-MAE	56.86	55.86	59.08	58.76
MERT	49.07	46.78	53.05	52.54
DAC	42.17	41.71	53.67	53.67
<i>fine-tuned</i>				
Audio-MAE	54.32	50.14	54.89	49.81
DAC	60.49	60.27	59.87	59.84

Table 4. 2-way paired-ranking test result for the competition dataset *ICPC-2015*.

Several interesting observations are made from this experiment: 1) Transferring the learnt expertise ranking into assessing competition-level playing (which should all belong to the virtuoso tier within our training) is challenging, considering the random guess baseline of 50% accuracy in predicting the better performer within a pair. The best we achieve is slightly above 60%, possibly because the outcomes of competitions often transcend mere audio content to include performative expression like gestures, resulting in a *sight over sound* phenomenon [52]. 2) Adaptation on the competition set does not significantly boost the performance. For the pre-trained embeddings the accuracies slightly increase after fitting, but it has no effects on the fine-tuned embeddings. 3) The fine-tuned DAC embeddings, despite having a lower performance in the ranking

task with three levels, largely outperform other models in ranking the candidates in a competition setting.

Using the paired prediction results from best model (fine-tuned DAC w/o. fitting), we translate pair-wise predictions into a global ranking. Each candidate is ranked by how many wins they obtain in the ‘paired matches’. Each candidate is involved in 272 pairs, given 137 candidates and we infer on each pair (136) and its inverse. Figure 2 shows the relationship between our predicted candidate win counts and the preliminary round pass hit-rate (i.e. what proportion of candidates actually passed the preliminary round). *Michał Szymanowski* is the predicted best candidate who wins in the most pairs. Overall, there exists a good correlation between our predicted win counts rank and candidates’ ground truth performance: the top 18 predicted candidates all passed the preliminary round, with many of them progressing into round 2 or 3 (demonstrated by the color in Figure 2). Down to the cut-off threshold of half of the candidates, 65% of them passed the preliminary round. Finalists, however, are not necessarily predicted accurately: the winner *Seong-Jin Cho* only “wins” 39 matches and is placed towards the end in this rank, as is the third placed *Kate Liu*. Only *Charles Hamelin* (2nd place) is placed relatively high in our ranking.

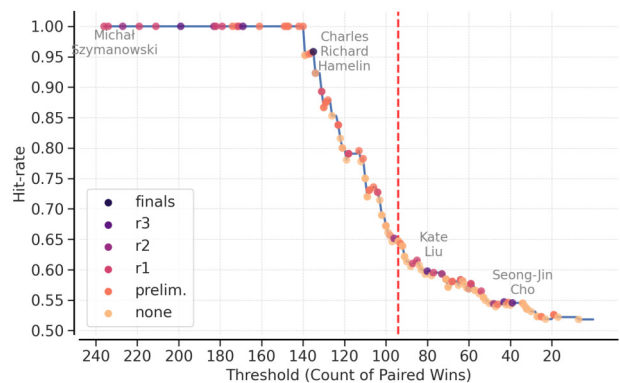


Figure 2. Paired win count threshold vs. hit-rate for preliminary round pass prediction. Each candidate is a data point colored by their ground-truth result tier (i.e. the highest round they progressed from). The red dashed line is the cut-off of half the candidates.

5.2 Difficulty estimation

The difficulty estimation experiments are defined in Section 3.1.2 on the *CIP1* dataset. In Table 3 (middle), we report the *accuracy within n* (Acc_n) for the 9 class prediction. Defined as Eq. 2, Acc_n aligns with the ordinal nature of the task and we observe results for $n = 0$ (exact match) and $n = 1$ (allowing for one-class deviation).

$$Acc_n = \frac{1}{|C|} \sum_{c \in C} \frac{|\{y \in S_c : |\hat{f}(x) - c| \leq n\}|}{|S_c|} \quad (2)$$

The Audio-MAE embeddings yield overall the best performance for both 9-way and 3-way estimation. But it is worth noting that the untrained spectrogram baseline actually achieved accuracy metrics on-par with the audio encoders (32.98% vs. 33.67% in 9-way estimation), even higher than the worst-performing embedding of DAC.

The best we achieve with 9-class Acc_0 is 33.67% (compared to the same-set symbolic data baseline [35] of 39.47%). However, this is based on the fact that our audio embeddings are capped to 5 minutes, removing the effect of the major feature of piece length. For the 3-way classification we achieved accuracy that is on-par with the symbolic baseline, with the best Acc of 79.03%, demonstrating that the complexity of piano repertoire can also be encoded with the current pre-trained representation.

Interestingly, the Acc_0 and Acc_1 metrics do not improve hand-in-hand: Jukebox embeddings achieved the highest Acc_0 among the pre-trained models, but performed worst on Acc_1 since its prediction is sparse and scattered from observing the confusion matrix. The fine-tuned models exhibit a modest enhancement in performance metrics, as in the Acc_0 for Audio-MAE and 3-way FI for DAC, as well as better generalization and less overfitting.

5.3 Technique identification

The technique identification experiment is performed as both multi-label and single-label prediction, as formulated in Section 3.1.3. In Table 3 (right), we report the mean-Averaged-Precision (mAP) and Area Under the Receiver Operating Characteristic Curve (AUC). The former accounts for the balance between precision and recall, while the latter computes area under the false positive rate and true positive rate (recall) which reflects the influence of the true negatives. We also note the multi-label accuracy Acc which accounts for all binary predictions of each class.

The most important observation on the result is that the spectrogram representation easily outperforms the audio encoder embeddings on this task, especially on the single-label prediction case (46.67%). This offers an interesting perspective on the learned embedding content: exact note onsets and texture patterns (that are associated with the piano technique classes) seem to be overlooked by the embeddings, capturing less performance-related details compared to the lower-level spectrogram. The results demonstrate that DAC and JukeBox are the least informative audio embeddings for this task (24.44% and 25.44%). Audio-MAE is the best-performing audio encoder, but the single-label prediction results do not improve with fine-tuning. On the

other hand, fine-tuning DAC on the solo piano data improved performance on this task by 2%, compared with its pre-trained version.

To gain a better understanding of the identified techniques we observe the class-wise mAP from the best-performing representation of spectrogram. As depicted in Figure 3, Repeated Notes emerge as the most accurately identified technique. Conversely, the Staccato class exhibits a decline in performance throughout the training, hinting at a potential acoustic overlap with Repeated Notes, as suggested by prior research [53]. Meanwhile, the precision for other techniques shows consistent improvement during training, achieving 40% to 60% even in more distinct technique categories like Scales and Arpeggios. However, with the highest accuracy for single-label 7-way prediction being 46.67%, it is clear that the model’s ability to pinpoint techniques could be further refined, especially considering these are easily discernible to the human ear.

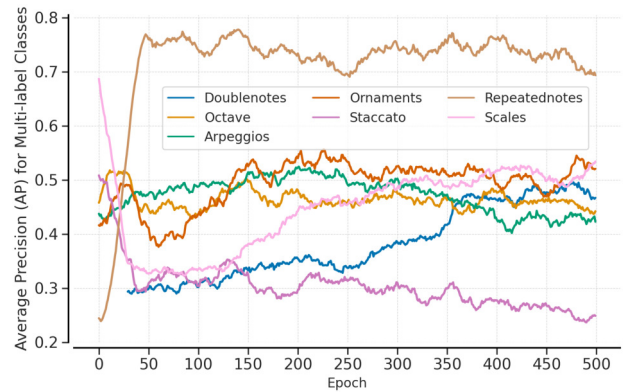


Figure 3. Average Precision for each class over epochs in multi-class prediction, from spectrogram representation.

6. CONCLUSION

Our research aimed to extend the capabilities of audio encoding models to the domain of solo piano performance understanding. Through this effort, we addressed tasks such as expertise ranking, difficulty estimation, and solo piano technique detection. The study introduced the Pianism-Labeling Dataset (PLD) and utilized a range of pre-trained audio encoders for evaluation. The curated set of performance-related attribute labels can contribute to multi-task learning or contrastive learning tasks in the future.

Our results, with the highest accuracy of 93.6% in expertise ranking, suggest that models like Audio-MAE hold promise for assessing aspects of musical performance, while the codified representations such as DAC or Jukebox struggle with capturing performance nuances. However, the studies on difficulty and especially techniques suggest the limitations of current pre-trained representations in capturing pianistic textures and patterns, as they fail to outperform the spectrogram baseline, prompting for the design of a performance-oriented audio representation. Meanwhile, the case study on the Chopin Piano Competition via transferring the assessment objective confirmed that we are still far from capturing the nuances of top-level human performance.

7. ACKNOWLEDGEMENT

This work is supported by the UKRI Centre for Doctoral Training in Artificial Intelligence and Music, funded by UK Research and Innovation [grant number EP/S022694/1], and the Engineering and Physical Sciences Research Council [grant number EP/T518086/1]. We are grateful for pianist Yan Zhou for helping with data verification.

8. REFERENCES

- [1] F. Korzeniowski and G. Widmer, “End-to-end musical key estimation using a convolutional neural network,” in *25th European Signal Processing Conference, EU-SIPCO*, 2017.
- [2] J. Pons, O. Nieto, M. Prockup, E. Schmidt, A. Ehmann, and X. Serra, “End-to-end learning for music audio tagging at scale,” *Proceedings of the 19th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 637–644, 2018.
- [3] H. Flores Garcia, A. Aguilar, E. Manilow, and B. Pardo, “Leveraging hierarchical structures for few-shot musical instrument recognition,” in *Proceedings of the 22th International Society for Music Information Retrieval Conference, ISMIR 2021*, 2021.
- [4] R. Castellon, C. Donahue, and P. Liang, “Codified audio language modeling learns useful representations for music information retrieval,” *Proceedings of the 22th International Society for Music Information Retrieval Conference, ISMIR 2021*, 2021.
- [5] M. C. McCallum, F. Korzeniowski, S. Oramas, F. Gouyon, and A. F. Ehmann, “Supervised and unsupervised learning of audio representations for music understanding,” in *Proceeding of the 21st International Society on Music Information Retrieval (ISMIR)*, 2022.
- [6] J. Gardner, S. Durand, D. Stoller, and R. M. Bittner, “LLark: A multimodal foundation model for music,” in *Proceedings of the International Conference on Machine Learning (ICML)*, 2024.
- [7] S. Doh, K. Choi, J. Lee, and J. Nam, “LP-MusicCaps: LLM-based pseudo music captioning,” in *Proceeding of the 24th International Society on Music Information Retrieval (ISMIR)*, 2023.
- [8] C. Palmer, “Music performance,” *Annual Review of Psychology*, vol. 48, 1997.
- [9] P. Ramoneda, N. Can Tamer, V. Eremenko, X. Serra, and M. Miron, “Score difficulty analysis for piano performance education based on fingering,” in *Proceeding of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.
- [10] P. Parmar, J. Reddy, and B. Morris, “Piano skills assessment,” in *IEEE 23th International Workshop on Multimedia Signal Processing (MMSP)*, 2021.
- [11] W. Wang, J. Pan, H. Yi, Z. Song, and M. Li, “Audio-based piano performance evaluation for beginners with convolutional neural network and attention mechanism,” *IEEE/ACM Transactions on Audio Speech and Language Processing*, vol. 29, pp. 1119–1133, 2021.
- [12] P. Seshadri and A. Lerch, “Improving music performance assessment with contrastive learning,” in *Proceedings of the 22nd International Society for Music Information Retrieval Conference (ISMIR)*, 2021.
- [13] J. Huang and A. Lerch, “Automatic assessment of sight-reading exercises,” in *Proceedings of the 20th International Society for Music Information Retrieval Conference (ISMIR)*, 2019.
- [14] A. Morsi, K. Tatsumi, A. Maezawa, T. Fujishima, and X. Serra, “Sounds Out of Pläce? Score-independent detection of conspicuous mistakes in piano performances,” in *Proceeding of the 24th International Society on Music Information Retrieval (ISMIR)*, 2023.
- [15] H. Kim, P. Ramoneda, M. Miron, and X. Serra, “An overview of automatic piano performance assessment within the music education context,” *International Conference on Computer Supported Education, CSEDU - Proceedings*, vol. 1, 2022.
- [16] A. Morsi, H. Zhang, A. Maezawa, S. Dixon, and X. Serra, “Simulating piano performance mistakes for music learning,” in *Proceedings of the Sound and Music Computing Conference (SMC)*, 2024.
- [17] A. Lerch, C. Arthur, A. Pati, and S. Gururani, “An interdisciplinary review of music performance analysis,” *Transactions of the International Society for Music Information Retrieval*, vol. 3, no. 1, pp. 221–245, 2020.
- [18] H. Zhang, E. Karystinaios, S. Dixon, G. Widmer, and C. E. Cancino-Chacón, “Symbolic music representations for classification tasks: A systematic evaluation,” in *Proceeding of the 24th International Society on Music Information Retrieval (ISMIR)*, Milan, Italy, 2023.
- [19] C. E. Cancino-Chacón, “Computational modeling of expressive music performance with linear and non-linear basis function models,” Ph.D. dissertation, Johannes Kepler University Linz, 2018.
- [20] C. Li, Z. Gan, Z. Yang, J. Yang, L. Li, L. Wang, and J. Gao, “Multimodal foundation models: From specialists to general-purpose assistants,” *arXiv preprint arXiv:2309.10020*, 2023.
- [21] M. Won, Y.-n. Hung, and D. Le, “A foundation model for music informatics,” in *2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024.
- [22] Y. Li, R. Yuan, G. Zhang, Y. Ma, X. Chen, H. Yin, C. Xiao, C. Lin, A. Ragni, E. Benetos, N. Gyenge, R. Dannenberg, R. Liu, W. Chen, G. Xia, Y. Shi,

- W. Huang, Z. Wang, Y. Guo, and J. Fu, "MERT: Acoustic music understanding model with large-scale self-supervised training," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024.
- [23] R. Kumar, P. Seetharaman, A. Luebs, I. Kumar, and K. Kumar, "High-fidelity audio compression with improved RVQGAN," in *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, 2023.
- [24] P.-y. H. Hu, X. Juncheng, C. Feichtenhofer, and M. Ai, "Masked autoencoders that listen," in *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, 2022.
- [25] K. Kosta, O. F. Bandtlow, and E. Chew, "Dynamics and relativity: Practical implications of dynamic markings in the score," *Journal of New Music Research*, vol. 47, no. 5, pp. 438–461, 2018.
- [26] H. Kim and X. Serra, "DiffVel : Note-level midi velocity estimation for piano performance by a double conditioned diffusion model," in *16th International Symposium on Computer Music Multidisciplinary Research (CMMR)*, Tokyo, Japan, 2023.
- [27] M. Grachten, W. Goebel, S. Flossmann, and G. Widmer, "Phase-plane representation and visualization of gestural structure in expressive timing," *Journal of New Music Research*, vol. 38, no. 2, pp. 183–195, Jun. 2009.
- [28] Z. Shi, "Computational analysis and modeling of expressive timing in chopin mazurkas," in *Proceedings of the 22nd International Society for Music Information Retrieval Conference (ISMIR)*, 2021.
- [29] S. R. M. Rafee, G. Fazekas, and G. A. Wiggins, "Performer identification from symbolic representation of music using statistical models," in *Proceedings of the International Computer Music Conference (ICMC)*, 2021.
- [30] Y. Zhao, C. Wang, G. Fazekas, E. Benetos, and M. Sandler, "Violinist identification based on vibrato features," in *European Signal Processing Conference*, 2021.
- [31] C. Sales, P. Wang, and Y. Jiang, "An interactive tool for exploring score-aligned performances: Opportunities for enhanced music engagement," *ACM International Conference Proceeding Series*, pp. 30–33, 2023.
- [32] H. Zhang, Y. Jiang, T. Jiang, and P. Hu, "Learn by referencing: Towards deep metric learning for singing assessment," in *Proceedings of the 22nd International Society for Music Information Retrieval Conference (ISMIR)*, 2021.
- [33] M. Matsubara, R. Kagawa, T. Hirano, and I. Tsuji, "CROCUS: Dataset of musical performance critiques," in *In Proceedings of the International Symposium on Computer Music Multidisciplinary Research (CMMR)*, 2021.
- [34] Y. Jiang, "Expert and novice evaluations of piano performances : Criteria for computer-aided feedback," in *Proceeding of the 24th International Society on Music Information Retrieval (ISMIR)*, 2023.
- [35] P. Ramoneda, D. Jeong, V. Eremenko, N. C. Tamer, M. Miron, and X. Serra, "Combining piano performance dimensions for score difficulty classification," *Expert Systems with Applications*, 2024.
- [36] C. Wang, V. Lostanlen, E. Benetos, and E. Chew, "Playing technique recognition by joint time frequency scattering," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.
- [37] N. Srivatsan and T. Berg-kirkpatrick, "Checklist models for improved output fluency in piano fingering prediction," in *Proceeding of the 23rd International Society on Music Information Retrieval (ISMIR)*, 2022.
- [38] C. V. Hall and J. T. O'Donnell, "Calibrating a bowing checker for violin students," *Journal of Music, Technology & Education*, vol. 3, no. 2-3, pp. 125–139, 2011.
- [39] P. Ramoneda, M. Lee, D. Jeong, J. J. Valero-Mas, and X. Serra, "Can audio reveal music performance difficulty? insights from the piano syllabus dataset," pp. 1–13, 2024.
- [40] Z. Borsos, R. Marinier, D. Vincent, E. Kharitonov, O. Pietquin, M. Sharifi, D. Roblek, O. Teboul, D. Grangier, M. Tagliasacchi, and N. Zeghidour, "Audiolm: A language modeling approach to audio generation," *IEEE/ACM Transactions on Audio Speech and Language Processing*, 2023.
- [41] A. Agostinelli, T. I. Denk, Z. Borsos, J. Engel, M. Verzetti, A. Caillon, Q. Huang, A. Jansen, A. Roberts, M. Tagliasacchi, M. Sharifi, N. Zeghidour, and C. Frank, "Musiclm: Generating music from text," *arXiv preprint arXiv:2301.11325*, 2023.
- [42] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, and I. Sutskever, "Jukebox: A generative model for music," *Computing Research Repository (CoRR)*, 2020.
- [43] E. Manilow, P. O'Reilly, P. Seetharaman, and B. Pardo, "Source separation by steering pretrained music models," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, vol. 2022-May, 2022, pp. 126–130.
- [44] S. Liu, A. S. Hussain, C. Sun, and Y. Shan, "Music understanding LLaMA: Advancing text-to-music generation with question answering and captioning," in *Proceeding of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024.
- [45] J. Liang, "Acoustic prompt tuning: Empowering large language models with audition capabilities," *arXiv preprint arXiv:2312.00249*, 2023.

- [46] H. F. Garcia, P. Seetharaman, R. Kumar, and B. Pardo, “VampNet: Music generation via masked acoustic token modeling,” in *Proceeding of the 24th International Society on Music Information Retrieval (ISMIR)*, 2023.
- [47] K. C. Puvvada, N. R. Koluguri, K. Dhawan, J. Balam, and B. Ginsburg, “Discrete audio representation as an alternative to mel-spectrograms for speaker and speech recognition,” in *Proceeding of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024.
- [48] C. Hawthorne, A. Stasyuk, A. Roberts, I. Simon, C. Z. A. Huang, S. Dieleman, E. Elsen, J. Engel, and D. Eck, “Enabling factorized piano music modeling and generation with the Maestro dataset,” in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019, pp. 1–12.
- [49] H. Zhang, J. Tang, S. Rafee, S. Dixon, and G. Fazekas, “ATEPP: A dataset of automatically transcribed expressive piano performance,” in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Bengaluru, India, 2022.
- [50] V. Konz, W. Bogler, and V. Arifi-M, “Saarland music data,” *Late-Breaking and Demo Session of the International Society on Music Information Retrieval (ISMIR)*, 2011.
- [51] G. Sandor, *On Piano Playing: Motion, Emotion and Expression*, 1981.
- [52] C.-J. Tsay, “Sight over sound in the judgment of music performance,” *Proceedings of the National Academy of Sciences*, 2013.
- [53] R. Bresin and G. Umberto Battel, “Articulation strategies in expressive piano performance analysis of legato, staccato, and repeated notes in performances of the Andante movement of Mozart’s sonata in g major (k.545),” *Journal of New Music Research*, vol. 29, no. 3, pp. 211–224, 2000.

TOWARDS EXPLAINABLE AND INTERPRETABLE MUSICAL DIFFICULTY ESTIMATION: A PARAMETER-EFFICIENT APPROACH

Pedro Ramoneda¹ Vsevolod Eremenko¹ Alexandre D’Hooge²
Emilia Parada-Cabaleiro³ Xavier Serra¹

¹ Music Technology Group, Universitat Pompeu Fabra, Barcelona, Spain

² Univ. Lille, CNRS, Centrale Lille, UMR 9189 CRIStAL, F-59000 Lille, France

³ Department of Music Pedagogy, Nuremberg University of Music, Germany

pedro.ramoneda@upf.edu

ABSTRACT

Estimating music piece difficulty is important for organizing educational music collections. This process could be partially automatized to facilitate the educator’s role. Nevertheless, the decisions performed by prevalent deep-learning models are hardly understandable, which may impair the acceptance of such a technology in music education curricula. Our work employs explainable descriptors for difficulty estimation in symbolic music representations. Furthermore, through a novel parameter-efficient white-box model, we outperform previous efforts while delivering interpretable results. These comprehensible outcomes emulate the functionality of a rubric, a tool widely used in music education. Our approach, evaluated in piano repertoire categorized in 9 classes, achieved 41.4% accuracy independently, with a mean squared error (MSE) of 1.7, showing precise difficulty estimation. Through our baseline, we illustrate how building on top of past research can offer alternatives for music difficulty assessment which are explainable and interpretable. With this, we aim to promote a more effective communication between the Music Information Retrieval (MIR) community and the music education one.

1. INTRODUCTION

Estimating the difficulty of music pieces aids in organizing large collections for music education purposes. However, manually assigning difficulty levels is laborious and might lead to subjective errors [1]. To address this, Music Information Retrieval (MIR) research has focused on automating this process for piano works represented in various modalities [2–6] as well as repertoires from other instruments [7, 8]. Furthermore, the interest of companies like Muse Group [9, 10] and Yousician [11] highlights the industry’s recognition of the importance of the task.

Previous work in this field has mainly focused on processing machine-readable symbolic scores [1–4, 12–15].

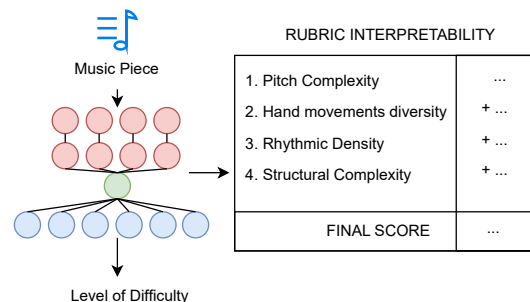



Figure 1: To promote a more objective and transparent assessment, in our white-box model *RubricNet*, similarly as educational rubrics, scores (here difficulty) are dependent on descriptors’ values. The Rubric Interpretability table displayed at the right is inspired by [17, Fig. 1]

These, unlike acoustic features extracted from audio whose understanding depends on signal processing knowledge, are both analyzable by computers and interpretable by humans. Musicians find also easier to understand symbolic features since based on music theory knowledge. Initial works towards interpretable difficulty assessment focused on visualization [12], with Chiu and Chen [13] making the first attempt to classify difficulty in the piano repertoire with explainable descriptors. Still, the continually increasing trend towards deep-learning based solutions [3,4], whose lack of transparency limits users’ understanding and therefore leads to an eventual non-acceptance in real life applications [16], can impair a fruitful implementation of such technologies in music educational practices.

With this background, we propose a white-box [18] model (cf. Figure 1), which through the concept of a rubric, i. e., an evaluation instrument from music education used to support objective assessment [19–22], allows a transparent interpretation of music difficulty. From this point forward, the white-box model will be denoted by *RubricNet*. Furthermore, to gain a profound understanding of what music difficulty means from an explainable perspective, we build upon the descriptors of Chiu and Chen [13] by proposing a new one focusing on music repetitive patterns. We also provide an interactive companion page¹ to visualize the evaluated data and scrutinize the results in light of its interpretability from a musical point of view.

Through eXplainable Artificial Intelligence (XAI), we

 © P. Ramoneda, V. Eremenko, A. D’Hooge, E. Parada-Cabaleiro, X. Serra. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** P. Ramoneda, V. Eremenko, A. D’Hooge, E. Parada-Cabaleiro, X. Serra. “Towards Explainable and Interpretable Musical Difficulty Estimation: A parameter-efficient approach”, in *Proc. of the 25th Int. Society for Music Information Retrieval Conf.*, San Francisco, USA, 2024.

¹ At: <https://pramoneda.github.io/rubricnet>

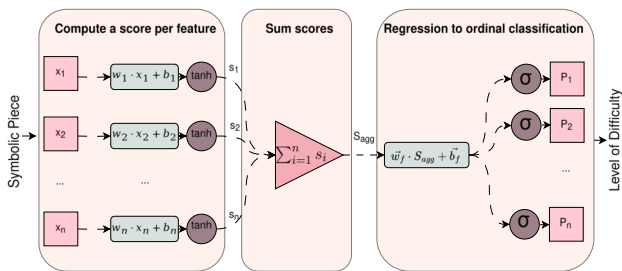


Figure 2: Detailed *RubricNet*'s architecture.

aim to contribute to music education by facilitating the understanding of measurable factors that determine a piece's difficulty. Our work builds on methods from music education, where objectively assessing abstract competences through measuring concrete criteria is consolidated by employing rubrics, i. e., tools which, unlike a black-box, break down complex concepts into simpler ones [19, 22].

Our interpretable methodology aims to bridge the gap between computational models and practical music education needs, enabling educators to make facilitated, but also informed decisions about curriculum development based on the difficulty levels of pieces. We release all the code and models of this research², in order to offer a baseline for further research in music difficulty assessment.

2. RELATED WORKS

Previous research aiming to automatically assess the difficulty of piano repertoire examined the link between fingering patterns and the pieces' difficulty level [2, 14, 15]. Recent studies [1, 3, 4, 23] have also made significant contributions. In [3], representations are used to feed three deep learning models—covering music notation, physical gestures, and expressiveness—to emulate Cook's dimensions [24]. These models' predictions are merged using an ensemble method to estimate the scores' difficulty. While we appreciate their musicology-inspired approach, its lack of interpretability harms its usability.

Difficulty estimation of piano pieces has also been investigated through hybrid methods that merge features with deep learning models [1, 23]. However, the absence of publicly shared data and code complicates performing comparative analyses with reference to these works. In [23], the authors combine the methods from Chiu and Chen [13] with deep learning models trained using piano roll as input. In a similar vein, [1] uses JSymbolic features [25] and deep learning models on a proprietary dataset.

In the study by Chiu and Chen [13], 159 pieces from the *8notes* website were used, whereas [23] utilized 1800 MIDI files from the same source. The categorization of these pieces, provided by users of *8notes*, raises concerns about their reliability. Unfortunately, neither study provides access to their data or details on how they were segmented. Other work [26] has attempted to understand the effectiveness of various features, including those proposed in [13] for categorizing the grade levels of a specific piano curriculum. Recent efforts by Zhang et al. [4] and

Ramonedá et al. [3] have focused on compiling datasets with difficulty annotations from the established piano publisher Henle Verlag, with the latter's dataset not only being the most extensive but also the only one made publicly available. Therefore, for our comparative analysis, we will use the open-source datasets presented in [3], namely *Can I Play It?* (CIPI), which has 9 levels of difficulty, and *Mikrokosmos-difficulty* (MKD), which includes 3 levels.

Finally, in order to validate our approach, we consider a different and established feature set, i.e., the standard music symbolic features available through Music21 library [27], which includes (amongst others) established JSymbolic features [25], thus facilitating a meaningful comparison with our proposed descriptors. In addition, we also contrast the results achieved with our novel descriptors with those obtained with the features by Chiu and Chen [13], which are also reimplemented and open-sourced in this study. Note that none of the approaches previously mentioned has focused on the interpretability of the descriptors, which is a key contribution of our work.

3. INTERPRETABLE *RubricNet*

The *RubricNet* model (cf. Figure 2) is designed to provide interpretability akin to a rubric, enabling its analysis and results to be intuitively aligned with established practices in music education. This approach ensures that the model's logic and outcomes are easily comprehensible, facilitating their usage in music education along to traditional tools.

3.1 Model Architecture

The network, comprising a series of linear layers dedicated to process individual input descriptors and followed by a nonlinear activation function, is formulated as follows:

Given a set of N input descriptors, each descriptor x_i is first processed through its dedicated linear layer with weight w_i and bias b_i , followed by a hyperbolic tangent activation function to yield:

$$s_i = \tanh(w_i \cdot x_i + b_i) \quad (1)$$

where s_i represents the processed score for the i -th descriptor. Scores are then aggregated in a single score S_{agg} :

$$S_{agg} = \sum_{i=1}^n s_i \quad (2)$$

The aggregated score S_{agg} is then passed through a final linear layer to obtain the logits for the class predictions, which are mapped to probabilities with a sigmoid function:

$$\vec{P} = \sigma(S_{agg} \cdot \vec{w}_f + \vec{b}_f) \quad (3)$$

where σ denotes the sigmoid function, \vec{w}_f and \vec{b}_f are the weight and bias of the final linear layer, respectively.

3.2 Ordinal Optimization

This model applies an ordinal optimization approach [28], predicting ordered categorical outcomes, i. e., difficulty

²At: <https://github.com/pramonedá/rubricnet>

Descriptor	Explanation
Pitch Entropy	Indicates pitch variety; higher values mean more diverse pitch collection
Pitch Range	Distance between the lowest and highest notes.
Average Pitch Displacement	Indicating the central pitch level.
Rate	Measures hand movement intensity across keys reflecting physicality in performance.
Average IOI	The average timing between note onsets, indicative of rhythmic density.
Pitch Set LZ	Indicative of structural complexity and repetitiveness within a pitch set sequence.

Table 1: Explanation of descriptors in musical terms.

levels such as beginner (1), intermediate (2), and advanced (3), through logits. These logits, computed using a mean squared error (MSE) loss, indicate the model’s predictions on the ordinal scale. Difficulty level is then obtained as:

$$\max\{i \text{ where } P_i \geq 0.5 \text{ and } P_j \geq 0.5, \forall j < i\} \quad (4)$$

3.3 Interpretability

In *RubricNet*, the *descriptors* (automatically computed from the data) are, to some extent, comparable to the formalized *evaluation criteria* defined in traditional rubrics; similarly, the *aggregated score*, might be comparable to a final *grade/mark* assigned in an educational scenario. Given the correspondences between both, we could consider the model a “white-box” approach, able to promote transparency and interpretability, similarly to a rubric.

It uses independent linear transformations on input descriptors to generate scores between -1 and 1, which directly influence the regression output, S_{agg} . Since negative scores, might be not fully understood in terms of difficulty level, we normalize scores between 0 and 1, rescaling S_{agg} between 0 and 12. This approach mirrors rubric’s ability to provide objective and structured feedback, with the simplicity of these transformations aiding in understanding the impact of features in predictions.

The interpretability of the model lies in its ability to dissect each descriptors’ influence on a piece’s difficulty level. Consequently, analyzing each descriptor’s scores might reveal its overall importance on the prediction. Lastly, S_{agg} is a continuous-ordered scalar with rank correlation to difficulty. Therefore, from S_{agg} , we retrieve ordered and discrete categories with clear decision boundaries.

4. EXPLAINABLE DESCRIPTORS

From codified musical scores, we extracted numeric features which are feed to a classification algorithm. We re-implemented a set of features from the literature [13] while proposing a novel one, Pitch Set LZ. In addition to explaining the features (cf. Table 1), we will provide their technical descriptions and analyze their relevance to difficulty and interdependencies using the data.

4.1 Descriptors

In our work, we analyze music sheets encoded in symbolic format, focusing on extracting pitch and timing. Following the approach suggested by Chiu and Chen [13], we

process left and right hand parts separately to clarify pedagogical aspects of musical difficulty. Our primary analysis involves sequences of pitch set events, each characterized by a pitch set S and onset time T . Pitch sets, represented by sets of MIDI numbers, are defined over the alphabet of all pitch sets \mathbf{S} that occurred in a score part, while onset times are calculated in seconds from the performance start by the music21 library [27] with reference to marked tempo information. This method emphasizes the timing of note attacks, duration and rests. Additionally, we consider a collection of pitch events, each defined by pitch P over the alphabet of all pitches \mathbf{P} . Our analysis started with the five features identified by Chiu and Chen [13] as most relevant to understanding musical difficulty.

Pitch Entropy. The entropy of pitches in the pitch events:

$$-\sum_{i \in \mathbf{P}} p(P = i) \log_2 p(P = i) \quad (5)$$

Pitch Range. The distance between the minimum and maximum MIDI pitches in a score part.

Average Pitch. The average MIDI pitch in a music sheet.

Displacement Rate. Initially proposed by [13], it quantifies the extent of hand movement across the keyboard during the performance of a score. It analyzes maximum pitch distances between consecutive pitch set events and is calculated as a weighted average of three categories: distances less than 7 semitones (assigned a weight of zero); distances over 7 semitones but under an octave (assigned a weight of one); and distances of an octave or larger (assigned a weight of two to emphasize larger movements).

Average IOI: Average Inter Onset Interval. A concept similar to the “Playing speed” introduced by [13], a term we consider deceptive since it actually decreases as the hand’s “speed” increases. This is an average time in seconds between onsets of two consecutive pitch set events. Let’s denote i^{th} onset time with T_i , then the value is:

$$\frac{\sum_{1 \leq i \leq N_{events}-1} (T_{i+1} - T_i)}{N_{events} - 1} \quad (6)$$

In 23% of the scores, information about the recommended performance tempo is missing. We then assume the tempo is 100 beats per minute (bpm). Thus, in cases of missing bpm, the Average IOI feature might not be relevant.

Pitch Set LZ. Lempel-Ziv complexity of pitch set sequence. Before introducing our proposed descriptor, it is crucial to provide context and motivation. Pitch Entropy, as emphasized by Chiu and Chen [13], is particularly relevant—a conclusion supported by the analysis of correlations between difficulty and features in the following section, as well as by informal experiments. As Sayood discusses [29], there’s a link between entropy of a task and the cognitive load it imposes on the performer, a concept that may also apply to music performance [30]. However, music is often perceived in terms of larger structures like phrases and sections, not just isolated pitches, prompting us to seek a descriptor that captures the “repetitiveness”

Feature	τ_c
Pitch Entropy (R)	0.583
Pitch Set LZ (L)	0.583
Pitch Entropy (L)	0.582
Pitch Set LZ (R)	0.573
Pitch Range (L)	0.567
Pitch Range (R)	0.554
Displacement Rate (R)	0.332
Displacement Rate (L)	0.273
Average IOI (R)	-0.209
Average IOI (L)	-0.208
Average Pitch (R)	0.088
Average Pitch (L)	0.017

Table 2: Features ordered by absolute values of their τ_c rank correlation with the difficulty level.

of music on a broader scale. To this end, we employ LZ-complexity, a measure of redundancy introduced by Lempel and Ziv [31]. In context of music research, it was used for binary encoded rhythm analysis by Shmulevich and Povel [32]. We apply LZ-complexity to sequence of pitch sets: scan a score part, identify all subsequences of pitch sets that cannot be reproduced from preceding material through a recursive copying procedure. The number of such unique subsequences is defined as the LZ-complexity of the part. This approach allows us to assess the structural complexity and redundancy of a musical piece, highlighting the cognitive demands placed on performers.

4.2 Feature Analysis

We assume that, for easier interpretability, features must on average change monotonically with the difficulty level. To measure this quality, we use the τ_c version of Kendall rank correlation coefficient due to its ability to deal with “heavily tied” rankings [33] (many musical pieces have the same difficulty, hence, we have multiple ties in the ranking by difficulty). τ_c is equal to 1 when feature and difficulty rankings are perfectly aligned in the same direction, -1 if they are aligned in opposite directions. As the number of nonconcordant cases increases, the coefficient approaches zero. In Table 2, the results show that the features related to pitch organization are the most correlated to difficulty. Hand displacement and Inter-onset intervals are less correlated, while average pitch seems almost irrelevant.

In addition, we aim to uncover dependencies among the features themselves while mitigating the influence of difficulty, with whom most features are correlated. To achieve this, we calculate conditional τ_c correlations for all feature pairs given a fixed difficulty level, and average the coefficients across all difficulty levels. We then convert these coefficients into a distance matrix and apply hierarchical agglomerative clustering based on average distance to identify clusters of correlated features. From the resulting dendrogram (cf. Figure 3), we observe that features correlated with difficulty—namely Pitch Entropy, Pitch Set LZ, and Pitch Range—are also interrelated. This is remarkable because the three most correlated features are not inherently dependent: one could envision a music piece with any of them maximized while maintaining low values for the others. However, pieces in CIPI typically exhibit coordinated

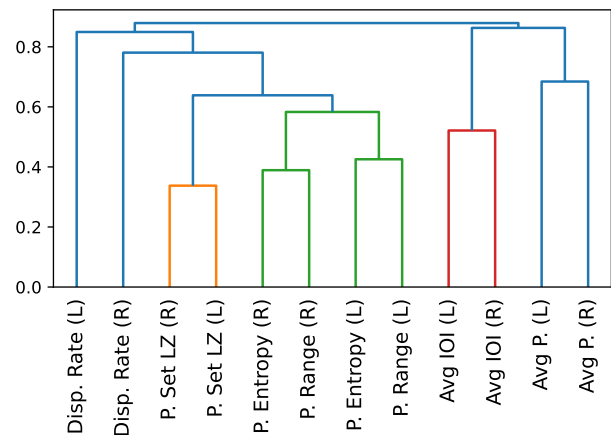


Figure 3: Hierarchical clustering of features based on their average correlation distance within each difficulty class.

values in these descriptors. Thus, we mostly observe the combined effect of these features, making it challenging to reliably decompose “difficulty” into an aggregate of independent components.

5. EXPERIMENTS

5.1 Experimental Setup

To evaluate the effectiveness of our proposed method, we utilized the *Mikrokosmos-difficulty* (MKD), and *Can I play it?* (CIPI) datasets [3]. For fair comparison, we use the 5-fold cross-validation approach defined in [3]. In each split, 60% of the data is used as a train set, while the remaining is equally divided into validation and test sets.

As in [3], we employ mean squared error (MSE) and accuracy within n classes ($\text{Acc-}n$) for evaluation. These metrics are chosen for their applicability to ordinal classification challenges, with $\text{Acc-}n$ assessing the model’s accuracy for n classes from the true labels, and MSE measuring the average squared prediction error across classes. The effects of dataset imbalances and a fair evaluation across classes are mitigated by macro-averaged metrics.

We optimize the models during training through Adam optimizer with a learning rate of 10^{-2} . The training process incorporates early stopping, based on the $\text{Acc-}n$ and MSE metrics from the validation set, to prevent overfitting. Through Ordinal Loss, we frame difficulty prediction as an ordinal classification task, as mentioned in Section 3. We apply a standard scaler and dropout to the features to prevent individual ones from dominating. For each experiment, we look for the best hyperparameters using Bayesian optimization [34]: batch size within the range from 16 to 128, dropout rate between 0.1 and 0.5, learning rate decay from 0.1 to 0.9, and the learning rate itself, tested over a logarithmic scale from $1e-5$ to $1e-1$. This approach allows us to systematically explore the hyperparameter space and identify the optimal settings for our models; thus, enabling a fair comparison between experiments.

5.2 Experimental Results

In Table 3, the results from the comparison between the performance of our novel approach with the presented de-

	CIPI		MKD
	Acc-9	MSE	Acc-3
argnn [3]	32.6(2.8)	2.1(0.2)	75.3(6.1)
virtuoso [3]	35.2(7.3)	2.1(0.2)	65.7(7.8)
pitch [3]	32.2(5.9)	1.9(0.2)	74.2(9.2)
ensemble [3]	39.5(3.4)	1.1(0.2)	76.4(2.3)
Ours	41.4(3.1)	1.7(0.5)	79.6(8.8)

Table 3: Experiment comparison of previous individual deep learning models [3], their ensemble and our explainable and interpretable method on CIPI and MKD.

Experiment	Acc-9	MSE
<i>RubricNet</i> proposed	41.4(3.1)	1.7(0.5)
"" with Chiu and Chen [13] descriptors	36.2(5.2)	1.7(0.3)
"" with Music21 descriptors	36.7(6.0)	1.3(0.2)
"" with ALL descriptors	38.9(4.3)	1.3(0.1)
"" proposed without Avg P.	39.0(5.6)	1.5(0.4)
"" with positive scores	38.5(3.5)	1.6(0.6)
"" without ordinal regression	36.2(1.3)	2.1(0.4)
Logistic regression	40.0(4.3)	1.5(0.3)

Table 4: Ablation study results for different feature sets (5 first rows) and model configurations (last 3 rows) on CIPI.

scriptors (cf. Sections 3 and 4) and the results achieved by three previous models from the literature (argnn [35], virtuoso [36], pitch) as well as their collective ensemble, are shown. Our model achieves the highest Acc-9 score of 41.4(±3.1) in CIPI, surpassing the ensemble’s 39.5(±3.4), while displaying the second lower MSE of 1.7(±0.5), only overtaken by the ensemble’s 1.1(±0.2). With an Acc-3 score of 79.6(±8.8) in the MKD dataset, our approach is superior to previous ones but with a higher standard deviation.

In the following, we examine the impact of various feature and model configurations on *RubricNet* performance (cf. Table 4). As baseline for comparison, we consider the configuration previously discussed (cf. Ours in Table 3).

Employing only the five Chiu and Chen [13] descriptors, i.e., excluding Pitch Set LZ, leads to a decrease in Acc-9 by −5.2, reflecting a performance drop from the baseline. The use of Music21 [27] descriptors, which include JSymbolic [25] and other descriptors widely used in the community, results in a decrease in Acc-9 by −4.7 and a decrease in MSE by −0.4, showing slight improvements in MSE but not in accuracy. However, note that a larger number of descriptors could decrease the explainability. Combining all the descriptors slightly decreases Acc-9 and MSE: −2.5 and −0.4, respectively; with the accuracy results still under the baseline. These results indicate that the descriptors discussed in Section 4 constitute the best option for difficulty estimation on CIPI. Since average pitch showed no relation to difficulty in the feature analysis, we repeated the experiments without this feature. This lead, however, to non-significant worsening of the results.

Concerning the impact of different model configurations, we replace the tanh by sigmoid non-linearities to guarantee positive scores. The obtained MSE rate is similar but the accuracy drops by −3.1. This means that negative scores could aid in training, which is why we keep them, but normalize the scores after the training phase. Besides, substituting the ordinal encoding used in the base-

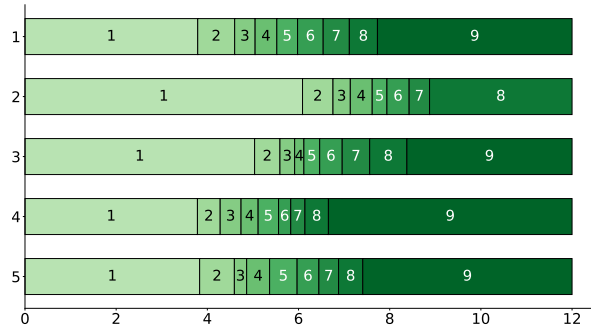


Figure 4: Decision boundaries of the model between grades on S_{agg} (X axis) for all splits (Y axis) on CIPI.

line with a traditional one-hot encoding with cross-entropy loss, results in a decrease in Acc-9 by −5.2 and an increase in MSE by +0.4, highlighting the importance of ordinal regression in achieving lower MSE rates. Lastly, logistic regression with ordinal loss decreases the Acc-9 by −1.4 while showing a decrease of MSE by −0.2. This offers a compromise for both metrics but without beating our setup and to our understanding, being less interpretable.

Overall, the gains offered by *RubricNet* with the features proposed are relatively modest compared to the baselines. However, having a smaller feature set is necessary for explainability. The novelty of our approach lies in aligning the interpretability of music education with rubric-like interpretability feedback. This alignment is essential for a successful application of our model in practice, as we will discuss in further sections.

5.3 Decision Boundaries

In *RubricNet*, the input features are combined into a single scalar before performing the final ordinal classification. Analysis of the results shows that the final layer defines optimized decision boundaries, setting thresholds for S_{agg} that progressively increase along with difficulty levels. Because of the final sigmoid activation, once S_{agg} exceeds a boundary, the corresponding difficulty level will always be active, which guarantees the ordinality of the predictions. By examining the decision boundaries (cf. Figure 4), we observe that the trends are similar across splits, displaying shorter valid ranges around intermediate levels. Note that in split 2, there are only 8 classes because the model ignored the last class. This can happen as we use numeric optimization, which sometimes falls into local minima. These minima might seem optimal based on the validation metrics but do not meet our overall performance expectations.

6. DISCUSSION AND LIMITATIONS

Now, we analyze whether *RubricNet* is interpretable from a musical point of view. To understand how features impact the final level suggested by the model, we evaluate the contribution of each descriptor to the aggregated score. Since learning to play an instrument is a progressive process, relative contributions of features to different levels

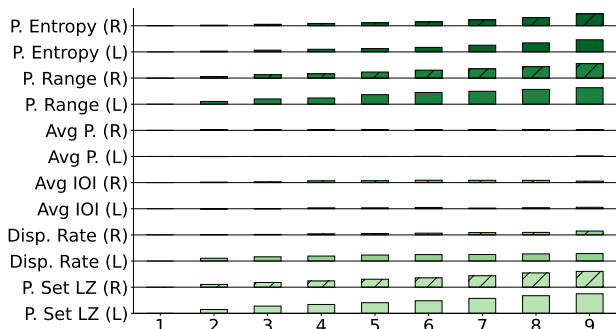


Figure 5: Average relative contribution of descriptors (Y axis) normalized between 0 and 1, across grades (X axis).

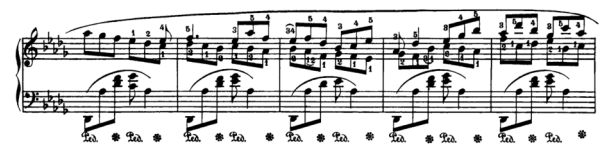
Descriptor Name	Score	Grade Divergence	Accumulative Score
Pitch Set Lz	+ 0.08	- 0.23	0.08
Pitch Range	+ 0.42	- 0.37	1.04
Average Pitch	+ 0.07	- 0.04	1.21
Average Ioi Seconds	+ 0.09	- 0.04	1.38
Displacement Rate	+ 0.63	- 0.25	2.85
Pitch Entropy	+ 0.4	- 0.2	3.82
...
Final Score	—	—	3.9

Figure 6: Simplified difficulty interpretable rubric for the *Nocturne op. 9, no. 3* (F. Chopin). Descriptors' values for the right hand and final score (for both hands) are shown.

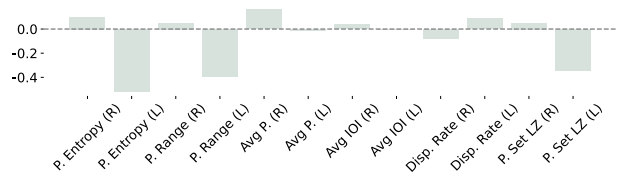
with reference to the grade 1 are displayed instead of absolute values. These contributions are averaged across splits on the test set and shown in Figure 5.

We observe a trend of higher contributions when the level increases for every descriptor. This observation is consistent with the fact that S_{agg} value increases for higher levels (cf. Figure 4). The most discriminative features are pitch entropy and pitch range, as well as the LZ descriptor for higher levels. Conversely, some features, e.g., average IOI or the average pitch, have low contribution to the model's decisions, as shown by their relatively constant and small values across grades. The latter is expected, since very different pieces could have the same average pitch, not disclosing anything about difficulty. The former might be explained by the averaging, which can remove information, especially when a piece can alternate between fast and slow parts. Besides, as mentioned before, tempo is often poorly annotated in the dataset.

To better understand the explainable capabilities of the proposed descriptors, in the following, we provide a musical examination of two concrete samples, by this demonstrating the interpretability of our approach. *Nocturne op. 9, no. 3* by F. Chopin, is labelled as level 7, but classified as level 2. All the descriptors are below the grade average, as shown in the rubric (cf. *Grade Divergence* in Figure 6). Our hypothesis is that this nocturne contains many challenges that go beyond the descriptors used. There are constant changes in dynamics, a variety of articulations, and as a key difficulty aspect, many types of polyrhythms between the right and left hands. Further research should address all the types of difficulty challenges, probably underrepresented in the existing datasets.



(a) *Berceuse in D-flat major, Op.57* (F. Chopin). Bars 6-10.



(b) Distance to the average for the grade of the scores. Extracted from original rubric (grade divergence column).

Figure 7: Musical excerpt (a) and a rubric outcome (grade divergence) plotted (b) from a piece in level 7.

The piece *Berceuse in D-flat major, Op.57* by F. Chopin, shown in Figure 7 is appropriately classified as grade 7. This is because it maintains a left-hand accompaniment with few changes, in contrast to the higher virtuosity of the right hand. The left hand has scores below average in most descriptors because of its few changes: Pitch Range (-0.41), Average IOI (-0.04), Pitch Set LZ (-0.34), Average Pitch (-0.01), and Pitch Entropy (-0.52). In contrast, the right hand shows more virtuosity, with higher than average scores for all the right hand features. These scores collectively contribute to a final cumulative score that accurately reflects the overall difficulty.

Finally, it should be noted that our approach primarily focuses on descriptors related to pitch sequences and onsets, while disregarding others. Still, the ablation study showed that other features sets (e.g., those from music21), even covering aspects like rhythm variety, do not enhance our classifier's performance either. In addition, expressive elements [37] such as dynamics, tempo changes, and articulation, since often left to performers' interpretation, are not always captured in musical notation [3], and therefore is a dimension our score-based model does not consider.

7. CONCLUSION AND FUTURE WORK

In our study, we proposed a novel white-box parameter-efficient model aligned with the music education community tools, i.e., rubrics, which outperforms previous approaches on difficulty estimation. In addition, we created an interactive companion page for visualizing CIPI and MKD datasets. In summary, we showed that analyzing explainable descriptors, unlike deep learning models, offers clarity, which gives both teachers and students specific insights into pieces. This approach not only underscores the importance of explainable artificial intelligence (XAI) in understanding music difficulty, but also emphasizes the potential for such technologies to contribute to the broader field of music education. For future research, we consider interesting to creating a dataset based on technical challenges like finger fluency and polyphonic complexity, as well as user studies for understanding the perception of interpretable feedback by music education community.

8. ETHICS STATEMENT

The system presented in this paper aims at obtaining the difficulty of a musical piece through several descriptors. In previous work, descriptors were not available, limiting access to the area. This situation underscores the need for open science practices. Therefore, we open our implementation, to facilitate access for new researchers. Besides, the dataset used for this study is available upon request for non-profit and academic research purposes. While this limits its use in commercial applications, it ensures the reproducibility of the results. The data consists of open-source scores of music that is no longer copyrighted, its use for open research can thus be considered fair.

The proposed work belongs to the area of assisted music learning. One might argue that such a tool can have a detrimental impact on music teaching jobs. While this is a valid concern, we think that an eventual solution of the addressed task, would not endanger music educators profession, whose role naturally goes much beyond than categorizing music in difficulty levels. Instead, this technology should be seen as a way to support them in the own teaching practices, for instance, by alleviating their burden on some duties, such as exploring large collections, and by this enabling them to easily discover forgotten musical works from our cultural heritage which fit students' needs. Moreover, through this research, we also aim to convey the message that the path to advancement does not solely lie in acquiring more data or creating larger models. By highlighting what drives its decisions, our proposed model aligns with the goals of eXplainable AI, something crucial for its acceptance in music education. Although our efforts in making the system interpretable and explainable will partly answer the common criticisms made to black-box approaches, the real impact of our system remains to be verified by its future use in real scenarios.

9. ACKNOWLEDGEMENTS

We want to thank Alia Morsi's previous work on difficulty estimation from a feature engineering perspective, which encouraged research in tabular data [26]. The team would also acknowledge Marius Miron for continuously insisting on aligning difficulty estimation with explainability. He also highlighted the direction of using a rubric-like explainability feedback, pointing out Ustun et al.'s work [17]. We also thank Roser Batlle-Roca for helping to discuss some concepts between interpretability and explainability based on her research [38].

This work was supported by "IA y Música: Cátedra en Inteligencia Artificial y Música" (TSI-100929-2023-1), funded by the Secretaría de Estado de Digitalización e Inteligencia Artificial, and the European Union-Next Generation EU, under the program "Cátedras ENIA 2022 para la creación de cátedras universidad-empresa en IA". This work was also supported by the ANR project TABASCO (ANR-22-CE38-0001) and the travel grant MERMOZ2-012047. Finally, this work was also possible through the support of the Hightech Agenda Bayern, funded by the Free State of Bavaria (Germany).

10. REFERENCES

- [1] D. S. Deconto, E. L. F. Valenga, and C. N. Silla, "Automatic music score difficulty classification," in *Proc. of the 30th IEEE Int. Conf. on Systems, Signals and Image Processing (IWSSIP)*, Ohrid, North Macedonia, 2023.
- [2] P. Ramoneda, N. C. Tamer, V. Eremenko, M. Miron, and X. Serra, "Score difficulty analysis for piano performance education," in *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Singapore, Singapore, 2022.
- [3] P. Ramoneda, D. Jeong, V. Eremenko, N. C. Tamer, M. Miron, and X. Serra, "Combining piano performance dimensions for score difficulty classification," *Expert Systems with Applications*, vol. 238, pp. 1–16, 2024.
- [4] H. Zhang, E. Karystinaios, S. Dixon, G. Widmer, and C. E. Cancino-Chacón, "Symbolic music representations for classification tasks: A systematic evaluation," in *Proc. of the 24th Int. Society for Music Information Retrieval Conf. (ISMIR)*, Milan, Italy, 2023.
- [5] P. Ramoneda, D. Jeong, J. J. Valero-Mas, and X. Serra, "Predicting performance difficulty from piano sheet music images," in *Proc. of the 19th Int. Society for Music Information Retrieval Conf. (ISMIR)*, Milano, Italy, 2023.
- [6] P. Ramoneda, M. Lee, D. Jeong, J. J. Valero-Mas, and X. Serra, "Can audio reveal music performance difficulty? insights from the piano syllabus dataset," *arXiv preprint arXiv:2403.03947*, 2024.
- [7] M. A. V. Vásquez, M. Baelemans, J. Driedger, W. Zuidema, and J. A. Burgoyne, "Quantifying the ease of playing song chords on the guitar," in *Proc. of the 24th Int. Society for Music Information Retrieval Conf. (ISMIR)*, Milan, Italy, 2023.
- [8] E. Holder, E. Tilevich, and A. Gillick, "Musiplectics: Computational assessment of the complexity of music scores," in *Proc. of the ACM Int. Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software (Onward!)*, Pittsburgh, USA, 2015.
- [9] "Musescore have automatic difficulty categories from year 2022," <https://musescore.com/>, accessed on April 12, 2024.
- [10] "Ultimate guitar have automatic difficulty categories from year 2022," <https://www.ultimate-guitar.com/>, accessed on April 12, 2024.
- [11] "System for estimating user's skill in playing a music instrument and determining virtual exercises thereof," Patent US9 767 705B1, 2017.
- [12] V. Sébastien, H. Ralambondrainy, O. Sébastien, and N. Conruyt, "Score analyzer: Automatically determining scores difficulty level for instrumental e-learning,"

- in *Proc. of the 13th Int. Society for Music Information Retrieval Conf. (ISMIR)*, Porto, Portugal, 2012.
- [13] S.-C. Chiu and M.-S. Chen, “A study on difficulty level recognition of piano sheet music,” in *Proc. of the IEEE Int. Symposium on Multimedia (ISM)*, Irvin, USA, 2012.
- [14] E. Nakamura, N. Ono, and S. Sagayama, “Merged-output hmm for piano fingering of both hands,” in *Proc. of the 15th Int. Society for Music Information Retrieval Conf. (ISMIR)*, Taipei, Taiwan, 2014.
- [15] E. Nakamura and S. Sagayama, “Automatic piano reduction from ensemble scores based on merged-output hidden markov model,” in *Proc. of the 41st Int. Computer Music Conf. (ICMC)*, Denton, USA, 2015.
- [16] D. Branley-Bell, R. Whitworth, and L. Coventry, “User trust and understanding of explainable ai: Exploring algorithm visualisations and user biases,” in *Proc. of the Int. Conf. on Human-Computer Interaction (HCI)*, Copenhagen, Denmark, 2020.
- [17] B. Ustun and C. Rudin, “Learning Optimized Risk Scores,” *Journal of Machine Learning Research*, vol. 20, no. 150, pp. 1–75, 2019.
- [18] O. Loyola-Gonzalez, “Black-box vs. white-box: Understanding their advantages and weaknesses from a practical point of view,” *IEEE access*, vol. 7, pp. 154 096–154 113, 2019.
- [19] M. E. Latimer, M. J. Bergee, and M. L. Cohen, “Reliability and perceived pedagogical utility of a weighted music performance assessment rubric,” *Journal of Research in Music Education*, vol. 58, pp. 168 – 183, 2010.
- [20] M. Álvarez-Díaz, L. M. Muñoz-Bascón, A. Soria-Alemany, A. Veintimilla-Bonet, and R. Fernández-Alonso, “On the design and validation of a rubric for the evaluation of performance in a musical contest,” *International Journal of Music Education*, vol. 39, pp. 66 – 79, 2020.
- [21] B. C. Wesolowski, “Understanding and developing rubrics for music performance assessment,” *Music Educators Journal*, vol. 98, pp. 36 – 42, 2012.
- [22] A. Jonsson and G. Svingby, “The use of scoring rubrics: Reliability, validity, and educational consequences,” *Educational Research Review*, vol. 2, pp. 130–144, 2007.
- [23] Y. Ghatas, M. Fayek, and M. Hadhoud, “A hybrid deep learning approach for musical difficulty estimation of piano symbolic music,” *Alexandria Engineering Journal*, vol. 61, no. 12, pp. 1–14, 2022.
- [24] N. Cook, “Analysing performance and performing analysis,” *Rethinking Music*, vol. 8, pp. 1–23, 1999.
- [25] I. F. McKay, Julie E. Cumming, “jSymbolic 2.2: Extracting features from symbolic music for use in musicological and MIR research.” in *Proc. of the 19th Int. Society for Music Information Retrieval Conf. (ISMIR)*, Paris, France, 2018.
- [26] A. Morsi, “Characterizing difficulty levels of keyboard music scores,” Master’s thesis, Music Technology Group, Universitat Pompeu Fabra, 2020. [Online]. Available: <https://doi.org/10.5281/zenodo.4090526>
- [27] M. S. Cuthbert and C. Ariza, “Music21: A toolkit for computer-aided musicology and symbolic music data,” in *Proc. of the 11th Int. Society for Music Information Retrieval Conf. (ISMIR)*, Utrecht, Netherlands, 2010.
- [28] J. Cheng, Z. Wang, and G. Pollastri, “A neural network approach to ordinal regression,” in *Proc. of the IEEE Int. Joint Conf. on Neural Networks (IJCNN)*, Hong Kong, China, 2008.
- [29] K. Sayood, “Information theory and cognition: A review,” *Entropy*, vol. 20, pp. 1–19, 2018.
- [30] C. Palmer, “The nature of memory for music performance skills,” in *Music, Motor Control and the Brain*, E. Altenmüller, J. Kesselring, and M. Wiesendanger, Eds. Oxford, UK: Oxford University Press, 2012.
- [31] J. Ziv and A. Lempel, “Compression of individual sequences via variable-rate coding,” *IEEE Transactions on Information Theory*, vol. 24, no. 5, pp. 530–536, 1978.
- [32] I. Shmulevich and D.-J. Povel, “Measures of temporal pattern complexity,” *Journal of New Music Research*, vol. 29, no. 1, pp. 61–69, 2000.
- [33] M. Kendall, *Rank Correlation Methods*, ser. Griffin books on statistics. Griffin, 1962.
- [34] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, “Optuna: A next-generation hyperparameter optimization framework,” in *Proc. of the 25th ACM SIGKDD Int. Conf. on Knowledge Discovery & Data mining (KDD)*, Anchorage, USA, 2019.
- [35] P. Ramoneda, D. Jeong, E. Nakamura, X. Serra, and M. Miron, “Automatic piano fingering from partially annotated scores using autoregressive neural networks,” in *Proceedings of the 30th ACM International Conference on Multimedia (MM ’22)*, Lisboa, Portugal, 2022.
- [36] D. Jeong, T. Kwon, Y. Kim, K. Lee, and J. Nam, “VirtuosoNet: A Hierarchical RNN-based System for Modeling Expressive Piano Performance,” in *Proceedings of the 20th International Society for Music Information Retrieval Conference (ISMIR)*, Delft, The Netherlands, 2019.

- [37] H. Zhang and S. Dixon, "Disentangling the horowitz factor: Learning content and style from expressive piano performance," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.
- [38] R. Batlle-Roca, E. Gómez, W. Liao, X. Serra, and Y. Mitsufuji, "Transparency in music-generative ai: A systematic literature review," 2023. [Online]. Available: <http://dx.doi.org/10.21203/rs.3.rs-3708077/v1>

PURPOSEFUL PLAY: EVALUATION AND CO-DESIGN OF CASUAL MUSIC CREATION APPLICATIONS WITH CHILDREN

Michele Newman¹ Lidia Morris¹ Jun Kato²
Masataka Goto² Jason Yip¹ Jin Ha Lee¹

¹ Information School, University of Washington, United States

² National Institute of Advanced Industrial Science and Technology (AIST), Japan

mmn13@uw.edu, ljmorris@uw.edu, jun.kato@aist.go.jp,

m.goto@aist.go.jp, jcyip@uw.edu, jinhalee@uw.edu

ABSTRACT

The rise of digital technologies has increased interest in democratizing music creation, but current creativity support tools often prioritize literacy and education over meeting children’s needs for casual creation. To address this, we conducted Participatory Design sessions with children aged 6-13 to explore their perceptions of casual music creation activities and identify elements of creative applications that support different expressions. Our study aimed to answer two key questions: (1) How do children perceive casual music creation activities and which elements of creative applications facilitate expression? and (2) What insights can inform the design of future casual music creation tools? Our findings indicate that children view casual music creation as involving diverse activities, with visuals aiding in understanding sounds, and engaging in various playful interactions leading to creative experiences. We present design implications based on our findings and introduce casual creation as "purposeful play". Furthermore, we discuss its implications for creative MIR.

1. INTRODUCTION

Digital technologies have sparked interest in democratizing creation as they enable diverse individuals to produce cultural objects [1–3], suggesting we may understand these tools as enhancers of human creativity [4–7]. For example, over the past two decades, there has been a rise in the development and study of Creativity Support Tools (CSTs) in the field of Human-Computer Interaction (HCI) [8]. Despite the abundance of music-related CSTs, including tools such as digital audio workstations [9–11], notation software [12, 13], style-specific composition/identification tools [14, 15], and music generation systems [16, 17], many fail to cater to children.

Children’s creative experiences are often shaped by a limited understanding of social norms [18], implying that systems designed for adults may not fully support their creative endeavors. Moreover, many music applications designed for youth primarily focus on literacy or are deployed in formal education contexts [19, 20]. However, previous work highlights the value of informal and casual music experiences in education [21, 22]. Building on this work, we explore the potential of casual music experiences for children, focusing on casual music creation. We define casual music creation as *creative musical experiences prioritizing the process of enjoyment over product outcome*, drawing inspiration from Compton’s research on casual creation systems [23, 24]. While casual musical experiences relate to informal learning [22], this study focuses on how music technology as part of CSTs supports these experiences, providing new ways for children’s self-expression rather than skill development. Research shows that supporting creativity is vital for children, as it helps to foster children’s identities [25, 26], develop confidence in their creative abilities [27, 28], as well as support brain development [29, 30] and social skills [31].

However, there is a gap in understanding children’s MIR needs. This is particularly true in creative MIR, or the use of retrieved music information for creative purposes [7, 32], despite growing interest in creative applications [7, 33]. For instance, only two ISMIR papers address children: one develops the *Children’s Song Dataset* for song synthesis [34], and the other involves children in designing a music organization app [35]. While these studies provide insights to children’s experiences with MIR tasks, there is still a broader question about how children interact with MIR tasks to meet their unique creative needs. Building off this prior research in MIR, music education, and HCI, we utilized a method of Participatory Design (PD) called Cooperative Inquiry, a type of PD that focuses on designing technology *with* and *for* children [36]. As children are a growing user group of creative technologies, PD can generate developmentally appropriate design ideas and feedback [37, 38], boosting children’s self-esteem through facilitating design in a casual setting [39]. We examine children’s creative needs while using musical CSTs within creative MIR contexts through two PD sessions with children aged 6-13, addressing two questions: (1) How do chil-



© M. Newman, L. Morris, J. Kato, M. Goto, J. Yip, and J.H. Lee. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** M. Newman, L. Morris, J. Kato, M. Goto, J. Yip, and J.H. Lee, "Purposeful Play: Evaluation and Co-Design of Casual Music Creation Applications with Children", in *Proc. of the 25th Int. Society for Music Information Retrieval Conf.*, San Francisco, United States, 2024.

dren perceive casual music creation activities and which elements of creative applications facilitate expression? and (2) What insights can inform the design of future casual music creation tools? This paper contributes to the democratization of music creation by addressing children’s unique creative needs in casual music application design. Furthermore, we present a set of design principles to support more playful interactions with music and discuss their implications for future work in creative MIR.

2. RELATED WORK

2.1 Children’s Musical Creativity

Creativity is the ability to generate original and valuable ideas [4]. In the realm of music, this translates to realizing such ideas through composition, analysis, or performance [40]. Scholars argue that musical creativity is embodied, meaning environmental factors play a role in shaping creative cognition in music [40–43]. Furthermore, music educator Peter Webster has suggested that musical creativity is more akin to creative processes, or what he terms as moving from a musical idea to a product [44].

For children, there has been a particular focus on understanding their creativity in reference to composition [45–47] in music education contexts. Yet, musical experiences begin early in childhood and are increasingly impacted by popular music experiences with new forms of technology [48–51]. Therefore, children experience music via playful interactions in a variety of modalities [52]. Notably, social-emotional environments, especially those shaped by parents and teachers, can serve as catalysts for children’s musical creativity through play [53–55], suggesting that children’s musical experiences are impacted by their development, environment, and interactions with technology. While play and technology are crucial in music education and cognitive development, questions remain about how CSTs can enhance casual music experiences and whether these interactions impact children’s creativity.

2.2 Creativity Support Tools

Creativity Support Tools (CSTs) are digital resources designed to enhance creativity [8]. Interactive musical systems (IMSs) have shown promise in supporting non-musicians’ engagement in music making [56], but domain expertise can influence creativity [40, 57]. Hence, specialized tools have been developed to meet novices’ needs, often incorporating critique [58], such as those for novice filmmakers [59] or digital painting systems [60]. Recognizing the importance of personally meaningful creative activities, referred to as "mini-c" creativity, there is a growing emphasis on integrating this perspective into CST evaluation methods [61, 62]. This acknowledgment underscores the significance of understanding children’s creative experiences with musical CSTs and adapting design and evaluation approaches accordingly [63]. While technology’s impact on children’s creativity has been explored in areas like storytelling and video creation [64, 65], its effects on musical creativity remain relatively unexplored

Pseudonym	Age	Gender	Ethnicity	Sessions
Annie	6	Female	Latino	DS2
Emma	9	Female	Black / White	DS1, DS2
Han	10	Male	Latino	DS2
Jayden	9	Male	Asian / Black	DS2
Keon	9	Male	Asian / Black	DS1, DS2
Liam	9	Male	Asian / White	DS1
Jin	13	Female	Asian / White	DS1, DS2
Taylor	10	Female	Asian / White	DS1, DS2
Seiko	10	Female	Asian	DS1
Zachary	8	Male	Asian / White	DS1, DS2

Table 1. Demographics of Child Participants

[50, 58, 66]. Though tools aiding children in music composition exist, they are often designed for general novices [63], even though previous work has suggested specific design recommendations for other musical acts by children, such as composition at home [45]. Previous research recognizes children’s unique creative needs and the potential of CSTs to foster creativity and learning. However, there is still uncertainty about the differences between children’s structured and casual creation with CSTs, and how these differences relate to creative MIR.

3. METHODS

3.1 Participatory Design

For this study, we utilized Cooperative Inquiry (CI) [36, 67], a Participatory Design (PD) method facilitating collaboration between designers and users, thus democratizing the design process. CI specifically emphasizes allowing children and adults to design as equals. This method offers insights into children’s learning [67], empowering them to articulate thoughts on complex issues such as family finances [68], gender [69], and creativity [70, 71].

3.2 Participants

The **KidsTeam UW** co-design group comprises adult design researchers (investigators, master’s students, and undergraduate students) and 10 child participants, using pseudonyms for confidentiality (see Table 1). Children were recruited through mailing lists and snowball sampling with parental consent and child assent obtained. The research received approval from the university’s Institutional Review Board. Two 90-minute design sessions were held in January and February 2024, with five to eight adult facilitators serving as design partners in each session.

3.3 Design Sessions

Our design sessions started with a 15-minute *Snack Time* for socializing, followed by a 15-minute *Circle Time* featuring a "Question of the Day" to warm up for the design activity. Then, participants engaged in small group design activities for 45 minutes in *Design Time*, followed by a 15-minute *Full Group Discussion* for presentations and reflection.

Child	App Name	Description
Annie	Color Block	Users compose by dragging and dropping colored blocks. Users can download the music to share.
Emma	Cat Choir	An app where users may drag different clothing representing different sounds onto cats to compose songs.
Han	Mixtape	Users create and share "mixtapes" by pulling music from streaming services and creating playlists. Also allows for composing with provided sounds and AI.
Jayden	Untitled	Users organize their music and can search, filter, and create albums. They can also remix other songs.
Keon	Untitled	An app to store music files, allowing users to drag and drop music files from other apps.
Jin	Dreamer	A music composition app that acts as a game where users are able to manipulate different environments to create music for a story.
Taylor	Sing-a-Song	Users can create songs by dragging instruments onto tracks and export them with a video or animated characters. Others can remix these songs.
Zachary	Piano God	An app meant to help pianist practice songs using animations to tell users which keys to play.

Table 2. Descriptions of Applications designed by children in DS2

3.3.1 Design Session 1: Playing with Casual Music CSTs

Design Session 1 (DS1) took place in January 2024. We asked the children to play with four different casual music tools to elicit their feedback on different types of casual music applications. The first is *TextAlive* [72, 73], a website that automatically synchronizes lyrics text with music, detects timing information of beats and other musical elements, and allows users to interactively create “lyric videos” – music videos in which lyrics animate in sync with the music. The second tool, *TextAlive Flow* [74] (available on tablet and desktop), is an extension of *TextAlive* that has a more casual user interface. It allows users to touch the screen to change the video’s visuals (typography, colors, motion patterns, etc.) while listening to the music. *Incredibox* [75] lets users create songs by dragging and dropping outfits onto animated characters, combining pre-recorded beatbox sounds and melodies. Lastly, *Sketch-a-Song* [76] is a tablet application that lets users tap and drag to add different pitches and sounds. These tools were selected to allow children to engage with various modes of interacting and making with music. During the session, we captured what the children liked, disliked, and design ideas for each app on a sticky note, organizing them into thematic groups on a whiteboard [36].

3.3.2 Design Session 2: Designing Casual Music CSTs

Design Session 2 (DS2) took place in February 2024. We asked the children to “design a casual music creation app.” We asked them to define what their app allowed them to do with music, and develop a user flow including how they moved between the homepage, creation interface, and to sharing their creations with others. We derived these design aspects from the themes that arose during DS1. Before breaking into our design groups, we shared an example of what a user flow looked like using *TextAlive Flow*. We supplied the children with a large bag with different craft materials and paper, asking them to engage in low-fidelity prototyping of their application [77].

3.4 Data Collection and Analysis

Our hybrid design sessions utilized Zoom for video and screen recordings across three computers for each design group. We recorded a total of 6 hours and 10 minutes of video. Researchers also documented creative artifacts

with a camera and took notes on a legal pad. Children’s thoughts were summarized during group discussions and collected on a *Google Slides* deck.

We utilized an inductive qualitative approach for data analysis [78]. The initial codebook was developed by the first author through inductively coding recorded session videos. Codes like “*Musical Activities – Remixing*” and “*Control – Variety of Options*” were included in the first iteration. Subsequently, two authors conducted consensus coding [79] on design artifacts, researcher memos, and session videos, adjusting the codebook as needed. In cases of disagreement, a third team member resolved discrepancies. This process led to the final version of the codebook. For example, we applied the code “*App Elements – Control*” to the quote “when you could see or hear a difference, it makes you feel like you’re in more control.” Further descriptions of design artifacts and applications of our codes can be found in our supplemental material.¹

4. FINDINGS

4.1 Children’s Perceptions of Casual Music Creation

Consistent with previous work [48,52], children saw music making as a holistic, process-focused experience [44], and expected to engage in multiple musical activities within a single app. Composing was the most referenced activity, as all DS2 apps except Keon’s, which stored music files, involved music composition. Listening to music was also prominent (Han, Jayden, Jin, & Taylor, DS2). Remixing, proposed only in Taylor’s and Jayden’s apps, was the least suggested activity.

Children’s views on the applications were shaped by their past experiences with music and technology, indicating their preferences often reflect experiences with other applications such as music streaming apps [48], as well as their cultural backgrounds and existing knowledge [80]. For example, many applications from DS2, shown in Table 2, also included references to other applications. Emma and Taylor’s applications referenced *Incredibox* and *Sketch-a-Song* respectively, imitating the drag and drop features for layering musical sounds. The interface of Zachary’s application was similar to the application *Syn-*

¹ Our supplemental material can be found at: <https://doi.org/10.17605/OSF.IO/5DNS6>.

thesia, with falling blocks that demonstrated which keys to play on a piano keyboard. Additionally, there were various ways suggested to supplement listening methods that were similar to other applications such as organizing playlists (Han & Jayden, DS2) or watching music videos (Jin & Taylor, DS2). Additionally, some of the children referenced their previous music education experiences. In our study, Jin, who has taken piano lessons, found *Sketch-a-Song* limiting due to its representation of musical pitches stating it felt “pedantic” (DS1). Zachary’s app included a piano in the interface, including letters for the different keys. Similarly, Annie included *solfège* (i.e., do, re, mi) as the notes for her app “Color Block.”

While the children’s interactions initially focused on the process of exploring with music, they mentioned the importance of these customization options to give them a sense of control as they created. This was especially important as the children formed creative products. As an illustration, reflecting on *TextAlive* and *Sketch-a-Song*, Taylor noted “You can’t create your own song [in the apps], you’re just designing it, and even then, you don’t have much control over it” (DS1).

4.2 Visuals as a Bridge to Music

Within our sessions, we found a connection between visuals and sounds, with children noting that the aesthetics of an application changed the way the music was perceived. For example, Jin stated: “I liked changing the colors because even if you’re given this format [in *TextAlive Flow*], since colors have a strong effect on how music is portrayed, you can change the whole vibe, even if you are restricted” (DS1). Other children noted changing colors to fit their experience was important demonstrated by three distinct sticky notes expressing appreciation for “many options for colors,” “all the color options,” and “cool color range” when discussing customization in *TextAlive*. During DS1, an adult co-designer also noted that children also became visibly excited when able to use animations in using *TextAlive* and *TextAlive Flow*, as evidenced by a sticky that read the options for animation in *TextAlive Flow* “are cool” and another that they liked the “active lyrics” as they moved across the screen (DS1 – Sticky Note²). Animations also were added into some of the children’s apps, such as Zachary, who had boxes that represented musical notes “fly down” from the top of the screen.

Furthermore, the children in our study showed a propensity toward characters and narrative to support their experiences with music. Children noted the reason they enjoyed *Incredibox* was because they “liked [the] *incredibox* character’s designs” (DS1 – Sticky Note). Yet, other children highlighted with their dislike of the “bad outfits,” (DS1 – Sticky Note), suggesting they would like other options that suited their ideas. Children extended the idea of characters into their own apps, such as Taylor and Emma who included an option to have animated characters sing or

perform the song the user created (DS2). During the full group discussion in DS1, Jin summed up the importance of the visuals noting, “when you could see or hear a difference, it makes you feel like you’re in more control of what is going on.”

Our analysis further suggests that the visual aspects of an app act as a bridge to better understand musical possibilities. One sticky note from DS1 captured that the kids disliked that the “*MVs [referencing the animation of the characters in Incredibox] don’t seem to match/vibe [of the music] naturally*” with the sounds and that it was distracting that the people were “not wearing clothes before you dress them” (DS1). In DS2, children also considered colors and aesthetics in their own designs. The sounds in Emma’s remixed version of *Incredibox*, “Cat Choir,” were all related to cats and cat activities (e.g., scratching, meowing, and purring), to match the sounds to the visuals of cats. Similarly, Annie used different colors to represent different pitches. Jin created the app “Dreamer,” a game that lets users play through a young girl’s dreams. Each dream had its own visual aesthetic or “vibe,” corresponding with the sounds and instruments, such as clam shells as percussion instruments in an “aquatic dream” as seen in Figure 1.

4.3 Interface Preferences and User Interactions

Children also showed a preference for direct interactions, preferring the ability to manipulate elements through touch, drag, and drop actions. For example, Taylor noted that she “like[ed] touching [the iPad] instead of the mouse because the mouse was harder to use” (DS1). Similarly, many of the children’s designs in DS2 also included the ability to drag and drop elements, such as the outfits that could be dragged onto cats in Emma’s application or the color blocks that could be dragged in Annie’s app.

When the gulf of execution [81] in the interfaces was large (i.e., the interfaces do not afford what happens when manipulated), children became frustrated. For example, some disliked the “confusing” parameter tuning interface of *TextAlive* that appears next to the video, sometimes forcing them to tweak parameters indirectly. They favored the more direct control of *TextAlive Flow* instead, which allows them to touch the video and change the parameters with their hands. They recommended design enhancements to improve interface usability, citing dislikes such as the absence of instrument labels and clarity on color-to-instrument/note mapping in *Sketch-a-Song* (DS1 - Sticky Note). In contrast, Emma noted a preference for *Incredibox* because it was “less frustrating” and “ignited creativ-

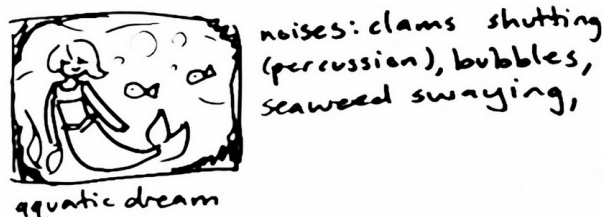


Figure 1. Jin’s app “Dreamer” from DS2

² We use “DS1–Sticky Note” to refer to Likes/Dislikes/Design Ideas captured on stickies during the design activity that were not attributed to a specific child, but instead to the design group.

ity" by representing sounds with symbols rather than traditional instruments (DS1). This suggests that achieving a balance between visual interface design and a certain level of ambiguity is crucial to foster casual creative experiences.

4.4 Sharing and Casual Creative Experiences

In our study, participants highlighted the importance of sharing their creations and ensuring the quality of the final product. For example, children suggested that a sharing option should be added to *Sketch-a-Song* (DS1 – Sticky Note), a feature present in the application, though many of the children were unable to find it. During DS1, another adult facilitator noted some children physically left their groups to share their creations with friends, explaining interesting interactions and experiences documenting their creative choices while using the apps. Taylor and Seiko requested time to share their song from *Sketch-a-Song* with the entire group, indicating pride in their work (DS1), and noted that they felt that it sounded good enough to share. In DS2, Emma, Han, Taylor, and Jin incorporated features allowing users to share and listen to others' creations or playlists in their apps as well. Jin and Taylor's apps even enabled users to create songs with accompanying videos inspired by others (DS2). This suggests that though children were exploring, they also wanted the output of these casual systems to sound good enough to share with others.

5. DISCUSSION

5.1 Purposeful Play: From Process "to" Outcome

Children in our study attempted to balance exploring the possibilities provided by the application with creating personal intermediate outcomes to help express themselves, seeing casual interactions with music as a form of *purposeful play*. We suggest that casual creation is better understood as a "process to outcome" rather than "process over outcome," as we initially stated. In this view, children see themselves as designers of creative works, with the play experience focused on expression.

In our findings, we observed that children wanted the ability to have some sense of control over their experiences (4.1) but that these came via scaffolds such as the visuals of the application (4.2) and interactions with the application (4.3) that lead to shareable outcomes (4.4). This conception of supported play aligns with previous work within music education [22] as well as MIR. Cunningham and Zhang, who conducted PD sessions with children to create a music organizer called *Kids Music Box* suggested the final design of their application offered a "*playground*" for interaction while listening to songs [35, p.190]. Similarly, *PlaceAndPlay*, an application design for creating and recording music, focused on children's ability to simply try things out, with their results noting "*all children had a great time when allowed to just play with the system*" [82, p.738]. Facilitating children's enjoyment and understanding of musical involvement entail not only promoting play

but also nurturing their comprehension of cultural contexts [18] and social conventions [25]. More broadly, play can be understood in relation to creative processes [83–85], as many of the cognitive and emotional functions linked to creativity are also evident during play [86].

Importantly, children also expressed a desire for their final product to be share-worthy (4.4), indicating an expectation that their experience would yield a creative product representing their musical experience and tastes (4.1). We suggest that what sets casual CSTs apart from other educational technologies are the creation of "intermediate products." The term "intermediate" can be understood as creative products that move users from what Beghetto and Kaufman suggest are "*intrapersonal creativity that is part of the learning process*" [61], to products recognized by others as creative. This concept of creativity is increasingly integrated into the evaluation of CSTs [62]. Furthermore, our findings underscore the importance of ensuring that casual CSTs for children focus on helping users create intermediate creative outcomes that remain coherent and aesthetically pleasing to support users' creative self-efficacy [28].

5.2 The Purposeful Play Design Toolbox

In this section, we introduce four design principles, deemed "tools," to foster the elements that lead to purposeful play, suggesting specific design features for each.

5.2.1 Controlled Serendipity

Previous work in creative MIR has shown that serendipity is a crucial aspect for supporting meaningful interactions with music information during the creative process [32]. This was an important element in supporting non-musicians in musical creativity [56]. When surprised by an app, children in our study felt excited and inspired, like Emma's excitement with *Incredibox* (4.3). However, they also wanted their creations to feel genuinely theirs, i.e., their individual exploration mattered (4.4). Therefore, casual music tools should offer structured control while guiding users towards aesthetically pleasing results that reflect their goals. Novices often do not have the domain knowledge to identify how to execute specific creative goals or whether those goals are domain relevant [87, 88]. Therefore, a system taking on the role of the guiding professional by supplying options that support a pleasing final product, may help children to feel excited about their creative outputs. For example, both *Incredibox* and *Sketch-a-Song* only supplied notes that corresponded to a specific chord progression, and as a result, any "seemingly" random combination of sound layers or feature options also sounded good to the children. Similarly, *TextAlive* and *TextAlive Flow* supplied templates or color combinations that looked aesthetically pleasing and matched the music. The carefully and intentionally constrained environment was able to provide the sense of serendipity but at the same time produce outcomes that children felt good about and wanted to show off.

Design Features. Implementing structured guidelines alongside controlled randomness provides a framework for fostering creativity. Feedback mechanisms that allow transparency serve to facilitate children in revisiting and elaborating upon moments of unexpected discovery.

5.2.2 Visual Scaffolds

The term scaffold can be understood as the use of a temporary framework for supporting learners as they aim to gain new skills [89]. During our analysis, children expressed consideration of the role of an application’s visuals when creating (4.2). In a sense, musical experiences were “scaffolded” by the visual aesthetics of the application, since the intention of the musical technology is to encourage children to develop an aesthetic perspective [90] through clear and direct visual communication of the application’s possibilities for creation. When visuals do not align with the sounds, or at least align in a way that a child expected, such as when the animations in *Incredibox* did not align with the music, it can be distracting and take away from understanding of the music, even if the UI design is clear. Yet, as Emma noted (4.3), some ambiguity in the visual scaffolding can also spark creative experiences as well. Specifically, our results emphasize that *color* and *characters* are two visual scaffolds that are effective for children. Furthermore, our findings suggest that children perceive casual creation as encompassing multiple mediums, often utilizing sound and video, aligning with the conception of children’s musical experiences being multimodal [52]. Prior research advocates for multimedia authoring activities that enable collaborative reflection among children [91, 92], promoting self-expression [93] at both individual and social levels.

Design Features. Visual elements like real-time visualizations, character-based imagery, customizable aesthetics, and visual ambiguity, when integrated into features designed to evoke serendipitous moments, along with multimodal outputs like videos, can act as scaffolds to support children’s musical interest.

5.2.3 Direct Manipulation

Children in our study preferred the ability to directly interact with the interface, which can be understood as a form of direct manipulation [94]. Shneiderman, who suggested the term, notes four features of user interfaces that utilize this concept: continuous representation of the object of interest, physical actions, immediate feedback, and the ability of novices to gain knowledge of the system quickly. Moreover, helping kids manipulate things effectively means showing clear connections within the subject area, which helps them link new skills with what they already know [95]. Furthermore, computer scientist Alan Kay [96] suggests that visuals play an important role in digital spaces—they offer representational systems that through manipulation lead to chains of abstract reasoning that creates *symbols*; in the semiotic terms, these symbols allow a user to externalize through the manipulation of representations [97]. This suggests a connection between the

visual scaffolds and potential direct interactions that lead to moments of play in digital creative systems.

Design Features. Interactive elements such as drag-and-drop functionalities, objects responding to user actions, and tactile interactions, will enhance children’s engagement and maintain their interest over time.

5.2.4 Shareable Intermediate Outputs

The children in our study wanted the ability to share the creative outputs they were proud of during their exploration of different tools (4.4). Allowing children to share their creative outputs can help build creative self-efficacy [28], which is essential to fostering their view of themselves as creators. Allowing children to share these objects encourages creativity at not only an individual level, but also a social level, which is particularly important as social-environmental factors have been shown to influence creativity of individuals [98]. This is particularly important for children as creativity is largely social for them [53, 55].

Design Features. Sharing options (email, file downloads, replay), galleries of user-generated content, ability to remix or elaborate on others outputs will help support self-efficacy of children as developing creators.

6. LIMITATIONS AND FUTURE WORK

While our research follows established precedents, it has limitations. The small sample size of 10 children, while comparable to similar co-design studies [68, 69], may limit the generalizability of the result. Participants were mainly from a single geographic area, with privileged backgrounds, and familiar with technology and co-design, which may not represent diverse socio-economic perspectives. Future studies should include more diverse demographics, explore evaluation methods for supporting design principles, and investigate features tailored to different MIR tasks in support of purposeful play.

7. CONCLUSION

Our study explored the creative preferences of one user group, children, in casual music creation applications. Through two Participatory Design sessions, we observed children’s perceptions of casual musical creation as a personally-oriented process, where visuals and direct interactions allowed children to generate creative works they wished to share with others. We highlighted the importance of purpose in play during casual music creation, suggesting that casual creation applications should facilitate the process of exploration of music with the intention of expression. Additionally, we discussed the potential impact of this playful approach on creative MIR by presenting four design tools to support purposeful play and suggesting a set of design features that support these principles. We further believe that these insights transcend children, offering design implications for individuals of various musical skills and recreational adults who wish to explore musical experiences in a variety of ways.

8. ACKNOWLEDGMENTS

We thank the child participants and adult co-designers for their contributions to the study, without whom this work would not be possible. This material is based upon work supported by the Institute of Museum and Library Services under Award #LG-252291-OLS-22. This work was supported in part by JST CREST Grant Number JP-MJCR20D4, Japan.

9. REFERENCES

- [1] H. Jenkins, *Convergence Culture : Where Old and New Media Collide*. New York University Press, 2006.
- [2] T. J. Tanenbaum, A. M. Williams, A. Desjardins, and K. Tanenbaum, “Democratizing technology: pleasure, utility and expressiveness in diy and maker practice,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI ’13. New York, NY, USA: Association for Computing Machinery, 2013, p. 2603–2612. [Online]. Available: <https://doi.org/10.1145/2470654.2481360>
- [3] M. Resnick, M. Flanagan, C. Kelleher, M. MacLaurin, Y. Ohshima, K. Perlin, and R. Torres, “Growing up programming: democratizing the creation of dynamic, interactive media,” in *CHI ’09 Extended Abstracts on Human Factors in Computing Systems*, ser. CHI EA ’09. New York, NY, USA: Association for Computing Machinery, 2009, p. 3293–3296. [Online]. Available: <https://doi.org/10.1145/1520340.1520472>
- [4] M. A. Boden, *The Creative Mind: Myths and Mechanisms*, 2nd ed. Routledge, 2004.
- [5] J. Koch, J. Pearson, A. Lucero, M. Sturdee, W. E. Mackay, M. Lewis, and S. Robinson, “Where art meets technology: Integrating tangible and intelligent tools in creative processes,” in *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, ser. CHI EA ’20. New York, NY, USA: Association for Computing Machinery, 2020, p. 1–7. [Online]. Available: <https://doi.org/10.1145/3334480.3375172>
- [6] L. Manovich, *Software Takes Command: Extending the Language of New Media*, ser. International Texts in Critical Media Aesthetics. Bloomsbury, 2013.
- [7] E. J. Humphrey, D. Turnbull, and T. Collins, “A brief review of creative mir,” *ISMIR Late-Breaking News and Demos*, 2013.
- [8] J. Frich, L. MacDonald Vermeulen, C. Remy, M. M. Biskjaer, and P. Dalsgaard, “Mapping the Landscape of Creativity Support Tools in HCI,” in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, ser. CHI ’19. Association for Computing Machinery, 2019, pp. 1–18. [Online]. Available: <https://dl.acm.org/doi/10.1145/3290605.3300619>
- [9] Apple Inc., “Logic pro,” v. 10.8.1. [Online]. Available: <https://www.apple.com/logic-pro/>
- [10] Ableton, “Ableton live,” v. 10.8.112. [Online]. Available: <https://www.ableton.com/en/live/>
- [11] Adobe, “Audition,” v. 24.0. [Online]. Available: <https://www.adobe.com/products/audition.html>
- [12] MakeMusic, Inc., “Finale,” v. 27.4. [Online]. Available: <https://www.finalemusic.com/>
- [13] Avid, “Sibelius,” v. 2024.3. [Online]. Available: <https://www.avid.com/campaigns/musical-notation-software>
- [14] C.-H. Chuan and E. Chew, “Quantifying the benefits of using an interactive decision support tool for creating musical accompaniment in a particular style.” in *ISMIR*, 2010, pp. 471–476.
- [15] M. Alinoori and V. Tzerpos, “Music-star: a style translation system for audio-based re-instrumentation.” in *Proceedings of the 23rd Int. Society for Music Information Retrieval Conf*, 2022, pp. 419–426.
- [16] J. Ens and P. Pasquier, “Flexible generation with the multi-track music machine.” in *21st International Society for Music Information Retrieval Conference (ISMIR)*, 2020.
- [17] Z. Wang and G. Xia, “Musebert: Pre-training music representation for music understanding and controllable generation.” in *Proceedings of the 22nd Int. Society for Music Information Retrieval Conf*, 2021, pp. 722–729.
- [18] V. T. Kudryavtsev, “The phenomenon of child creativity,” *International Journal of Early Years Education*, vol. 19, no. 1, pp. 45–53, 2011. [Online]. Available: <http://offcampus.lib.washington.edu/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=eue&AN=62823302&site=ehost-live>
- [19] M. P. Downton, “The aesthetics, creativity and craftsmanship of fourth graders’ compositions,” *Journal of Music, Technology & Education*, vol. 8, no. 3, pp. 273–286, 2015.
- [20] J. Garcia, T. Tsandilas, C. Agon, and W. E. Mackay, “Structured observation with polyphony: a multi-faceted tool for studying music composition,” in *Proceedings of the 2014 conference on designing interactive systems*, 2014, pp. 199–208.
- [21] M. Callanan, C. Cervantes, and M. Loomis, “Informal learning,” *Wiley Interdisciplinary Reviews: Cognitive Science*, vol. 2, no. 6, pp. 646–655, 2011.
- [22] E. Harwood and K. Marsh, “3.2 Children’s Ways of Learning Inside and Outside the Classroom,” in *The Oxford Handbook of Music Education, Volume 1*. Oxford University Press, 09 2012. [Online]. Available: https://doi.org/10.1093/oxfordhb/9780199730810.013.0020_update_001

- [23] K. Compton, “Casual creators: Defining a genre of autotelic creativity support systems,” Ph.D. dissertation, University of California, Santa Cruz, 2019.
- [24] K. Compton and M. Mateas, “Casual creators.” in *ICCC*, 2015, pp. 228–235.
- [25] M. A. Runco and N. Cayirag, “The Development of Children’s Creativity,” in *Handbook of Research on the Education of Young Children*, O. N. Saracho and B. Spodek, Eds. Taylor & Francis Group, 2012, pp. 102–114. [Online]. Available: <http://ebookcentral.proquest.com/lib/washington/detail.action?docID=1114640>
- [26] S. C. Hurwitz, “To be successful—let them play!(for parents particularly),” *Childhood Education*, vol. 79, no. 2, pp. 101–103, 2002.
- [27] M. Helfand, J. C. Kaufman, and R. A. Beghetto, “The four-C model of creativity: Culture and context,” in *The Palgrave Handbook of Creativity and Culture Research*, V. P. Glăveanu, Ed. Palgrave Macmillan UK, 2016, pp. 15–36. [Online]. Available: https://doi.org/10.1057/978-1-137-46344-9_2
- [28] P. Tierney and S. M. Farmer, “Creative Self-Efficacy: Its Potential Antecedents and Relationship to Creative Performance,” *The Academy of Management Journal*, vol. 45, no. 6, pp. 1137–1148, 2002. [Online]. Available: <https://www.jstor.org/stable/3069429>
- [29] J. L. Frost, “Neuroscience, play, and child development.” 1998. [Online]. Available: <https://eric.ed.gov/?id=ED427845>
- [30] C. S. Tamis-LeMonda, J. D. Shannon, N. J. Cabrera, and M. E. Lamb, “Fathers and mothers at play with their 2-and 3-year-olds: Contributions to language and cognitive development,” *Child development*, vol. 75, no. 6, pp. 1806–1820, 2004.
- [31] R. J. Erickson, “Play contributes to the full emotional development of the child.” *Education*, vol. 105, no. 3, 1985.
- [32] H. K. G. Andersen and P. Knees, “Conversations with expert users in music retrieval and research challenges for creative MIR.” in *17th International Society for Music Information Retrieval Conference (ISMIR)*, 2016.
- [33] C. Bauer, “Report on the ISMIR 2020 special session: How do we help artists?” in *ACM SIGIR Forum*, vol. 54, no. 2. ACM New York, NY, USA, 2021, pp. 1–7.
- [34] S. Choi, W. Kim, S. Park, S. Yong, and J. Nam, “Children’s song dataset for singing voice research,” in *International Society for Music Information Retrieval Conference (ISMIR)*, 2020.
- [35] S. J. Cunningham and E. Zhang, “Development of a music organizer for children.” in *ISMIR*, 2008, pp. 185–190.
- [36] M. L. Guha, A. Druin, and J. A. Fails, “Cooperative Inquiry revisited: Reflections of the past and guidelines for the future of intergenerational co-design,” *International Journal of Child-Computer Interaction*, vol. 1, no. 1, pp. 14–23, 2013. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2212868912000049>
- [37] J.-S. Baek and K.-P. Lee, “A participatory design approach to information architecture design for children,” *Co-Design*, vol. 4, no. 3, pp. 173–191, 2008.
- [38] M. Coenraad, J. Palmer, D. Franklin, and D. Weintrop, “Enacting identities: Participatory design as a context for youth to reflect, project, and apply their emerging identities,” in *Proceedings of the 18th ACM International Conference on Interaction Design and Children*, ser. IDC ’19. New York, NY, USA: Association for Computing Machinery, 2019, p. 185–196. [Online]. Available: <https://doi.org/10.1145/3311927.3323148>
- [39] S. Schepers, K. Dreessen, and B. Zaman, “Fun as a user gain in participatory design processes involving children: a case study,” in *Proceedings of the 17th ACM Conference on Interaction Design and Children*, ser. IDC ’18. New York, NY, USA: Association for Computing Machinery, 2018, p. 396–404. [Online]. Available: <https://doi.org/10.1145/3202185.3202763>
- [40] A. Schiavio and M. Benedek, “Dimensions of Musical Creativity,” *Frontiers in Neuroscience*, vol. 14, 2020. [Online]. Available: <https://www.frontiersin.org/journals/neuroscience/articles/10.3389/fnins.2020.578932>
- [41] D. van der Schyff, A. Schiavio, A. Walton, V. Velardo, and A. Chemero, “Musical creativity and the embodied mind: Exploring the possibilities of 4e cognition and dynamical systems theory,” *Music & Science*, vol. 1, p. 2059204318792319, 2018.
- [42] A. Cox, *Music & Embodied Cognition*. Indiana University Press, 2017.
- [43] A. L. Veloso, “Composing music, developing dialogues: An enactive perspective on children’s collaborative creativity,” *British Journal of Music Education*, vol. 34, no. 3, pp. 259–276, 2017.
- [44] P. Webster, “Creative thinking,” *Music Educators Journal*, vol. 76, no. 9, pp. 21–37, 1990.
- [45] C. Ford and N. Bryan-Kinns, “Identifying engagement in children’s interaction whilst composing digital music at home,” in *Proceedings of the 14th Conference on Creativity and Cognition*, ser. C&C ’22. New York, NY, USA: Association for Computing Machinery,

- 2022, p. 443–456. [Online]. Available: <https://doi.org/10.1145/3527927.3532794>
- [46] D. Miell and R. MacDonald, “Children’s creative collaborations: The importance of friendship when working together on a musical composition,” *Social Development*, vol. 9, no. 3, pp. 348–369, 2000.
- [47] P. R. Webster, “Children as creative thinkers in music,” *The Oxford handbook of music psychology*, pp. 421–428, 2009.
- [48] P. Burnard, “Rethinking ‘musical creativity’ and the notion of multiple creativities in music,” in *Musical creativity: Insights from music education research*. Routledge, 2016, pp. 27–50.
- [49] —, “Understanding children’s meaning-making as composers,” in *Musical creativity*. Psychology Press, 2006, pp. 127–149.
- [50] W. G. Crow, “Remixing the music curriculum: The new technology, creativity, and perceptions of musicality in music education,” Ph.D. dissertation, Institute of Education, University of London, 2012.
- [51] M. Hickey, “The computer as a tool in creative music making,” *Research Studies in Music Education*, vol. 8, no. 1, pp. 56–70, 1997.
- [52] S. E. Trehub and M. W. Weiss, *The Routledge Companion to Music Cognition*. Taylor & Francis Group, 2017, ch. Music Cognition: Developmental and Multimodal Perspectives, pp. 403 – 414.
- [53] S. Young, “The interpersonal dimension: A potential source of musical creativity for young children?” *Musicae Scientiae*, vol. 7, no. 1_suppl, pp. 175–191, 2003.
- [54] P. R. Webster, “Creative Thinking in Music, Twenty-Five Years On,” *Music Educators Journal*, vol. 102, no. 3, pp. 26–32, 2016. [Online]. Available: <http://journals.sagepub.com/doi/10.1177/0027432115623841>
- [55] P. Burnard, “The Individual and Social Worlds of Children’s Musical Creativity,” in *The Child as Musician: A Handbook of Musical Development*, G. McPherson, Ed. Oxford University Press, 2006, p. 0. [Online]. Available: <https://doi.org/10.1093/acprof:oso/9780198530329.003.0018>
- [56] Y. Wu and N. Bryan-Kinns, “Supporting Non-Musicians? Creative Engagement with Musical Interfaces,” in *Proceedings of the 2017 ACM SIGCHI Conference on Creativity and Cognition*. ACM, 2017, pp. 275–286. [Online]. Available: <https://dl.acm.org/doi/10.1145/3059454.3059457>
- [57] R. W. Hass, R. Reiter-Palmon, and J. Katz-Buonincontro, “Chapter 12 - are implicit theories of creativity domain specific? Evidence and implications,” in *The Creative Self*, ser. Explorations in Creativity Research, M. Karwowski and J. C. Kaufman, Eds. Academic Press, 2017, pp. 219–234. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/B9780128097908000121>
- [58] J. J. Y. Chung, S. He, and E. Adar, “The intersection of users, roles, interactions, and technologies in creativity support tools,” in *Proceedings of the 2021 ACM Designing Interactive Systems Conference*, ser. DIS ’21. New York, NY, USA: Association for Computing Machinery, 2021, p. 1817–1833. [Online]. Available: <https://doi.org/10.1145/3461778.3462050>
- [59] N. Davis, A. Zook, B. O’Neill, B. Headrick, M. Riedl, A. Grosz, and M. Nitsche, “Creativity support for novice digital filmmaking,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI ’13. New York, NY, USA: Association for Computing Machinery, 2013, p. 651–660. [Online]. Available: <https://doi.org/10.1145/2470654.2470747>
- [60] L. Benedetti, H. Winnemöller, M. Corsini, and R. Scopigno, “Painting with bob: assisted creativity for novices,” in *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, ser. UIST ’14. New York, NY, USA: Association for Computing Machinery, 2014, p. 419–428. [Online]. Available: <https://doi.org/10.1145/2642918.2647415>
- [61] R. A. Beghetto and J. C. Kaufman, “Toward a broader conception of creativity: A case for “mini-c” creativity.” *Psychology of aesthetics, creativity, and the arts*, vol. 1, no. 2, p. 73, 2007.
- [62] E. Cherry and C. Latulipe, “Quantifying the creativity support of digital tools through the creativity support index,” *ACM Trans. Comput.-Hum. Interact.*, vol. 21, no. 4, jun 2014. [Online]. Available: <https://doi.org/10.1145/2617588>
- [63] M. H. Hagen, D. S. Cruzes, L. Jaccheri, and J. A. Fails, “Evaluating digital creativity support for children: A systematic literature review,” *International Journal of Child-Computer Interaction*, p. 100603, 2023.
- [64] C. Zhang, C. Yao, J. Wu, W. Lin, L. Liu, G. Yan, and F. Ying, “StoryDrawer: A Child–AI Collaborative Drawing System to Support Children’s Creative Visual Storytelling,” in *CHI Conference on Human Factors in Computing Systems*. ACM, 2022, pp. 1–15. [Online]. Available: <https://dl.acm.org/doi/10.1145/3491102.3501914>
- [65] S. McRoberts, Y. Yuan, K. Watson, and S. Yarosh, “Behind the scenes: Design, collaboration, and video creation with youth,” in *Proceedings of the 18th ACM International Conference on Interaction Design and Children*, ser. IDC ’19. New York, NY, USA: Association for Computing Machinery, 2019, p.

- 173–184. [Online]. Available: <https://doi.org/10.1145/3311927.3323134>
- [66] C. K. Lam, “Technology-enhanced creativity in k-12 music education: A scoping review,” *International Journal of Music Education*, vol. 0, no. 0, p. 02557614231194073, 0.
- [67] A. Druin, “The role of children in the design of new technology,” *Behaviour and Information Technology*, vol. 21, no. 1, pp. 1–25, 2002.
- [68] J. C. Yip, F. M. T. Ello, F. Tsukiyama, A. Wairagade, and J. Ahn, ““money shouldn’t be money!” : An examination of financial literacy and technology for children through co-design,” in *Proceedings of the 22nd Annual ACM Interaction Design and Children Conference*, ser. IDC ’23. New York, NY, USA: Association for Computing Machinery, 2023, p. 82–93. [Online]. Available: <https://doi.org/10.1145/3585088.3589355>
- [69] C. A. Liang, K. Albertson, F. Williams, D. Inwards-Breland, S. A. Munson, J. A. Kientz, and K. Ahrens, “Designing an online sex education resource for gender-diverse youth,” in *Proceedings of the Interaction Design and Children Conference*. New York, NY, USA: Association for Computing Machinery, 2020, pp. 108–120.
- [70] P. Alves-Oliveira, P. Arriaga, A. Paiva, and G. Hoffman, “Yolo, a robot for creativity: A co-design study with children,” in *Proceedings of the 2017 Conference on Interaction Design and Children*, ser. IDC ’17. New York, NY, USA: Association for Computing Machinery, 2017, p. 423–429. [Online]. Available: <https://doi.org/10.1145/3078072.3084304>
- [71] M. Newman, K. Sun, I. B. Dalla Gasperina, G. Y. Shin, M. K. Pedraja, R. Kanchi, M. B. Song, R. Li, J. H. Lee, and J. Yip, ““i want it to talk like darth vader”: Helping children construct creative self-efficacy with generative ai,” in *Proceedings of the CHI Conference on Human Factors in Computing Systems*, ser. CHI ’24. New York, NY, USA: Association for Computing Machinery, 2024. [Online]. Available: <https://doi.org/10.1145/3613904.3642492>
- [72] J. Kato, T. Nakano, and M. Goto, “Textalive: Integrated design environment for kinetic typography,” in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, ser. CHI ’15. New York, NY, USA: Association for Computing Machinery, 2015, p. 3403–3412. [Online]. Available: <https://doi.org/10.1145/2702123.2702140>
- [73] “TextAlive,” <https://textalive.jp>.
- [74] “TextAlive Flow,” <https://flow.textalive.jp>.
- [75] “Incredibox,” <https://www.incredibox.com>.
- [76] “Sketch-a-Song,” <https://www.sketchasong.com>, v. 3.1.0.
- [77] J. Yip, T. Clegg, E. Bonsignore, H. Gelderblom, E. Rhodes, and A. Druin, “Brownies or bags-of-stuff? domain expertise in cooperative inquiry with children.” in *Proceedings of the 12th International Conference on Interaction Design and Children (IDC ’13)*, 2013.
- [78] A. L. Strauss, *Qualitative Analysis for Social Scientists*. Cambridge University Press, 1987.
- [79] C. E. Hill, S. Knox, B. J. Thompson, E. N. Williams, S. A. Hess, and N. Ladany, “Consensual qualitative research: An update,” *Journal of Counseling Psychology*, vol. 52, no. 2, pp. 196–205, 2005. [Online]. Available: <http://offcampus.lib.washington.edu/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=pdh&AN=2005-03263-009&site=ehost-live>
- [80] N. González, L. C. Moll, and C. Amanti, *Funds of knowledge: Theorizing practices in households, communities, and classrooms*. Routledge, 2006.
- [81] D. Norman, *The design of everyday things: Revised and expanded edition*. Basic books, 2013.
- [82] Y. Akiyama and S. Oore, “Placeandplay: a digital tool for children to create and record music,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI ’08. New York, NY, USA: Association for Computing Machinery, 2008, p. 735–738. [Online]. Available: <https://doi.org/10.1145/1357054.1357170>
- [83] S. C. Robyn M. Holmes, Lynn Romeo and M. Grushko, “The relationship between creativity, social play, and children’s language abilities,” *Early Child Development and Care*, vol. 185, no. 7, pp. 1180–1197, 2015.
- [84] L. S. Vygotsky, “Imagination and creativity in childhood,” *Journal of Russian & East European Psychology*, vol. 42, no. 1, pp. 7–97, 2004.
- [85] L. G. Hammershøj, “Creativity in children as play and humour: Indicators of affective processes of creativity,” *Thinking Skills and Creativity*, vol. 39, p. 100784, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1871187120302583>
- [86] S. W. Russ, “Play and creativity: Developmental issues,” *Scandinavian Journal of Educational Research*, vol. 47, no. 3, pp. 291–303, 2003.
- [87] C. Remy, L. MacDonald Vermeulen, J. Frich, M. M. Biskjaer, and P. Dalsgaard, “Evaluating creativity support tools in hci research,” in *Proceedings of the 2020 ACM Designing Interactive Systems Conference*, ser. DIS ’20. New York, NY, USA: Association for Computing Machinery, 2020, p. 457–476. [Online]. Available: <https://doi.org/10.1145/3357236.3395474>

- [88] J. Baer, "The importance of domain-specific expertise in creativity," *Roeper Review*, vol. 37, no. 3, pp. 165–178, 2015.
- [89] D. Wood, J. S. Bruner, and G. Ross, "The role of tutoring in problem solving," *Journal of Child Psychology and Psychiatry*, vol. 17, no. 2, pp. 89–100, 1976. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1469-7610.1976.tb00381.x>
- [90] C. hu Ko and M. ju Cho, "Aesthetics in young children's lives: From music technology curriculum perspective," *International Journal of Management, Economics and Social Sciences*, vol. 2, no. 4, pp. 265–273, 2013.
- [91] A. Davis and D. Weinshenker, *Digital Storytelling and Authoring Identity*. Cambridge University Press, 2012, ch. Digital Storytelling and Authoring Identity, pp. 47 – 74.
- [92] A. Druin and C. Solomon, *Designing multimedia environments for children*. New York: J. Wiley & Sons, 1996.
- [93] N. Wardrip-Fruin, *Expressive Processing : Digital Fictions, Computer Games, and Software Studies*, ser. Software Studies. MIT Press, 2009.
- [94] Shneiderman, "Direct Manipulation: A Step Beyond Programming Languages," *Computer*, vol. 16, no. 8, pp. 57–69, 1983. [Online]. Available: <http://ieeexplore.ieee.org/document/1654471/>
- [95] J. Jacobs, S. Gogia, R. Mundefinedch, and J. R. Brandt, "Supporting expressive procedural art creation through direct manipulation," in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, ser. CHI '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 6330–6341. [Online]. Available: <https://doi.org/10.1145/3025453.3025927>
- [96] A. Kay, "User interface: A personal view," *The art of human-computer interface design*, pp. 191–207, 1990.
- [97] Y. Yamamoto and K. Nakakoji, "Interaction design of tools for fostering creativity in the early stages of information design," *International Journal of Human-Computer Studies*, vol. 63, no. 4, pp. 513–535, 2005. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1071581905000480>
- [98] T. M. Amabile, *Creativity in Context: Update to "The Social Psychology of Creativity"*. Westview Press, 1996.

EL BONGOSERO: A CROWD-SOURCED SYMBOLIC DATASET OF IMPROVISED HAND PERCUSSION RHYTHMS PAIRED WITH DRUM PATTERNS

Nicholas Evans*

Behzad Haki*

Daniel Gómez-Marín

Sergi Jordà

Music Technology Group, Universitat Pompeu Fabra, Barcelona, Spain

nicholas.evans@upf.edu, behzad.haki@upf.edu, daniel.gomez@upf.edu, sergi.jorda@upf.edu

ABSTRACT

We present El Bongosero, a large-scale, open-source symbolic dataset comprising expressive, improvised drum performances crowd-sourced from a pool of individuals with varying levels of musical expertise. Originating from an interactive installation hosted at Centre de Cultura Contemporània de Barcelona, our dataset consists of 6,035 unique tapped sequences performed by 3,184 participants. To our knowledge, this is the only symbolic dataset of its size and type that includes expressive timing and dynamics information as well as each participant’s level of expertise. These unique characteristics could prove to be valuable to future research, particularly in the areas of music generation and music education. Preliminary analysis, including a step-wise Jaccard similarity analysis on a subset of the data, demonstrate that this dataset is a diverse, non-random, and musically meaningful collection. To facilitate prompt exploration and understanding of the data, we have also prepared a dedicated website and an open-source API in order to interact with the data.

1. INTRODUCTION

Symbolic drum datasets derived from live performance typically feature a select number of experienced drummers improvising or playing a composed piece. However, given that rhythm perception is a fundamental human trait [1], we contend that an expressive crowd-sourced drum dataset representing a diverse range of musical expertise could offer unique research utility not fulfilled by existing datasets. Although it may be possible to compile a dataset of this nature by scraping the web for recorded performances, it would be unlikely that a web-scraped dataset would include expressive performance information along with the level of expertise of each performer.

* Equal contribution

This past year, our research lab participated in an exhibition at Centre de Cultura Contemporània de Barcelona (CCCB) centered around the history, ethics, and creative possibilities of Artificial Intelligence. More specifically, we were tasked with preparing a 6-month installation that would be included in the "Data Worlds" section of the exhibition, the purpose of which was to examine the role of data in generative systems and the methods employed to gather data. We addressed both of these aspects with a two-part installation. In the first activity, participants used a “bongo-like” two-voice MIDI pad to interact with a Variational Auto-Encoder (VAE) model. This model had the capability to transform the participant’s tapped rhythmic sequences into symbolic multi-voice, expressive drum patterns [2,3], which were subsequently synthesized to audio. In the second activity, which serves as the focus of this paper, participants were given the opportunity to contribute to a crowd-sourced dataset that may later be used to improve the generative model they had just interacted with. They were invited to use the MIDI pad to tap along to a multi-voice drum pattern in a genre and tempo of their choosing. Providing minimal instructions, this task serves as an examination of how participants freely improvise alongside another rhythm. Upon completing the task, the participant could choose to contribute their tapped, improvised sequence to our public dataset or to submit nothing and delete their data.

In this paper, we present El Bongosero, a crowd-sourced expressive symbolic dataset consisting of 6,035 improvised tapped sequences performed by 3,184 participants with varying levels of musical expertise. Each sample contains expressive timing and dynamics information and is annotated with the participant’s level of musical expertise, the genre of the selected pattern, the chosen tempo, the total duration to complete the activity, and a user-rating for their performance and how much they enjoyed the exhibit. We anticipate that this dataset can promote further research in the following areas:

- Advancing the development of more nuanced generative models capable of accommodating a range of skill levels.
- Facilitating music education studies focused on music understanding and rhythm expertise.



© N. Evans, B. Haki, D. Gómez-Marín, S. Jordà. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** N. Evans, B. Haki, D. Gómez-Marín, S. Jordà, “El Bongosero: A Crowd-sourced Symbolic Dataset of Improvised Hand Percussion Rhythms Paired with Drum Patterns”, in *Proc. of the 25th Int. Society for Music Information Retrieval Conf.*, San Francisco, United States, 2024.

- Evaluating the proficiency, diversity, and creativity with which humans improvise rhythms.

Furthermore, the collection of data in this study adheres to rigorous ethical standards. Unlike other collection methods such as web-scraping, which may involve utilizing data in ways unintended by the original providers, our approach prioritizes clarity and consent with the participants throughout the entire process.

2. RELATED WORK

In this section, we will review other notable datasets consisting of human-performed recordings or synthesized web-scraped symbolic sequences. Reviewing these datasets aims to underscore the various applications and constraints associated with each approach.

The earliest open-source drum dataset we identified is the ENST-Drums dataset [4]. This is a fairly comprehensive dataset, consisting of around 225 minutes of annotated audio and video recordings of 3 live drummers. While still useful, this is significantly smaller than other datasets compiled via web-scraping or crowd-sourcing.

The TMIDT (Towards Multi-Instrument Drum Transcription) dataset, consisting of 259 hours worth of synthesized audio, was created via web-scraping every MIDI track from a freely available online collection¹ [5]. In a similar manner, the ADTOF (Automatic Drums Transcription On Fire) dataset, containing over 114 hours of annotated music, is constructed of openly shared² crowd-sourced symbolic annotations, typically a MIDI file, of real songs for use in rhythm games [6]. As such, this data does not contain detailed expressive information unlike Magenta’s MAESTRO (MIDI and Audio Edited for Synchronous TRacks and Organization) [7]. Although MAESTRO consists of ten years of International Piano-Competition performances on a Yamaha Disklavier, it is relevant to include here as it is a large-scale, crowd-sourced dataset. This dataset, which has been used effectively in generative models, is comprised of over 172 hours of finely aligned (~3 ms) audio waveforms and expressive MIDI information.

Similar to MAESTRO, Magenta’s Groove MIDI Dataset (GMD) is composed of 13.6 hours of aligned MIDI and (synthesized) audio of human-performed, expressive drumming [8]. The nature of this dataset has proven to be useful for predictive generative models such as GrooVAE [8] as well as for perceptual experiments such as TapTamDrum [9].

The TapTamDrum dataset was the result of an experiment in which 4 experienced drummers were given the task of reducing expressive, multi-voice drum patterns from Magenta’s GMD to dual-voice representations. The resultant dataset includes 1,116 total dualizations annotated with expressive timing and velocity from 345 unique patterns.

Dataset	Format			Annotations		
	Audio	Symbolic	Human-Performed	Velocity	Genre	Level of Expertise
ENST	✓	✓	✓			
TMIDT	✓	✓				
ADTOF	✓	✓				
GMD	✓	✓	✓	✓	✓	
MAESTRO	✓	✓	✓	✓		
TapTamDrum	✓	✓	✓	✓		
MAST	✓		✓			
El Bongosero		✓	✓	✓	✓	✓

Table 1. Comparison of datasets.

Lastly, there is the MAST (Musical Aptitude Standard Test) Rhythmic Dataset, sourced from university examinations in which candidates were expected to reproduce a tapped rhythmic pattern after it had been played two times by a member of the jury [10]. Therefore, this audio dataset includes 2,681 recordings of jury members performing the target rhythm, along with 1,040 recordings of student attempts annotated with their grade (pass or fail).

Table 1 offers a comparison of the datasets based on two key attributes: format and annotations. Format indicates whether the dataset comprises audio or symbolic samples and whether these samples are derived from recorded human performances. Annotations, on the other hand, encompass details such as the presence of velocity annotations for each onset, the genre of the sample, and the level of expertise of the performer. As shown in the table, El Bongosero is the only dataset that annotates the performer’s level of expertise.

3. METHODOLOGY

As mentioned above, the installation consisted of two parts. In the first part, participants were to engage with a generative model. In the second part, they were asked to contribute to a dataset that may be used for training future iterations of the generative model used in the first part. The focus of this paper is on the latter part of the installation, specifically, the collection of a symbolic dataset of rhythmic improvisations played alongside a selected number of drum patterns.

As the installation was to be used in a public exhibition space, it was imperative to design an interface that could accommodate a broad spectrum of participants without assuming specific technical or musical expertise. To this end, we made several decisions in designing the data collection stage of the installation. First, we ensured that the interactive elements in the system were nearly identical to the first stage of the installation. This strategy was aimed at eliminating any need for participants to acquaint themselves with the mechanics of the system. Second, we minimized the instructions provided to participants. The intention here was to encourage the participants to improvise freely using their personal intuition and creativity, rather than adhering to a very specific procedure. Lastly, before initiating the second part of the installation, we informed the participants that we would ask for their consent to contribute to our dataset at the conclusion of this activity. The purpose of this approach was to ensure that participants

¹ <http://www.midiworld.com>

² <https://rhythmgamingworld.com/>

were fully aware of this aspect of the activity prior to deciding if they wished to interact. While the primary motivation behind implementing this level of transparency was to adhere to ethical principles, it was also our aim to foster more open and genuine interaction with the installation by making participants feel valued and secure.

In the following subsections we discuss the tasks presented to the participants (3.1), the installation setup (3.2), and the drum pattern curation process (3.3).

3.1 Overview of Tasks

Figure 1 provides an overview of the tasks involved in the installation.

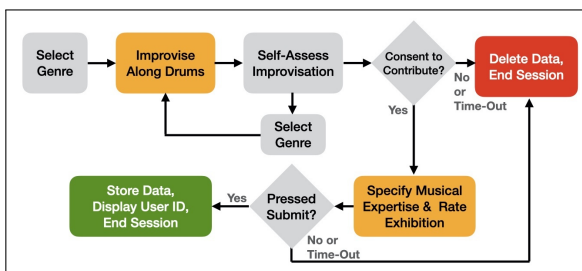


Figure 1. Flow-chart of the installation steps

The main objective of the data collection part of the installation was to present the participants with a randomly selected drum pattern and ask them to improvise alongside the pattern. To accommodate the diverse musical tastes and backgrounds of the participants, we allowed the participants to select the genre of the drum pattern.

Once the genre was selected, the improvisation environment was initiated. In this environment, the participant was presented with a looping 2-bar drum pattern and was asked to improvise alongside it using a provided two-voice MIDI pad.

The starting tempo of the session would be associated with the selected drum pattern, however, to accommodate participants of various skill levels, we allowed them to modify the tempo of the session. Each tap on the MIDI pad was recorded in a real-time looping 2-bar buffer, allowing participants to listen to previous taps and overdub additional taps. Lastly, participants were given as much time as needed.

Once the participant stopped the session, we asked them to self-assess their performance using a 5-level Likert scale. Once the assessment was provided, the participant was given two choices. They could select a new pattern to improvise alongside or they could finish the session. Once the participant finished the session, they were asked if they wish to contribute to the dataset. If they decided not to contribute, the session would end. Otherwise, they were presented with a brief questionnaire and then subsequently asked to press a button to explicitly submit their data.

Given that this was a public installation, we presented consenting users with only 2 questions: (1) "How would you assess your level of musical expertise?", and (2) "How much did you enjoy this exhibit?". In order to assure the

participants that the only aim of the installation was to collect improvisations, as opposed to metadata related to the participants, we avoided any demographic questions on gender, age, and occupation. Recognizing the vast spectrum of musical proficiency among participants, from novices to experienced musicians, the question on "Musical Expertise" aimed to contextualize the improvisational outcomes within a broader narrative of skill and experience. Furthermore, we recognize that the term "expertise" in this context may be subject to interpretation, with participants not necessarily associating it solely with musical proficiency. Our intention was to allow participants to define "expertise" based on their own understanding within the context of their improvisations.

Once the final questions were answered, the submission button would be enabled to finalize the contribution. Note that participant data was only added to the dataset if the "Submit" button was pressed. That is, we wanted to ensure that participants explicitly consented to the contribution. In any case that the participants left the session mid-experiment, explicitly chose not to contribute, or forgot to press the submission button, their data was immediately deleted.

3.2 Installation Setup

The installation, shown in Figure 2, consisted of a touch screen application and an *Embodme's ERAE Touch MPE* controller³ for registering the improvisations.

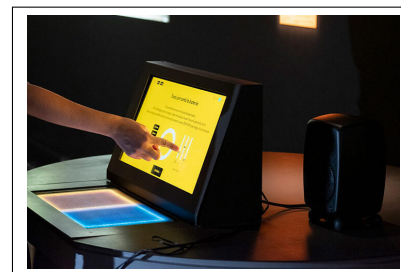


Figure 2. A photo of the installation (photo credit: CCCB)

The touchscreen application was used to prompt the participant and to allow the user to navigate through different stages of the installation. During the improvisation sessions, the graphical interface visualized the "bongo" performance using a circular representation of the 2-bar looping buffer filled with each tap onset registered on the MIDI pad (refer to the center of Figure 3 for the actual graphic representation). In this section, the participant was allowed to remove a specific onset by double tapping its location in the buffer, or to remove all of the onsets using a dedicated button; however, they were not allowed to reposition any registered onsets. In other words, the timing of the onsets were only to be associated with the timing registered from the performed taps on the MIDI pad.

The visual interface was implemented using the *PyQt5*⁴ Graphical User Interface (GUI) toolkit. To ensure

³ <https://www.embodme.com/erae-touch>

⁴ <https://www.qt.io/>

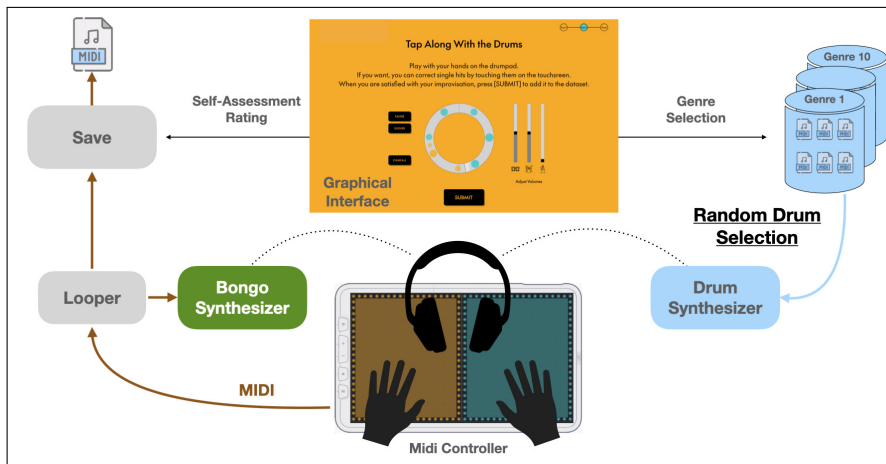


Figure 3. Installation setup

precise synchronization between the synthesis and playback of the source drum pattern and the MIDI recording of each improvisation, we developed a C++ backend using *JUCE* framework⁵. The recorded performances are provided in two formats: (1) a linear sequence preserving the timing of each tap onset throughout the activity’s entire duration, and (2) a 2-bar loop preserving the timing of each overdubbed onset within the 2-bar recording buffer.

As shown in Figure 3, the touch MIDI pad was customized into two distinct pitch regions, with the right region pitched lower than the left. Each “bongo” tap was displayed on the graphical interface using a circle located within the 2-bar looping buffer. Each circle was color-coded to match the region from which the tap onset was registered and the radius of the circle was correlated with the velocity of the registered tap onsets.

In order to ensure that participants could properly listen to the sounds of the installation and to reduce the the possibility of a participant feeling that their performance was being judged by other visitors, we decided to only provide headphones to the participants rather than loudspeakers⁶.

Lastly, dedicated sliders were provided on the interface to allow the participant to adjust the volumes of the bongo sounds and the drum sounds as needed. Moreover, a dedicated slider (initially muted) was also provided to participants to utilize a synchronized metronome track if needed.

3.3 Drum Pattern Selections

The source drum patterns were in a 4/4 metric, selected from a large in-house collection of over 200,000 MIDI files, which included both open-source and proprietary MIDI collections. The MIDI files in this collection were divided into 10 genres: Afrobeat, Afrocuban, Bossanova, Disco, Electronic, Funk, Hiphop, Jazz, Rock, and Soul.

For each pattern in the collection, we extracted the rhythmic features provided in *GrooveToolbox* [11] and *Rhythm Toolbox* [12]. The extracted features were normalized and subsequently mapped to a two-dimensional

space using Principal Component Analysis (PCA). For each genre, the mapped values were grouped into 100 clusters using k-means clustering method, and subsequently, a single pattern was randomly selected from each cluster.

In order to ensure a small subset of patterns with a sizeable collection of varied responses per pattern, we opted to limit the Electronic genre to 16 patterns. We selected this genre as we suspected it would be the most popular choice among participants.⁷

4. DATASET

In this section the contents of the collected dataset are described. A total of 4 variables were recorded per participant (ID, number of attempts, level of musical expertise, and exhibition rating). The ID was assigned in sequential order and each participant could attempt multiple improvisations. For each attempt 8 variables were collected (attempt duration, assessment time, attempt tempo, drum pattern, genre, improvisation pattern, level of expertise, and exhibition rating).

Table 2 presents a summary segmented by participants’ level of musical expertise. The mean level of musical expertise is 2.95 (std = 1.25). The most common level of musical expertise is level 2 (915 participants) and the least common is level 1 (392 participants). The mean number of attempts per participant is 1.89 (std = 1.32). The highest amount of attempts were carried out by participants of musical expertise level 2 (1692) and the lowest amount of attempts were carried out by participants of musical expertise level 1 (691). The mean number of unique patterns presented per level is 591.8 (std = 103.8). Participants with musical expertise of level 2 were exposed to the largest number of unique musical patterns (720) while participants with musical expertise of level 1 were exposed to the least number (433). The mean number of attempts increases with the level of musical expertise, as participants with more expertise made more attempts on average.

Figure 4 presents the number of attempts per genre. The mean number of attempts per genre is 603.5 (std =

⁵ <https://juce.com/>

⁶ A loudspeaker was available in the setup (as in Figure2), however, it was only used in special occasions decided by CCCB organizers.

⁷ The results discussed in next section confirm this speculation.

	Level of Musical Expertise				
	1	2	3	4	5
No. participants	392	915	805	594	478
Attempt count	691	1692	1536	1074	1042
Mean no. attempts	1.76	1.85	1.91	1.81	2.18
Unique patterns	433	720	691	562	553

Table 2. Summary of participants, attempts, patterns, and musical expertise.

169.67). The Electronic and Rock genres represent the highest amount of attempts (919 and 850 respectively) while the Soul genre represents the lowest amount (371).

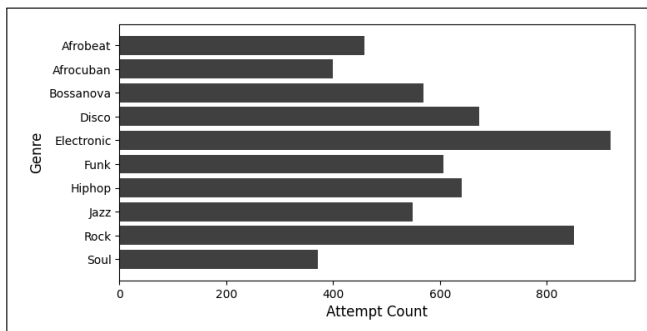


Figure 4. Histogram of attempts per genre.

Figure 5 presents an overview of the number of attempts per level of expertise and genre. The combination with the highest number of attempts is the Rock and Electronic genres combined with expertise levels 2 and 3. The combination with the least attempts is the Afrobeat genre and expertise level 1.

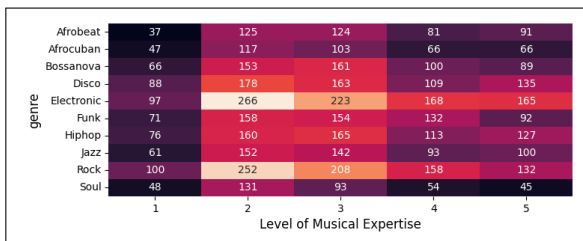


Figure 5. Attempts per genre and level of expertise.

As described in Section 3.3, all drum patterns used in the installation are in the 4/4 metric. Figure 6 presents the step densities of both the original patterns and the recorded improvisations, obtained by adding onsets at each step and dividing by the step with most onsets. Patterns are wrapped to 16 steps for convenience and onsets quantized to the closest 16th note. In order to establish a comparison, the normalized theoretical metrical weight is displayed. Notice how densities at each inter-pulse group of steps (0-3, 4-7, 8-11, 12-15) complies with the "high, low, mid, low" contour expressed in the theoretical metrical weights for a 4/4 rhythmic pattern. This suggests that participants consistently induced a meter from the source drum patterns. The improvised rhythms by the participants (Figure 6 below) showcase the same general intra-pulse contour with two differences. First, the low contours are higher (uneven

steps), and second, the first step contains less onset density than its intra-pulse set (steps 1, 2 and 3). However steps 1, 2, and 3 comply with the "low-mid-low" contour observed in the original pattern's intra-pulse density. We believe many of the participants were slightly inaccurate at the beginning of the loop, thus causing onsets intended for the first step to be played early, registering in the last step of the previous bar, or played late, registering in the second step of the current bar.

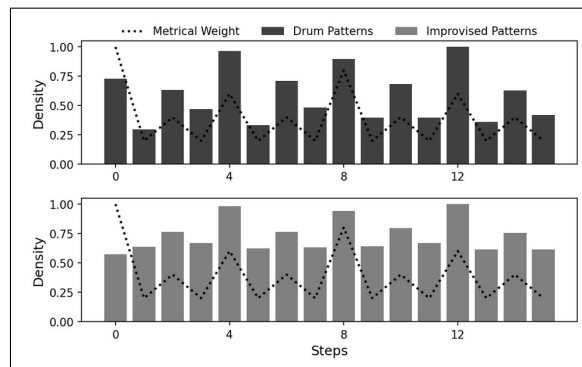


Figure 6. Onset density and metrical weight per step.

Typical of crowd-sourced datasets, these general observations led us to identify some instances of crowd-driven bias. Specifically, we observed a preference for two genres (Rock and Electronic) out of a possible ten, along with a distribution of musical expertise leaning towards mid-low levels. On the other hand, the general compliance of participants' patterns with metrical expectations suggest that their improvisations were carried out under expected pulse-entrainment conditions. Thus, in general, it seems that the data gathered corresponds to sensory-motor activities and not a random collection of taps.

5. PRELIMINARY INSIGHTS

As explained in Section 3.3, we limited the number of Electronic patterns to 16 in order to increase the number of reproductions per pattern for different levels of musical expertise. The brief preliminary analysis presented here focuses solely on the Electronic genre and explores the patterns used and assesses the similarity between the reproduced drum patterns and the participants' improvisations.

The number of attempts per Electronic pattern is presented in Figure 7. The range of attempts fluctuates between 46 (pattern 1) and 65 (pattern 9). The mean is 57.44 attempts and the standard deviation is 5.33.

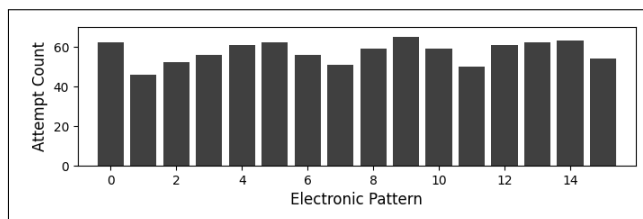


Figure 7. Number of attempts per Electronic pattern.

In order to establish a first metric that can account for comparing the multi-voice drum patterns with the participants' tapped improvisations, the Jaccard similarity metric is used. Jaccard is a common similarity metric used in data analysis, especially suited for comparing two sets of elements. The simplest implementation of the metric is the quotient of the sum of the intersection elements with the sum of the union elements. The more elements in common between the intersection and the union, the closer the Jaccard similarity gets to 1.

We implemented Jaccard similarity comparing a step-wise flattened version of the drum pattern and participants' improvisations. The rationale of this metric is: in a step where (at least) one onset in the drum pattern is observed, (at least) one onset is expected in the participant's improvisation. The intersection is composed of steps with onsets in the drum pattern that coincide with steps with onsets in the improvised pattern. The union comprises all steps from the pattern and the improvisation containing an onset. If a participant produces an onset every time the drum pattern produces an onset, Jaccard similarity is equal to one. If all of a participant's onsets are on steps where the drum pattern is silent, Jaccard similarity is 0.

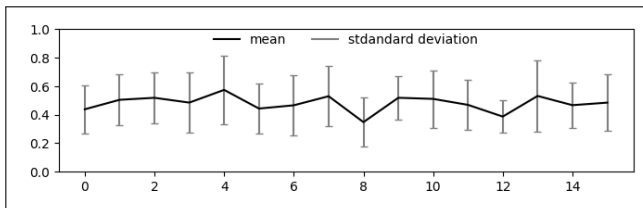


Figure 8. Jaccard similarity means and standard deviation for every pattern in the Electronic music genre.

Figure 8 shows that all participant improvisations to Electronic patterns exhibit a very similar mean (from 0.348 for pattern 8 to 0.574 for pattern 4) and spread standard deviation (from 0.115 for pattern 12 to 0.249 for pattern 13). There is no apparent agreement (there are no high mean values) towards any of the Electronic patterns, suggesting diversity in the improvisations for all patterns of this genre.

For more detail, Figure 9 presents a spread of similarity by Electronic pattern and level of musical expertise. The expertise level with the highest Jaccard similarity sum for all patterns (5.98) is level 1 while level 2 has the lowest Jaccard similarity sum for all patterns (5.63). The most diverse case, signified by a low mean Jaccard similarity (0.18), is observed in improvisations for pattern 7 performed by participants of expertise level 5. On the other hand, improvisations with the most average agreement with the reference, signified by a high mean Jaccard similarity (0.51), is observed in improvisations for pattern 10 performed by participants of expertise level 1.

The consistent mid agreement presented in Figure 8 and Figure 9 suggests improvisations were not exhibiting an automatic onset-for-onset behavior. On the contrary, there seems to be rich musical behavior to be explored within the Electronic genre.

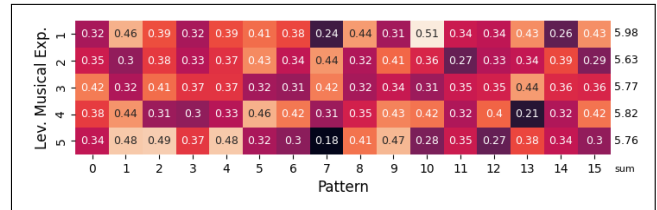


Figure 9. Jaccard similarity among all improvisations for different musical levels and pattern.

6. DISCUSSION AND CONCLUSION

In this paper, we introduced El Bongosero, a crowd-sourced dataset consisting of 6,035 tapped improvised rhythms performed by a total of 3,184 participants with varying levels of musical expertise. The improvisations are collected across 10 genres, each corresponding to a set of 100 unique drum patterns selected for that genre, except for the Electronic genre, which includes 16 samples. The main focus of this work was the collection, curation, and organization of this data from an interactive public exhibit.

Preliminary analysis, including a step-wise Jaccard similarity analysis on the Electronic genre data subset, demonstrate that this dataset is a diverse, non-random, and musically meaningful collection of improvised rhythms. Through our review of existing datasets, we identified the unique qualities of El Bongosero that could make it particularly useful for music generation and music education research. More specifically, it is the combination of the sheer number of participants, the diverse range of participants' level of musical expertise, and the inclusion of expressive performance information that distinguishes this dataset.

For example, in the context of a model that generates music based on a rhythmic input, a skilled musician may have different expectations than a novice musician regarding how a model should interpret an input rhythm or how it should respond to subtle variations in timing or dynamics. Integrating a diverse crowd-sourced dataset, such as El Bongosero, with the development of generative models could prove to be an effective approach to constructing more nuanced models that are capable of adjusting to individuals with varying skill levels.

Similarly, in the context of music education, deep analysis of El Bongosero may allow educators to gain insights into the learning trajectory of percussion students and help them to better develop a curriculum that supports skill development. As an evaluation tool, this dataset could serve as a valuable resource for developing assessments and criteria for drumming proficiency. Furthermore, researchers may be able to identify key indicators of musical growth and proficiency by comparing performances across different levels of expertise.

To conclude this work and facilitate prompt exploration of the collected data, we have prepared a dedicated website and an open-source API available at:

<https://elbongosero.github.io/>

7. ETHICS STATEMENT

Conscientious consideration of ethical principles has been central throughout this project. We recognize that as researchers it is our responsibility to ensure that there is complete transparency of the collection process and that participants have a full understanding of their involvement. Accordingly, this study attempts to uphold ethical standards at every stage of the data life cycle, from collection to utilization.

Firstly, the installation was crafted so that prior to starting the activity, participants were explicitly notified that we would later request their permission to store the data they generate while interacting with the exhibit. At the end of the activity, participants had to explicitly consent once more in order to be included in the dataset. If they declined, or took no action, their data was not stored.

In addition to ensuring explicit consent from participants, we also gave careful consideration to exactly which data we collected. To this end, we opted to collect no personal or demographic information from the participants. The collected data from consenting participants included only their interactions with the installation, resulting in a symbolic representation of their tapped improvised pattern, along with their responses to two questions: “How would you assess your level of musical expertise?” and “How much did you enjoy this exhibit?”.

Moreover, ethical considerations extend beyond the initial data collection phase to encompass the subsequent use and application of the data. As stewards of this dataset, we are committed to employing the collected data solely for academic research purposes, ensuring that it is used in a manner consistent with what was communicated to participants.

8. ACKNOWLEDGMENTS

This research was partly funded by the Secretaría de Estado de Digitalización e Inteligencia Artificial, and the European Union-Next Generation EU, under the program Cátedras ENIA 2022. "IA y Música: Cátedra en Inteligencia Artificial y Música" (Reference: TSI-100929-2023-1).

9. REFERENCES

- [1] H. Honing, “Without it no music: beat induction as a fundamental musical trait,” in *Annals of the New York Academy of Sciences*, vol. 1252, 2012, pp. 85–91.
- [2] B. Haki, M. Nieto, T. Pelinski, and S. Jordà, “Real-Time Drum Accompaniment Using Transformer Architecture,” in *Proceedings of the 3rd Conference on AI Music Creativity (AIMC)*, September 2022.
- [3] N. Evans, B. Haki, and S. Jorda, “GrooveTransformer: A Generative Drum Sequencer Eurorack Module,” in *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*, September 2024.
- [4] O. Gillet and G. Richard, “ENST-Drums: an extensive audio-visual database for drum signals processing,” in *Proceedings of 7th International Society for Music Information Retrieval Conference (ISMIR 2006)*, Victoria, BC, Canada, October 2006, pp. 156–159.
- [5] R. Vogl, G. Widmer, and P. Knees, “Towards multi-instrument drum transcription,” in *Proceedings of the 21th International Conference on Digital Audio Effects (DAFx18)*, Aveiro, Portugal, September 2018, pp. 57–64.
- [6] M. Zehren, M. Alunno, and P. Bientinesi, “ADTOF: A large dataset of non-synthetic music for automatic drum transcription,” in *Proceedings of the 22nd International Society for Music Information Retrieval Conference (ISMIR 2021)*, Online, November 2021, pp. 818–824.
- [7] C. Hawthorne, A. Stasyuk, A. Roberts, I. Simon, C.-Z. A. Huang, S. Dieleman, E. Elsen, J. Engel, and D. Eck, “Enabling factorized piano music modeling and generation with the maestro dataset,” in *Proceedings of the International Conference on Learning Representations (ICLR 2019)*, New Orleans, Louisiana, USA, May 2019, pp. 9092–9103.
- [8] J. Gillick, A. Roberts, J. H. Engel, D. Eck, and D. Baman, “Learning to groove with inverse sequence transformations,” in *Proceedings of the 36th International Conference on Machine Learning (ICML 2019)*, Long Beach, California, USA, vol. 97, June 2019, pp. 2269–2279.
- [9] B. Haki, B. Kotowski, C. L. I. Lee, and S. Jordà Puig, “TapTamDrum: a dataset for dualized drum patterns,” in *Proceedings of the 24th Conference of the International Society for Music Information Retrieval (ISMIR 2023)*, Milan, Italy, November 2023, pp. 114–120.
- [10] F. Falcao, B. Bozkurt, X. Serra, N. Andrade, and O. Baysal, “A dataset of rhythmic pattern reproductions and baseline automatic assessment system,” in *Proceedings of the 20th Conference of the International Society for Music Information Retrieval (ISMIR 2019)*, Delft, The Netherlands. International Society for Music Information Retrieval (ISMIR), November 2019, pp. 439–445.
- [11] F. Bruford, O. Lartillot, S. McDonald, and M. B. Sandler, “Multidimensional similarity modelling of complex drum loops using the GrooveToolbox,” in *Proceedings of the 21th International Society for Music Information Retrieval Conference (ISMIR 2020)*, Montreal, Canada, October 2020, pp. 263–270.
- [12] D. Gómez-Marín, S. Jordà, and P. Herrera, “Drum rhythm spaces: From polyphonic similarity to generative maps,” in *Journal of New Music Research*, vol. 49, no. 5. Taylor & Francis, 2020, pp. 438–456.

UTILIZING LISTENER-PROVIDED TAGS FOR MUSIC EMOTION RECOGNITION: A DATA-DRIVEN APPROACH

J. Affolter, M. Rohrmeier

Ecole Polytechnique Fédérale de Lausanne, EPFL

ABSTRACT

This work introduces a data-driven approach for assigning emotions to music tracks. Consisting of two distinct phases, our framework enables the creation of synthetic emotion-labeled datasets that can serve both Music Emotion Recognition and Auto-Tagging tasks. The first phase presents a versatile method for collecting listener-generated verbal data, such as tags and playlist names, from multiple online sources on a large scale. We compiled a dataset of 5,892 tracks, each associated with textual data from four distinct sources. The second phase leverages Natural Language Processing for representing music-evoked emotions, relying solely on the data acquired during the first phase. By semantically matching user-generated text to a well-known corpus of emotion-labelled English words, we are ultimately able to represent each music track as an 8-dimensional vector that captures the emotions perceived by listeners. Our method departs from conventional labeling techniques: instead of defining emotions as generic “mood tags” found on social platforms, we leverage a refined psychological model drawn from Plutchik’s theory [1], which appears more intuitive than the extensively used Valence-Arousal model.

1. INTRODUCTION

Several studies on music listener behavior have identified an increasing interest in music discovery based on its emotional content [2]. It is therefore hardly surprising that the field of Music Emotion Recognition (MER), which explores how emotions can be identified in music [3], is a growing area of research.

MER research is dominated by the use of supervised machine learning methods, in which systems are trained on music excerpts previously labeled with emotion descriptors through crowdsourcing. A major hurdle in this field is the lack of large-scale emotion-annotated datasets [4]. The complexity of collecting suitable training data contributes significantly to this issue, as the process is time-consuming, labor-intensive and expensive. The subjective

nature of musical emotions further complicates the data collection process [5].

Recognizing language as a powerful medium for conveying musical signification, we proceed on the premise that emotions can be inferred from textual data—specifically, from listener-generated tags and playlist names on music platforms. We thus introduce a novel method for assigning, to any given song, an emotion vector within an 8-dimensional space defined by Plutchik’s model. This enables us to propose a new dataset comprising 5,892 tracks, specifically tailored for Music Emotion Recognition (MER) tasks.

2. RELATED WORK

Yuan et al. [4] propose the Music Audio Representation Benchmark for universal Evaluation (*MARBLE*) as a unified standard for assessing various Music Information Retrieval (MIR) tasks. They employ 12 publicly available datasets to evaluate 18 distinct tasks, including the *Emo-music* [5] and *MTG-MoodTheme* [6] datasets for MER evaluation. Table 1 provides an overview of commonly used datasets in MER research, along with their size, data collection method and emotion labeling approach.

Dataset	Size	Data collection	Emotion model	Ref.
Emomusic	744	C	AV	[5]
MTG-MT	17,982	DM	56 labels	[6]
AMC	600	C	5 clusters	[7]
EMMA	364	C	GEMS	[8]
CAL500	500	C	174 labels	[9]
MoodSwings	240	Game	AV	[10]
NTWICM	2,648	C	AV	[11]
Soundtracks	470	C	9 labels	[12]
DEAP	120	EEG	AVD	[13]
AMG1608	1,608	C	AV	[14]
Emotify	400	Game	GEMS	[15]
Moodo	200	C	AV	[16]
4Q-emotion	900	C	AV	[17]
PMemo	794	EEG	AV	[18]

Table 1: Overview of existing MER datasets.

C: crowdsourcing, DM: data mining, AV(D): arousal/valence/dominance, EEG: electroencephalography, GEMS: Geneva Emotion Music Scale

Through the examination of these datasets, three areas for potential improvement have been identified.

Dataset size. Datasets annotated with labels according to a psychological emotion model (AV, AVD, GEMS) do not exceed 2,648 tracks, with an average size of 801. Furthermore, most datasets fail to cover a wide range of



musical genres—they are often limited to four or fewer, or do not provide clear genre definition, resulting in imbalanced datasets. This limited size and diversity complicate the training of accurate music emotion recognition models, raising concerns about issues such as group fairness and generalization capability.

Data collection. Most datasets rely on human annotations from crowdsourcing/online games, or from EEG experiments, which, while reliable, are both expensive and time-consuming. Moreover, these datasets encounter challenges in participant diversity. Typically, the assignment of an emotion label to a track requires consensus between few annotators. In the case of datasets featuring mood tags, it is common for tracks to have, on average, no more than two tags associated with them, potentially leading to misleading data.

Emotion model. Emotion labeling generally falls into two categories. (1) Mood-based emotion tags. For instance, in *MTG-MoodTheme*, the 56 emotion labels correspond to tags directly retrieved from the Jamendo music platform. This can result in a large number of emotion labels, making it difficult for end-users to understand and use the system. This approach may also not align with an established emotion model. (2) Discrete- or continuous-based annotations derived from a predefined emotion model. While the VA model, with its two-dimensional structure, has been criticized to be restrictive and open to overly subjective interpretation [19], the Geneva Emotion Music Scale (GEMS) is specifically crafted for the music domain, and proposes a more detailed taxonomy.

3. APPROACH

This section introduces the key design decisions underlying our methodology for inducing music-evoked emotion descriptors from a collection of tracks. Our approach aims to enhance the study of emotions in music by introducing a novel representation of emotions based on a psychological model that has been hitherto unacknowledged in the field of MER.

3.1 Plutchik’s Emotion Model

We recognize the importance of grounding our research framework in a well-established emotion model. In search of a more intuitive alternative than the Valence/Arousal (VA) framework, we opted for Plutchik’s model, which, to our knowledge, has not yet been utilized in the field of music and, we believe, strikes a good balance between complexity and usability. Plutchik’s emotion model is founded on eight primary emotions (joy, fear, anger, sadness, disgust, surprise, anticipation, trust) that we believe are accessible and instinctive for listeners. As a recognized model in psychology, it has been employed across various domains beyond music, enabling us to leverage existing resources, such as the *NRC Lexicon* [20], a crowdsourced list of 14, 182 English words and their binary associations with Plutchik’s primary emotions. Highly aligned with

our research goal, its single-word structure bears a strong resemblance to our textual data, which includes tags and playlist names. Its origin in actual annotations by human subjects, rather than derivative interpretations, is also crucial to the accuracy of our emotion mappings. Furthermore, the model’s categorical approach can be expanded by combining emotions as depicted in Figure 1, thus enabling the representation of more complex emotions.

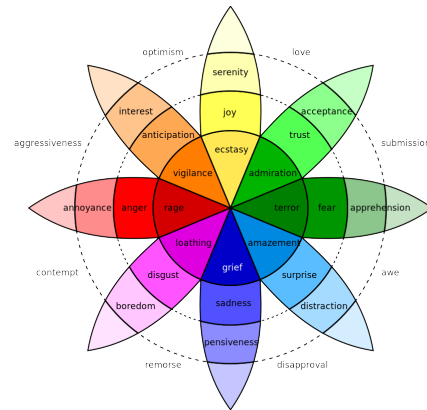


Figure 1: Plutchik’s wheel of emotions [1].

3.2 Emotion Vector Representation

Instead of viewing emotions as discrete labels, as traditional classification MER systems do, we propose to represent a music track as an 8-dimensional emotion vector that captures the emotions perceived by listeners. We define our emotion domain mathematically as a vector space V , represented by a basis $B = \{\text{joy, fear, anger, sadness, disgust, surprise, anticipation, trust}\}$. Each emotion in B corresponds to a standard basis vector in V , with $e_{\text{joy}} = [1, 0, \dots, 0]$ through $e_{\text{trust}} = [0, \dots, 0, 1]$. The emotion vector v of a track is defined as

$$v = \sum_{i \in B} \lambda_i e_i, \tag{1}$$

where $\lambda_i \in [0, 1]$ represents the intensity of emotion i .

By representing emotion intensities within an 8-dimensional vector, our framework aims to effectively capture the complex spectra of music-evoked emotions and discern the subtle emotional nuances of music tracks.

3.3 Textual Data Encoding

To effectively encode textual data, we selected the Sentence-BERT (SBERT) model [21], an NLP neural network known for generating semantically meaningful embeddings at the sentence level. Specifically fine-tuned for semantic similarity tasks, SBERT enhances the original BERT architecture by integrating siamese and triplet network structures. NLP techniques such as Semantic Search are significantly enhanced with this encoding model, as it enables the retrieval of the closest elements in the embedding space based on semantic similarity.

By using SBERT, we are able to create single embeddings that accurately encode the semantic content of each

textual element while preserving context. This choice is particularly suitable for our data, which includes short phrases—such as single- and multi-word tags, playlist names, and English words from the *NRC Lexicon*—that need to be compared in terms of semantic similarity.

4. IMPLEMENTATION: A TWO-STAGE FRAMEWORK FOR EMOTION ATTRIBUTION

Building upon the challenges and insights discussed in Section 2, we introduce a two-stage framework for extracting music-evoked emotions from a collection of tracks. Drawing inspiration from *MTG-MoodTheme*, the first phase focuses on collecting verbal tags through data mining across platforms such as *Last.fm* and *Spotify*, while the second phase leverages NLP techniques to computationally associate emotion vectors with music tracks by relying solely on the tags acquired during phase one.

4.1 Large-Scale Listener-Generated Data Collection

4.1.1 Track Selection Process.

We started with a baseline dataset of 20,000 music tracks, spanning 20 distinct genres. We selected the top 1,000 tracks with the highest popularity index on *Spotify* for each genre, in order to increase the likelihood of finding them in multiple sources when retrieving tags.

4.1.2 Data Mining.

We extracted listener-generated tags from three popular rating websites—*Last.fm*, *AllMusic*, *Rate Your Music*¹—and retrieved playlist names from the dataset provided for the *Spotify Million Playlist Dataset Challenge*, which includes 1,000,000 playlists created by *Spotify* listeners between 2010 and 2017 [22].

4.1.3 Data Pre-Processing.

While the tags from *Rate Your Music* and *All Music* were already normalized by the platform, those from *Spotify* and *Last.FM* required extensive cleaning. The objective was twofold: first, to eliminate irrelevant data, such as playlist names along the lines of ‘*Favorite hits*’, and second, to remove tags that could introduce bias when assigning emotions. Indeed, some tags—such as album names, artist names, or musical genres—are intended as mere filters for finding music. Others, like ‘roadtrip tunes’, are too neutral and may suggest contexts unrelated to emotions, while tags such as ‘love it’ reflect personal opinions and could bias our results by conflating perceived with induced emotions [23].

We first translated multilingual text into English, expanded abbreviations, replaced slang words and emoticons with their standard equivalents, and corrected misspelled words. We then implemented four iterative filtering processes to eliminate listener-generated tags that cannot be considered emotion descriptors.

Metadata Filtering. Since artists, song titles, album names, and musical genres were retrieved as metadata for all tracks in our dataset, we first eliminated any textual inputs containing terms from these categories. The set of musical genres was expanded to include a broader range beyond the 20 genres under study.

Named Entities Filtering. We then used the pre-trained BERT model fine-tuned for Named Entity Recognition (NER)² to identify named entities within predefined categories, such as person names, song titles, and locations. We filtered out sequences containing at least one token classified as a named entity of a target category with a confidence score above 0.9.

Neutral Tag Filtering. Sentiment analysis was subsequently performed using the pre-trained RoBERTa model fine-tuned for this task³. We removed tags with a neutral sentiment proportion greater than 70% (where 100% was distributed among positive, neutral, and negative sentiments for each input sequence). This threshold was deliberately chosen to avoid losing potentially useful tags like ‘energetic’. In subsequent stages of this framework, tags that are too neutral and not intended for emotion description will nonetheless be matched with words from the *NRC Lexicon* that do not have associated emotions, thereby not impacting the final emotions associated with music tracks.

Listener Judgment Filtering. Finally, we eliminated tags closely tied to listener preferences and judgments. Briefly put, we established predefined categories specifically designed to capture tags for exclusion, based on their semantic content. For example, we defined a category titled ‘*This track is great*’ and tags like ‘*Love it!*’ would semantically align with this category and be filtered out. To do so, we computed sentence-level embeddings for both the tags and the categories (augmented by the *NRC Lexicon*) using the SBERT model to capture their semantic content. We then matched each tag to its closest category using cosine similarity on their embeddings, removing tags that fell into any unwanted category.

4.2 Emotion Vector Attribution

The second phase of our approach relies on the *NRC Lexicon* to computationally associate emotion vectors with music tracks by relying solely on the acquired tags. We decided to represent words from the *Lexicon* as vectors w within the Plutchik emotion space, where $w = \sum_{i \in B} c_i e_i$ and c_i is a binary indicator denoting the absence or presence of the corresponding emotion.

Given that tags are assigned by individual listeners on music platforms, we can treat them as independent entities. This assumption enables us to first assign emotions to each unique tag in the dataset, and then derive the emotion vector of a track by combining the emotions of its associated tags.

² <https://huggingface.co/dslim/bert-base-NER>

³ <https://huggingface.co/cardiffnlp/twitter-roberta-base-sentiment-latest>

¹ <https://www.last.fm/>, <https://www.allmusic.com/>, <https://rateyourmusic.com/>

4.2.1 Assigning Emotions to Individual Tags

To infer the emotions connoted by a given tag, we first input both the tag and words from the *NRC Lexicon* into the **SBERT model**. This model generates embeddings for each, representing both the tag and words in the same semantic space.

We then perform **semantic search**, which involves retrieving the top- k entries $\{y_i\}_{i=1,\dots,k} \in Y$ of a corpus (NRC Lexicon) that are closest to a query x (the tag) by maximizing the cosine similarity on their embeddings, effectively identifying the words that are semantically similar to the tag: $y_i \in \arg \max_{y \in Y} \frac{x \cdot y}{\|x\| \|y\|}$.

Finally, a **weighted majority vote** is performed. This method involves directly selecting emotion vectors when a match with a high similarity score is found. If no such match exists, emotions with the highest consensus among a broader set are chosen.

Algorithm 1 Weighted Majority Vote

- 1: **Input:** Hyperparameters $\alpha_1, \alpha_2, \alpha_3, \beta$; tag embedding x ; embeddings, similarity scores, and emotion vectors from the top- k matches $\{(x_i, s_i, w_i) \mid s_u \geq s_v \text{ when } u < v\}_{i=1,\dots,k}$;
 - 2: **Output:** Emotion vector of the tag $v \in R^8$
 - 3: Initialize the set of chosen matches : $m \leftarrow \emptyset$.
 - 4: **if** $s_1 \geq \alpha_1$ **then**
 - 5: $m \leftarrow \{w_1\}$
 - 6: **else**
 - 7: $m \leftarrow \{w_i \mid s_i \geq \alpha_2\}$
 - 8: **if** $m = \emptyset$ **then**
 - 9: $m \leftarrow \{w_i \mid s_i \geq \alpha_3\}$
 - 10: $\mu \leftarrow \sum_{w_i \in m} s_i w_i \in R^8$
 - 11: $v \leftarrow (v_i)_i$ where $v_i = 1$ if $\mu_i > \beta$, 0 otherwise
 - 12: **Return** v
-

We conducted hyperparameter tuning using Grid Search to optimize the parameters $\alpha_1, \alpha_2, \alpha_3, \beta$ and k with the evaluation method outlined in Section 5.2. By selecting hyperparameters that maximize the F1-score between the original and inferred vectors, we ensured optimal accuracy in identifying the correct emotion vectors from words in the lexicon, considering both false positives and false negatives. The optimal values obtained were 0.95, 0.9, 0.5, 0.5, and 7, respectively.

4.2.2 Deriving the Emotion Vector for Each Track

Now that each tag is assigned an emotion vector with binary values indicating the presence or absence of each primary emotion, we can derive the emotion vector for each track. However, two issues must be addressed first. Tag occurrences should be normalized to ensure comparability across different sources; and intersubjective variability in music perception should be accounted for, since it can lead to differing tags among listeners and misleading inferred emotion vectors.

Tag Occurrences Normalization. We divide each occurrence by the maximum occurrence encountered within the source, resulting in normalized occurrences within the [0,1] range. For tags from *Rate Your Music*, where occurrences were not provided, we set their count to the average occurrence at the track level.

Tag Selection for Inter-rater Agreement. For each track, we select tags that exhibit good inter-rater agreement, estimated using the Intra-class Correlation Coefficient (ICC) with one-way random effects for absolute agreement [24]—a widely used metric for assessing inter-rater reliability when the same set of raters evaluates all subjects. In our approach, each emotion is treated as an individual "subject" and each tag as a "rater". The emotion vectors of each tag, weighted by their normalized occurrences, serve as ratings for the respective emotion.

To attain the acceptable threshold of 0.75 for inter-rater agreement (values between 0.75 and 0.90 indicate good reliability, according to [24]), we perform backward selection to iteratively eliminate conflicting tags. Starting with the initial set of tags for a given track, we remove the tag whose exclusion results in the highest ICC score. This process continues until the threshold is attained or only two tags remain.

Track Emotion Vector. We derive the emotion vector of a track by calculating the weighted average of the emotion vectors v_i from the p tags that demonstrated good inter-rater agreement. The weights α_i are set to the tags' normalized occurrences, thus giving more importance to emotions from prevalent tags.

$$v = \frac{1}{\sum_{i=1}^p \alpha_i} \sum_{i=1}^p \alpha_i w_i = \sum_{j \in B} \lambda_j e_j, \quad (2)$$

5. EVALUATION

5.1 Tag Extraction Method

To assess the reliability of our tag extraction method, we compared our tags to *human-generated* annotations from two crowdsourced MER datasets: *AMC Mirex* [7] and *Cal500* [9]. These datasets were selected for being the only ones to include emotion tags and share common tracks with our collection.

First, we calculate the percentage of common tags at the track level between each dataset and ours. Next, to assess the alignment between emotion tags, we derive emotion vectors of tracks from the two crowdsourced datasets using our method for emotion vector attribution (see Section 4.2), and then compare them with ours using semantic similarity.

	AMC		Cal500	
	mean	med.	mean	med.
Percentage of common tags for each track	56.0	100.0	3.24	0.0
Similarity score between emotion vectors	0.68	0.78	0.75	0.82

Table 2: Comparison of tags and emotion vectors

Comparing the resulting tags either directly or via the emotions they convey, our findings demonstrate that our method's results align well with human-generated annotations. In the *AMC* dataset we observed a strong direct match with our tags, with an average tag overlap of 56% at the track level ([41.69, 70.81] 95% CI) and a median reaching 100%. Considering the structure of the *AMC* dataset,

whose tracks are usually assigned only one tag, this means that for 56% of the tracks the *AMC* tag is contained in the set of tags we collected from music platforms. For *Cal500*, despite a low tag overlap of 3.24%, we observed significant alignment, with a mean similarity score of 0.75 between the derived emotion vectors. Note that the lower similarity score observed in *AMC* (0.68) may be attributed to its limited number of tags—with an average of one per track—compared to ours (~8 tags per track) and that of *Cal500* (~15 tags per track).

5.2 Tag Emotion Assignment Method

To assess the reliability of our method for assigning emotions to individual tags (see Section 4.2.1), we applied the same technique to the words in the NRC Lexicon. By treating each word as a ‘query’ and using the NRC Lexicon, excluding the query word itself, as the ‘corpus’, we derive emotion vectors for each word and compare them with the original vectors provided by the NRC Lexicon.

We achieved an average accuracy of 84% in identifying emotions represented by a given tag, with balanced scores across emotions. Joy was the most accurately identified emotion (93%), while fear had the lowest score (76%). Our F1-score was 77%—an expected result, as our method prioritizes emotions that align across all matched words, leading to a higher number of false negatives and thus lower recall. Nonetheless, the method’s ability to generally identify emotions demonstrates its overall effectiveness.

6. DISCUSSION

In this section, we discuss the strengths and weaknesses of our approach for generating emotion-labeled datasets.

6.1 A Dual-Use Model and a Reproducible Framework

The proposed framework, methodically divided into two distinct phases, facilitates the creation of synthetic datasets suitable for both Music Emotion Recognition (with emotion vectors) and auto-tagging tasks (with tags retrieved from music platforms). Its flexibility renders it applicable to any existing dataset that includes tags on music tracks, therefore allowing researchers to create their own emotion-labeled dataset⁴.

Additionally, our work can be easily extended to incorporate future resources similar to the NRC Lexicon, albeit based on other emotion models (GEMS etc.), as such resources become available. Since this approach only requires a mapping from English words to emotion labels, collecting these resources is significantly easier than obtaining direct emotion annotations on music excerpts.

6.2 Large-Scale Data Collection with Emphasis on Data Quality

Our method for extracting listener-generated textual data from music platforms overcomes the usual limitations

of data collection—including time, cost and feasibility constraints—through crowdsourced experiments and therefore enables data collection on a larger scale. Our final collection contains 5,892 emotion-labelled tracks, more than twice the size of the hitherto largest emotion model-based dataset of 2,648 tracks (*NTWICM* [11]).

Notably, specific attention was paid to retain only relevant tags, removing those that lack overt emotional significance, represent value judgments, or describe musical genres. Consequently, our dataset underwent significant refinement, with only a small percentage of total and unique tags retained (4.8% and 1.1%, respectively), enhancing its quality while maintaining its diversity (see Table 3). We actually ended up with 1,013 unique emotion tags, significantly more than the 157 emotion labels in the *MER* dataset with the largest number of labels to our knowledge (*Cal500*). Tag distribution across the dataset and within each source is presented in Figure 2, where the size of each tag reflects its frequency, taking into account its occurrence.

	Before	After
Tags across all tracks	1,007,847	48,737 (4.8%)
Unique tags	90,699	1,013 (1.1%)
Unique tracks	12,515	5,892 (47.1%)

Table 3: Data filtering overview, before pre-processing, after pre-processing

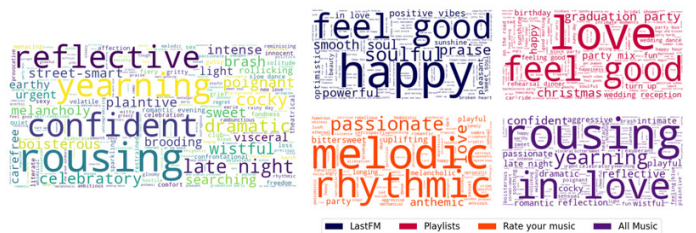


Figure 2: Tag Frequency Across Dataset and Sources

6.3 Reliance on Music Platforms

Our proposed framework relies on the availability of data on music platforms and may face challenges due to the under-representation of certain musical genres. Indeed, the final dataset exhibits a significant imbalance across genres, with world music and classical music represented only by 29 and 24 tagged tracks. We nonetheless pursued the creation of a balanced dataset by selecting the 280 most popular tracks for the top 15 most-represented genres, yielding 4,200 tracks in total⁵.

The tag frequencies (see Figure 2) reveal a prevalence of certain tags, with positive-emotions particularly prominent. However, it is uncertain whether this reflects listener preferences, a wider music industry trend, or a tendency of listeners to engage with music platforms when in a positive mood. The latter possibility could potentially introduce bias into our results.

⁴ Python code to derive emotion vectors from a set of tags is provided: <https://github.com/joanne-affolter/PlayMood>

⁵ The dataset and its balanced version are made public: <https://github.com/joanne-affolter/PlayMood>

6.4 Emotion Association: A Focus on Explainability

Some critics might argue that our methodology for generating emotion-labeled datasets could introduce bias when training Music Emotion Recognition (MER) systems, as it relies on synthetic data. However, the use of a direct mapping from tags to emotions using a crowd-sourced Lexicon aims to ensure model explainability and interpretability. We deliberately chose not to use machine learning models to predict word emotions, opting instead for a resource curated by humans. However, we acknowledge that the strong reliance on the NRC Lexicon renders our work subject to the latter’s limitations, including socio-cultural biases [25], the possibility of incorrect, nonsensical, or pejorative entries due to human error—inevitable with large-scale annotations—and potential ambiguities due to a lack of context in the lexicon [26].

6.5 Towards More Generalizable Findings

In addition to significantly reducing the need for human-generated annotations, our synthetic dataset in fact leverages the size and diversity of social music platforms. For instance, it features a larger average number of tags per track (mean: 7.52, min: 1, max: 171) compared to crowd-sourced datasets, which typically rely on a few tags (on average 1.62 for *MTG-MT* and 1.01 for *AMC Mixex*). This variety enables a broader range of interpretations and a more nuanced evaluation of listener feedback, although it may also present challenges in identifying emotions. Furthermore, agreement among listeners on music platforms tends to be relatively high, as indicated by the frequency of tag occurrences at the track level (mean: 6.01, min: 1, max: 200). In contrast to crowdsourced studies that generally require agreement between a few annotators, our method has an intrinsic potential for more robust and generalizable findings thanks to the higher number of listeners involved in the tagging process.

6.6 Emotion Modeling: Paving the Way for Future Research

By grounding our method on a set of eight primary emotions, we offer an intuitive alternative to the VA framework. Meanwhile, by using a continuous vector representation in the Plutchik emotion space, our framework is also able to capture subtle emotional nuances of music tracks, as illustrated by the emotional profiles in Figure 3⁶.

Notably, we found that as the number of tags increased, the emotional spectra of a track became more complex, involving a wider variety of emotions with varying intensities. One may wonder whether to consider feedback from all listeners, resulting in more intricate emotion representations, or retain the best-aligned tags alone, thereby increasing consistency at the risk of missing individual nuances. In this work, we opted for mutually consistent emotion vectors with an emphasis on inter-rater agreement. By

filtering out tags that represent contrasting emotions, we negotiated, on the one hand, the intersubjective variability of music perception and, on the other hand, its socially communicative potential by selecting a significant number of tags with high agreement, effectively producing complex emotion representations validated by the majority. We thus achieved an average ICC score of 0.76 for the emotion ratings associated with the selected tags for each track, indicating *good reliability* according to [24], compared to the initial score of 0.52, which suggested *moderate reliability*. It is noteworthy that, despite the filtering process, the average number of tags per track decreased only slightly from 7.52 to 6.55, demonstrating that our dataset still reflects the diversity of its participant pool.

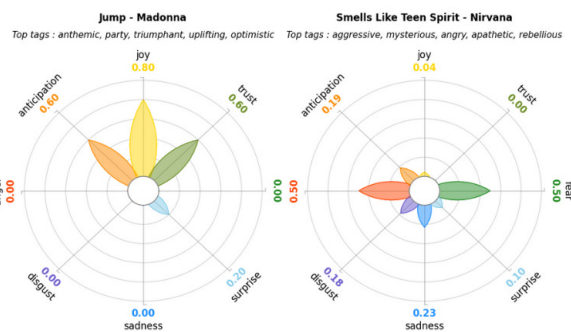


Figure 3: Visualization of tracks’ emotions.

7. CONCLUSION

Our investigation introduces a novel approach to Music Emotion Recognition (MER) by benefiting from large-scale, listener-generated tagging alongside an original application of Plutchik’s emotion model. With this work we aimed not only to address the scarcity of annotated datasets in the MER domain, but also to challenge traditional paradigms of music emotion by adopting an intensely empirical, psychological model-based framework. Through meticulous data collection and cleaning, we generated a dataset that surpasses existing collections in size and diversity, while maintaining a high degree of alignment with human-generated annotations. We believe that the integration of Natural Language Processing techniques in the semantic analysis of music tags is a methodological innovation that may effectively transpose the problem from audio to the textual domain. Furthermore, our approach’s dual utility in MER and Auto-Tagging tasks demonstrates its versatility and potential for wide-ranging applications in Music Information Retrieval. By narrowing the gap between psychological emotion theories and computational music analysis, we pave the way for future research endeavors aimed at enriching our understanding of listeners’ emotional engagement with music.

⁶ A notebook for visualizing the emotional profiles across all tracks in our collection is available: <https://github.com/joanne-affolter/PlayMood>

8. ETHICS STATEMENT

Ethical Considerations in Data Handling It is important to note that our dataset does not contain any listener-specific information; our research involved the analysis of publicly available data only. By design, our approach prevents any direct links to individual listeners within the dataset, mitigating concerns around privacy and data security.

Addressing Societal and Cultural Considerations

The diversity of music across cultures presents a challenge for Music Information Retrieval technologies, which should strive to prevent cultural homogenization in interpretive systems. Despite efforts to include a wide range of genres and styles, our dataset may not fully capture the breadth of global musical diversity. Additionally, the platforms from which data was sourced may primarily serve specific demographics, potentially biasing our dataset towards the musical preferences and emotional expressions of a particular segment of the global population. Future research should prioritize the collection of tags from a more culturally diverse set of sources, work towards the further mitigation of such biases, and enhance the inclusivity of MIR technologies.

9. ACKNOWLEDGMENTS

I would like to sincerely thank I. Rammos for his invaluable supervision throughout the entire project and his significant contributions to the review and correction of the paper. His expertise and support greatly enhanced the quality of this work.

10. REFERENCES

- [1] R. Plutchik, *A general psychoevolutionary theory of emotion*, R. Plutchik and H. Kellerman, Eds. New York: Academic Press, 1980, vol. 1.
- [2] M. Barthelet, G. Fazekas, and M. Sandler, "Music emotion recognition: From content- to context-based models," in *Proceedings of the International Society for Music Information Retrieval Conference*, Virtual, 2021, pp. 1–7.
- [3] D. Han, Y. Kong, J. Han, and G. Wang, "A survey of music emotion recognition," *Frontiers of Computer Science*, vol. 16, no. 6, p. 166335, 2022.
- [4] R. Yuan, Y. Ma, Y. Li, G. Zhang, X. Chen, H. Yin, L. Zhuo, Y. Liu, J. Huang, Z. Tian, B. Deng, N. Wang, C. Lin, E. Benetos, A. Ragni, N. Gyenge, R. Dannenberg, W. Chen, G. Xia, W. Xue, S. Liu, S. Wang, R. Liu, Y. Guo, and J. Fu, "Marble: Music audio representation benchmark for universal evaluation," in *Proceedings of the Neural Information Processing Systems (NeurIPS)*, 2023, camera-ready version. arXiv:2306.10548 [cs.LG]. DOI: <https://doi.org/10.48550/arXiv.2306.10548>.
- [5] M. Soleymani, M. N. Caro, E. M. Schmidt, C.-Y. Sha, and Y.-H. Yang, "1000 songs for emotional analysis of music," in *Proceedings of the 2nd ACM international workshop on Crowdsourcing for multimedia*, 2013, pp. 1–6.
- [6] D. Bogdanov, M. Won, P. Tovstogan, A. Porter, and X. Serra, "The mtg-jamendo dataset for automatic music tagging," in *International Conference on Machine Learning (ICML)*, 2019.
- [7] X. Hu, J. S. Downie, C. Laurier, M. Bay, and A. F. Ehmann, "The 2007 mirex audio mood classification task: Lessons learned," in *Proceedings of the International Society for Music Information Retrieval Conference*, 2008, pp. 462–467. [Online]. Available: https://ismir2008.ismir.net/papers/ISMIR2008_263.pdf
- [8] H. Strauss, J. Vigl, P. Jacobsen *et al.*, "The emotion-to-music mapping atlas (emma): A systematically organized online database of emotionally evocative music excerpts," *Behavioral Research*, vol. 56, pp. 3560–3577, 2024. [Online]. Available: <https://doi.org/10.3758/s13428-024-02336-0>
- [9] D. Turnbull, L. Barrington, D. Torres, and G. Lanckriet. (Accessed 2024) Cal500. [Online]. Available: <http://calab1.ucsd.edu/~datasets/cal500/>
- [10] Y. Kim, E. Schmidt, and L. Emelle, "Moodswings," Accessed 2024, offline.
- [11] B. Schuller, J. Dorfner, and R. Gerhard, "Now that's what i call music," Accessed 2024. [Online]. Available: <http://openaudio.eu/NTWICM-Mood-Annotation.arff>
- [12] T. Eerola and J. K. Vuoskoski, "Soundtracks," Accessed 2024. [Online]. Available: <https://osf.io/p6vkg/>
- [13] S. Koelstra, C. Muehl, M. Soleymani, J.-S. Lee, A. Yazdani, T. Ebrahimi, T. Pun, A. Nijholt, and I. Patras, "Deap," Accessed 2024. [Online]. Available: <http://www.eecs.qmul.ac.uk/mmv/datasets/deap/>
- [14] Y.-A. Chen, Y.-H. Yang, J.-C. Wang, and H.-H. Chen, "Amg1608," Accessed 2024. [Online]. Available: <https://amg1608.blogspot.com/>
- [15] A. Aljanaki, F. Wiering, and R. Veltkamp, "Emotify," Accessed 2024. [Online]. Available: <http://www2.projects.science.uu.nl/memotion/emotifydata/>
- [16] M. Pesek, G. Strle, A. Kavčič, and M. Marolt, "Moodo," Accessed 2024. [Online]. Available: <http://moodo.musiclab.si>
- [17] R. Panda, R. Malheiro, and R. P. Paiva, "4q emotion dataset," Accessed 2024. [Online]. Available: <http://mir.dei.uc.pt/downloads.html>
- [18] K. Zhang, H. Zhang, S. Li, C. Yang, and L. Sun, "Pmemo," Accessed 2024. [Online]. Available: <https://github.com/HuiZhangDB/PMemo>

- [19] T. Eerola and J. K. Vuoskoski, “A review of music and emotion studies: Approaches, emotion models, and stimuli,” *Music Perception: An Interdisciplinary Journal*, vol. 30, no. 3, pp. 307–340, 2013.
- [20] S. M. Mohammad and P. Turney, “NRC Word-Emotion Association Lexicon (aka EmoLex),” Non-Commercial Use Only — Research or Educational, Released 2011. [Online]. Available: <http://saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm>
- [21] N. Reimers and I. Gurevych, “Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks,” *arXiv preprint*, 2019. [Online]. Available: <https://doi.org/10.48550/arXiv.1908.10084>
- [22] Spotify Million Playlist Dataset Challenge. Accessed 2024. [Online]. Available: <https://www.aicrowd.com/challenges/spotify-million-playlist-dataset-challenge>
- [23] G. Kreutz, U. Ott, D. Teichmann, P. Osawa, and D. Vaitl, “Using music to induce emotions: Influences of musical preference and absorption,” *Psychology of music*, vol. 36, no. 1, pp. 101–126, 2008.
- [24] T. K. Koo and M. Y. Li, “A guideline of selecting and reporting intraclass correlation coefficients for reliability research,” *Journal of Chiropractic Medicine*, vol. 15, no. 2, pp. 155–163, 2016.
- [25] S. Zad, J. Jimenez, and M. Finlayson, “Hell hath no fury? correcting bias in the nrc emotion lexicon,” in *Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021)*, 2021.
- [26] S. M. Mohammad, “Practical and ethical considerations in the effective use of emotion and sentiment lexicons,” 2020. [Online]. Available: <https://ar5iv.labs.arxiv.org/html/2011.03492>

PICOGEN2: PIANO COVER GENERATION WITH TRANSFER LEARNING APPROACH AND WEAKLY ALIGNED DATA

Chih-Pin Tan^{1,2} Hsin Ai¹ Yi-Hsin Chang¹ Shuen-Huei Guan² Yi-Hsuan Yang¹

¹ Graduate Institute of Communication Engineering, National Taiwan University, Taiwan

² KKCompany Technologies

tanchihpin0517@gmail.com, yhyangtw@ntu.edu.tw

ABSTRACT

Piano cover generation aims to create a piano cover from a pop song. Existing approaches mainly employ supervised learning and the training demands strongly-aligned and paired song-to-piano data, which is built by remapping piano notes to song audio. This would, however, result in the loss of piano information and accordingly cause inconsistencies between the original and remapped piano versions. To overcome this limitation, we propose a transfer learning approach that pre-trains our model on piano-only data and fine-tunes it on weakly-aligned paired data constructed without note remapping. During pre-training, to guide the model to learn piano composition concepts instead of merely transcribing audio, we use an existing lead sheet transcription model as the encoder to extract high-level features from the piano recordings. The pre-trained model is then fine-tuned on the paired song-piano data to transfer the learned composition knowledge to the pop song domain. Our evaluation shows that this training strategy enables our model, named PiCoGen2, to attain high-quality results, outperforming baselines on both objective and subjective metrics across five pop genres.

1. INTRODUCTION

Piano cover generation, which involves recreating or arranging an existing music piece as a new piano version, is popular within music-creative communities and the music production industry. On media sharing sites like YouTube, piano cover creators often have lots of subscribers. Additionally, many music producers create and distribute piano arrangements on music streaming platforms.

Attempts have been made in the field of music information retrieval (MIR) to automatically generate piano covers from existing musical pieces. Takamori *et al.* [1] proposed a regression method to generate piano reductions, which can be considered simplified versions of piano covers, using acoustic features and structural analysis of the

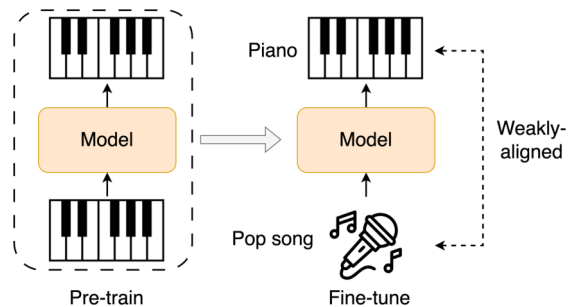


Figure 1. The proposed model is trained with two stages: firstly pre-trained on piano-only data and then fine-tuned on the weakly-aligned song-to-piano pairs.

input music. With the recent surge in deep learning, Choi *et al.* [2] introduced a model named Pop2Piano that tackles piano cover generation by leveraging the concept of piano transcription and employing the MT3 architecture [3], originally designed for transcription, as their model backbone. They collected pop songs and the corresponding piano covers from the Internet, and built a song-piano *synchronized* dataset by “remapping” the piano notes to the song audio with a warping algorithm (thereby modifies, or warps, the piano cover). The algorithm entails evaluating the similarity between the pitch contour of the vocal signal extracted from the song audio with the top line of the piano MIDI. They then trained the model with the synchronized data, guiding the model to learn the pitch and onset/offset timing of each note in the generated piano cover.

However, as shown in Table 1, the statistics in the ratio of audio length difference and tempo difference between the original songs and original piano covers (i.e., before note-remapping) they collected¹ show that a piano cover and its original song are not perfectly aligned to each other (for otherwise the difference ratio would be equal to 1.00). This indicates that the tasks cover generation and transcription are inherently different, and that forcing a piano cover to be synchronized with its original song may be inappropriate. Actually, we notice that the note-remapping process of Pop2Piano—i.e., adjusting piano note timing according to the time mapping function obtained by synchronizing piano notes to the song audio—breaks the relation of original piano notes and thereby incurs loss of piano informa-



© C.-P. Tan, H. Ai, Y.-H. Chang, S.-H. Guan, and Y.-H. Yang. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** C.-P. Tan, H. Ai, Y.-H. Chang, S.-H. Guan, and Y.-H. Yang, “PiCoGen2: Piano cover generation with transfer learning approach and weakly aligned data”, in *Proc. of the 25th Int. Society for Music Information Retrieval Conf.*, San Francisco, United States, 2024.

¹ https://github.com/sweetcocoa/pop2piano/blob/main/train_dataset.csv

duration deviation	tempo deviation	IOI deviation
1.10 ± 0.12	1.16 ± 0.25	1.14 ± 0.17

Table 1. The first two statistics contrast the original songs with their original piano covers (i.e., no note-remapping) in the Pop2Piano dataset [2], evaluating the length of the **duration** (in seconds) of the longer one divided by that of the shorter one, and similarly the deviation ratio in **BPM**. The last statistic is similarly the deviation ratio in terms of the average inter-onset intervals (**IOIs**; in seconds), but between the original & adjusted (synchronized) piano covers.

tion. Moreover, from a musical perspective, the way human creates piano covers is by nature different from the way human transcribes music. For cover generation, musicians may firstly analyze the original song in terms of aspects such as melody, chord progression and rhythm section, then decide how to interpret the original song with their composition knowledge, and finally make the piano cover based on the piano performance techniques.

Inspired by the process of human composition for piano cover songs, we propose in this paper a novel approach for piano cover generation by involving the concept of transfer learning [4]. Instead of relying on the *strongly-aligned* pairs [5] that necessitates note-remapping, we use *weakly-aligned* data with the correspondence in “beat” level between song-piano pairs. This approach incurs no rhythmic distortion of the piano covers, retaining their musical quality. Besides, to mitigate the inaccuracy of data alignment, the model is pre-trained on piano-only data to learn the concept of piano performance first, and then fine-tuned on the weakly-aligned paired data to learn the conversion of song to piano, as shown in Figure 1. We also employ a prior model SheetSage [6], pre-trained for lead sheet transcription, as an encoder component that helps our model learn high-level musical concepts for cover generation.

We compare the proposed model, named “PiCoGen2”, against other baselines with objective and subjective measures, validating the effectiveness of the weak-alignment method for pairing and the two-step training strategy. We share source code and audio samples at a project page.²

2. BACKGROUND

Piano arrangement, i.e., the process of reconstructing and reconceptualizing a piece, is related to various conditional music generation tasks, including lead sheet³-conditioned accompaniment generation, transcription and reorchestration, and piano reduction [7–9]. Beyond arrangement, piano cover generation involves creating new musical elements and modifying the original elements via improvisation, tempo changes, stylistic shifts, etc. We briefly review some related topics below.

Symbolic-domain music generation is about generating music in a symbolic form such as pianorolls [10] and dis-

crete MIDI- (Musical Instrument Digital Interface) [11] or REMI-like tokens [12–15], rather than audio signals. The task encompasses unconditional generation (i.e., from-scratch generation) and conditional generation. While the goal of piano cover generation is to generate piano audio given a song audio input, we can treat it as a conditional symbolic music generation task, for we can generate piano in the MIDI domain first, and then use off-the-shelf high-quality piano synthesizers to convert it into audio.

Automatic music transcription (AMT), which aims to precisely transcribe music content from audio signals into a symbolic representation over time, is also related to piano cover generation. AMT tasks can be categorized based on the completeness of information captured from the input audio. One category of AMT tasks aims to capture all music content presenting in the audio, such as automatic piano transcription [3, 16–20]. These methods transcribe the complete polyphonic piano performance from the audio signal. Another category focuses on transcribing a reduced representation of the input, like melody transcription [21, 22] and lead sheet transcription [6, 23, 24]. These tasks extract only the lead melody line and chord progressions, representing a sparse subset of the full musical content. Piano cover generation also requires the exploration of music content reduction and additionally relies on generative modeling conditioned on the reduced representation. For example, Pop2Piano uses MT3 [3] as its backbone to convert audio features into a symbolic piano performance representation. However, following the paradigm of transcription approaches, Pop2Piano requires paired data consisting of pop songs and their corresponding temporally-synchronized piano cover.

Transfer learning is generally consider as the concept of adopting the model to the target domain by re-using parameters that are trained on a source domain, thereby transferring the knowledge between the domains [25]. There have been several works on transfer learning in the field of MIR, e.g., music classification [26–28] and music recommendation [29, 30]. However, to our best knowledge, little attempts have been made to apply transfer learning to the task of cover song generation.

Besides Pop2Piano [2], this work is also closely related to PiCoGen [31], an early version of the current work. We explore the two-stage training strategy for piano cover generation for the first time there. However, in PiCoGen we use discrete symbolic lead sheet as the intermediate representation, instead of continuous conditions supplied by an encoder as done here (see Section 3.2). We note that the sampling process of lead sheet extraction in PiCoGen might loss musical information such as instrumentation and vibes of the input audio. Moreover, we do not explore the idea of transfer learning (Section 3.3) there.⁴

The work of Wang *et al.* [32] is also related, for they deal with the similar problem of converting audio signals

² <https://tanchihpin0517.github.io/PiCoGen/>

³ A music notation consisting of lead melody and chord progression.

⁴ As the previous work [31] was also under review at the time we submitted the current paper, we did not empirically compare PiCoGen and PiCoGen2 in the experiments here. Instead, we provide examples of their generation results for the same input songs on the demo page, which should demonstrate that PiCoGen2 works better than PiCoGen.

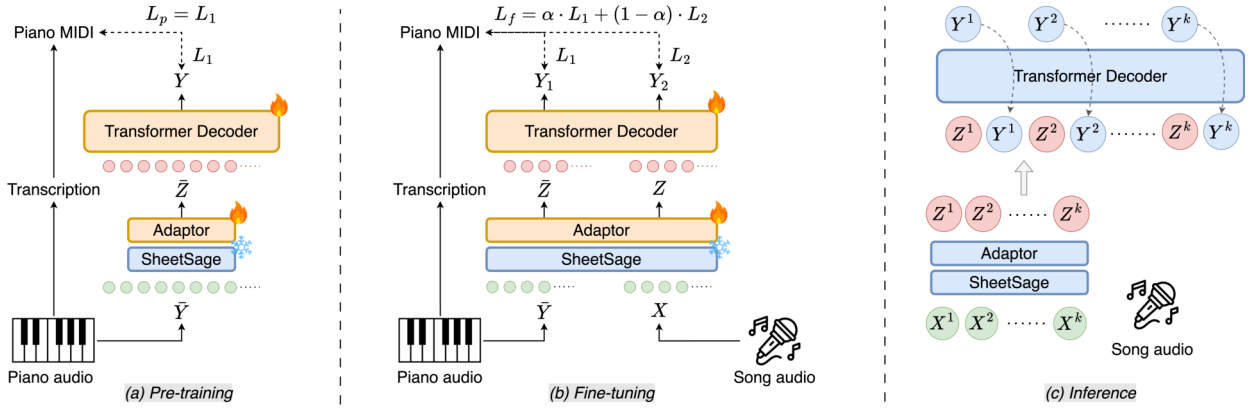


Figure 2. A diagram of the proposed model, PiCoGen2. The fire and snowflake symbols indicate the trainable and frozen parts. For example, the parameters for SheetSage [6], a model pre-trained for lead sheet transcription, are always frozen.

into piano MIDI performances. However, they apply a piano transcription prior and thus using strongly-aligned data as Pop2Piano [2], and they employ a more sophisticated disentanglement-based method to get an intermediate representation. Moreover, they assume that the vocal of the input audio has been removed beforehand, thus actually generating a piano backing track rather than a piano cover.

3. METHODOLOGY

Viewing piano cover generation as a conditional symbolic music generation task, we formulate it as a sequence-to-sequence problem. The objective is to generate a sequence of symbolic tokens Y representing the piano performance, conditioned on the input audio X of the original song.

3.1 Weakly-Aligned Data

In Pop2Piano, Choi *et al.* [2] propose a data preprocessing algorithm to synchronize the piano MIDI to the song audio. They utilize SyncToolBox [33] to analyze the chroma features of two audio segments to obtain a warping path of mapping the time from the piano performance to the song audio. Based on the analysis, they adjust the timing of notes transcribed from the piano performance by using a linear mapping function calculated from the temporal warping information. These remapped notes is then quantized to align with the beat locations of the song audio. However, the rhythmic distortion caused by note-remapping is practically unavoidable, even disregarding the inaccuracy of the synchronization process. The chroma feature only reflects a rough overall alignment between the piano performance and song audio which cannot precisely describe the nuanced amount of timing shift for each individual note. This is evident when examining the changes in the inter-onset intervals (IOIs) between the original piano notes and the remapped version, shown in Table 1.

To avoid the rhythmic distortion of note-remapping, we propose a weak-alignment approach that does not change the timing of piano notes. The idea is to let the alignment rely on only the *beats* of each song-to-piano pair. We construct the time mapping function F_{time} by computing the

warping path for the audio pair like the way of Pop2Piano. Given a time of piano performance t_p , the function outputs the corresponding time of song audio $t_s = F_{\text{time}}(t_p)$ according to the temporal warping information. Specifically, we detect the beat locations with Beat Transformer [34] to get the beat times $Q^p = [q_1^p, \dots, q_{l_p}^p]$ of the piano performance and $Q^s = [q_1^s, \dots, q_{l_s}^s]$ of the song audio, where l_p and l_s denote the number of beats of each of them. Then we define an aligning function F_{beat} as:

$$F_{\text{beat}}(i) = \arg \min_j (F_{\text{time}}(q_i^p) - q_j^s). \quad (1)$$

For any beat index $i \in [1, \dots, l_p]$ of the piano performance, the aligning function outputs the corresponding beat index $j \in [1, \dots, l_s]$ of the song audio, and q_j^s is the nearest beat time to $F_{\text{time}}(q_i^p)$. We consider a song-piano pair to be weakly-aligned if the correspondence between them is determined by F_{beat} . See the project page for an illustration.

3.2 Model

An aerial view of our model is depicted in Figure 2. We employ a decoder-only Transformer to accept an input sequence bundling condition X (song audio) and target Y (piano performance) together, and generates the output tokens for Y autoregressively. This approach of providing both the condition and target as a bundled input sequence to the Transformer has been applied in previous studies [13, 14, 35] and has shown success in better informing the model of the temporal correspondence between the condition and desired output. We divide Y into *bars* with the detected beat information and get $Y = [Y^1, \dots, Y^{B_p}]$, where B_p is the number of bars in the piano cover, and there exists an song audio sequence $X = [X^1, \dots, X^{B_p}]$ for Y , where each sub-sequence X^k is weakly aligned to Y^k . We then rearrange them with an interleaving form and train the decoder with the bar-wise mix $S = [X^1, Y^1, \dots, X^{B_p}, Y^{B_p}]$. The decoder model would learn to generate k -th bar of piano performance Y^k depending on (i.e., can attend to) the current and preceding sub-sequences of song audio $[X^1, \dots, X^k]$ and the preceding sub-sequences of piano performance $[Y^1, \dots, Y^{k-1}]$.

To reduce the sequence length of X and extract better musical information, we employ a prior audio encoder to transform X into an intermediate representation Z . Different from those works which use Mel-spectrograms [3] or audio codecs [36] for Z , we use SheetSage [6], which is trained for lead sheet transcription, cascaded with a neural adapter as the prior audio encoder. We consider the output embeddings of SheetSage more suitable for representing the input, since they carry information of musical elements connecting a cover with the original song, such as melody, chords and vibes. With the prior encoder, the song audio $[X^1, \dots, X^{B_p}]$ is transformed into a sequence of latent embeddings $[Z^1, \dots, Z^{B_p}]$ before being passed to the decoder, yielding the input sequence $[Z^1, Y^1, \dots, Z^{B_p}, Y^{B_p}]$ of the decoder, as illustrated in Figure 2c.

3.3 Transfer Learning

While the weak-alignment approach eliminates inner temporal distortions for piano performance, there can still be alignment errors between the piano segments and their corresponding song segments. This is because a piano cover is not guaranteed, in the beat level, to have a strict one-to-one mapping with the original song.

To abate such alignment errors, we propose a transfer learning-based training strategy, dividing the training into two steps: pre-training (Figure 2a) and fine-tuning (Figure 2b). In the *pre-training* stage, we train the model with an input sequence $\bar{S} = [\bar{Y}^1, Y^1, \dots, \bar{Y}^{B_p}, Y^{B_p}]$ where \bar{Y} is the original piano audio recording of the symbolic piano tokens Y . The same as the song audio, the original recording \bar{Y} is encoded to an inter-representation \bar{Z} by the prior encoder. We expect the model to learn to generate piano performances Y with high-level musical features extracted by SheetSage from the piano audio \bar{Y} , rather than merely detecting note onsets/offsets like in a piano transcription task. Importantly, there will be no alignment errors between Y and \bar{Y} , ensuring that the model can firstly learn the complete concept of piano composition and generation in the pre-training stage, without being impeded by cross-domain alignment issues.

In the *fine-tuning* stage, we train the model with the mixture of \bar{S} and S . Following [36–38], we train the model with the objective of minimizing the cross entropy loss on the tokens of piano performance Y . Let L_1 and L_2 stand for the cross entropy losses for \bar{S} and S , respectively. The loss L_p in the pre-training stage and the loss L_f in the fine-tuning stage can be written as:

$$\begin{aligned} L_p &= L_1, \\ L_f &= \alpha \cdot L_1 + (1 - \alpha) \cdot L_2, \end{aligned} \quad (2)$$

where α is the weighting factor determining the proportion of losses contributed from \bar{S} and S during fine-tuning. We expect that α helps the model retain the knowledge about piano performance learned from the pre-training stage.

3.4 Data Representation

For the piano performance sequences Y , we adopt a modified version of the REMI token representation [12], which

has been shown to work well for modeling pop piano. Our representation consists of 7 token classes. `Spec` contains special tokens such as `[bos]` (beginning-of-sentence) and `[ss]` (song-start) for controlling the model behavior. `Bar` indicates the property of each bars. `Position`, `Chord` and `Tempo` are metric-related tokens for 16th-note offsets within bars, chord changes (11 roots \times 12 qualities), and tempo changes (64 levels). `Pitch`, `Duration` and `Velocity` are note-related tokens for note pitches (A0 to C8), durations (1 to 32 16th-notes), and note velocities (32 levels). There are in total 428 tokens in the vocabulary. In our implementation, `[Bar_start]` and `[Bar_end]` always occur at the start and end of each bar in the input sequence S and \bar{S} .

4. EVALUATION

4.1 Dataset

We follow the instructions provided in the Pop2Piano source code to rebuild the training dataset, collecting 5,844 pairs of pop songs and their corresponding piano covers from the Internet. We filter out song pairs with a melody chroma accuracy (MCA) [39] lower than 0.05 or an audio length difference exceeding 15%, leaving 5,503 remaining pairs. In the pre-training stage, all the piano performances from these remaining pairs are used for training. In the fine-tuning stage, we remove invalid bars from the piano performances where the first and last beats of a bar were mapped to the same beat of the original song by the mapping function F_{time} . Around 50% of the bars are removed from the piano performances accordingly. We note that the large number of such invalid bars implies the alignment algorithm of Pop2Piano [2] may not be robust enough and future work can be done to study this.

For objective and subjective evaluations, we collect additional 95 song-to-piano pairs from the Internet, containing 19 Chinese Pop (**Cpop**), 20 Korean Pop (**Kpop**), 16 Japanese Pop (**Jpop**), 20 Anime Song (**Anime**), 20 Western Pop (**Western**) pairs. All the songs contain vocals. We share the URLs of these songs at the project page.

4.2 Experiment Setup

We implement PiCoGen2 using GPT-NeoX [40] as the piano token decoder and SheetSage [6] cascaded with an adapter network as the song audio encoder. The decoder consists of 8 layers, each with 8 attention heads. The adapter is a 4-layer Transformer encoder with 8 attention heads per layer. Our full model has approximately 39M learnable parameters, not counting the SheetSage part for we use it as is with its parameters frozen.

There are 2 ablations compared in the experiment, both of them sharing the same architecture as our full model, but one ablation (Ablation 1) is trained on song-to-piano data *without pre-training*, and the other ablation (Ablation 2) is trained on piano-only data (i.e., *without fine-tuning*). For baselines, besides Pop2Piano, we also include the piano transcription model by Kong *et al.* [20] to validate the effectiveness of the encoder component in our model.

Model	objective evaluation			subjective evaluation ($\in [1, 5]$)		
	$MCA \uparrow$	$GS \uparrow$	$H_4 \downarrow$	$OVL \uparrow$	$SI \uparrow$	$FL \uparrow$
Pop2Piano [2]	0.42 \pm 0.07	0.86 \pm 0.09	2.46 \pm 0.18	2.71 \pm 0.98	2.63 \pm 1.01	2.72 \pm 1.1
Transcription [20]	0.19 \pm 0.06	0.67 \pm 0.09	2.78 \pm 0.30	1.48 \pm 0.74	1.69 \pm 0.88	1.45 \pm 0.71
Proposed (PiCoGen2)	0.17 \pm 0.06	0.84 \pm 0.06	2.46 \pm 0.22	3.48 \pm 0.93	3.55 \pm 1.06	3.66 \pm 1.02
- Ablation 1 (w/o pre-training)	0.16 \pm 0.05	0.87 \pm 0.06	2.45 \pm 0.23	3.09 \pm 1.03	2.96 \pm 1.02	3.22 \pm 1.09
- Ablation 2 (w/o fine-tuning)	0.15 \pm 0.05	0.81 \pm 0.06	2.57 \pm 0.19	3.09 \pm 1.02	3.30 \pm 1.07	3.08 \pm 1.16
Human	0.16 \pm 0.06	0.81 \pm 0.06	2.59 \pm 0.18	4.30 \pm 0.87	4.23 \pm 0.95	4.33 \pm 0.9

Table 2. The results of objective evaluations and the MOS of the subjective study (\uparrow/\downarrow : the higher/lower the better).

We train the models with Adam optimizer, learning rate $1e-4$, batch size 4 and segment length 1,024. The full model is pre-trained for 100K steps on the piano-only data, and then fine-tuned for an additional 70K steps on the song-to-piano paired data. Ablation 1 is trained from scratch for 100K steps directly on the paired data. Ablation 2 is trained for 50K steps only on the piano-only data, without any exposure to the song-to-piano pairs. During the fine-tuning stage for the full model, we tune the weighting factor α that controls the balance between the piano-only loss and song-to-piano loss, and find that the model achieved the best performance when α is set to 0.25.

For the objective and subjective evaluations, all models are used to generate piano covers of the 95 testing songs (cf. Section 4.1). To eliminate the bias caused by the varying quality of piano recordings, the ground truth human piano performances are first transcribed into MIDI note sequences. These MIDI sequences are then synthesized back into audio using the same FluidSynth-based MIDI synthesizer [41] employed for the model outputs.

4.3 Objective Metrics

We adopt the following existing metrics to assess the quality of the generated piano covers from different aspects, including similarity to the original song and coherence of the piano performance itself.

- **Melody Chroma Accuracy (MCA)** [39] evaluates the similarity between two monophonic melody sequences. The melody line plays a crucial role in deciding whether a song cover resembles the original song. Following Pop2Piano [2], we compute the MCA between the vocals extracted by Spleeter [42] from the test song audio, and the top melodic line extracted from the generated piano cover MIDI using the skyline algorithm [43].
- **Pitch Class histogram Entropy (H_4)** [37] evaluates the harmonic diversity of a musical segment by computing the entropy of the distribution of note pitch class counts. A lower entropy value indicates lower harmonic diversity, but implies a more stable and consistent tonality across the segment. The subscript (“4”) indicates the number of bars over which the entropy is calculated.
- **Next-Bar Grooving Pattern Similarity (GS)** is modified from the grooving pattern similarity proposed in [37]. It originally measures the global rhythmic stability across an entire song. Instead of calculating over all

pairs in the target, we adapt the metric to focus on local rhythmic stability within a song, evaluating the rhythmic coherence between each bar and its succeeding bar.

4.4 User Study

For subjective evaluation, we conduct an online listening test involving 52 volunteers: 5 professional music producers, 13 amateurs, and 34 pro-amateurs with more than 3-year music training. The volunteers are randomly assigned to distinct test sets, with each set containing 3 songs randomly selected from different genres, and for each song, there are 6 piano performances presented anonymously in random order. These piano performances include: a human piano performance, outputs of our full model and the two ablated versions, and outputs of the Pop2Piano and piano transcription model baselines. All of them are truncated to 40-second audio clips from the beginning. Subjects are asked to listen to these audio clips and provide ratings on a 5-point Likert scale for the following aspects:

- **Similarity (SI):** The degree of similarity between the piano performances and the original song.
- **Music Fluency (FL):** The degree of perceived fluency in the music, representing the smoothness and coherence of the piano performances.
- **Overall (OVL):** How much do the participants like the piano cover in the personal overall listening experience?

4.5 Results

Table 2 displays the results of the objective evaluation metrics and mean opinion scores (MOS) from the user study. In the objective evaluation, Pop2Piano shows a leading MCA score compared to other models and the human piano performances, which indicates it excels at matching the original song’s melodic contour. Except for the transcription baseline model, there is no significant difference in GS and H_4 across models, suggesting comparable local rhythmic coherence and harmonic variety.

Next, we pay attention to the result of user study. Much to our delight, the full model leads with the best scores across all aspects in the user study with statistical significance ($p < 0.05$), but there remains a gap compared to the human reference performances. Ablation 1 achieves higher scores than Pop2Piano in all aspects of the user study, both of which are trained on the paired data. This suggests that

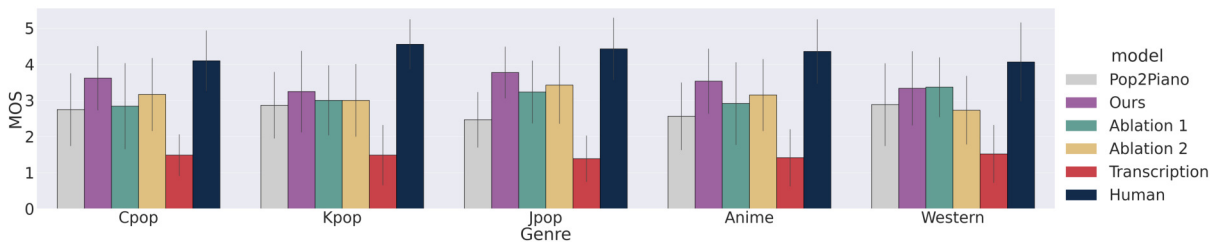


Figure 3. The MOS in overall scores (OVL) of the user study in different genres.

utilizing the weakly-aligned paired data, which avoids distorting the original piano performances, helps increase the overall listening experience quality of the model outputs for human raters. Moreover, both Ablation 2 and the transcription baseline are trained on piano-only data, but Ablation 2 performs significantly better than the baseline in both objective and subjective evaluations. This can be seen as evidence that SheetSage, as the encoder, extracts more relevant features beneficial for the piano cover generation task compared to the baseline transcription model.

5. DISCUSSION

In the experiment, we note that while Pop2Piano exhibits a significantly higher MCA score than the other models, even higher than the human performances, it fails to achieve comparably high SI ratings in the user study. We suggest this conflict arises from the assumption in MCA that two melodies must temporally correspond to each other on a fixed “time grid.” That is, the corresponding chroma features must be located at precisely the same time instants. For human listening experiences, two similar melodies only need to be coordinated on beats rather than a rigid time grid. Specifically, human perception of melodic similarity allows for the tempo or duration to be slightly changed in the same ratio, as long as their notes are located on the same underlying musical beat positions. As mentioned in Sections 1 & 2, different from transcription or arrangement, a cover song is not usually temporally aligned to the original song, i.e., the musical elements such as tempo, melody, rhythmic changed in the composition process of the piano cover. This temporal flexibility suggests that MCA as an objective measure for the cover generation task may not be adequate and calls for future endeavor to develop better alternatives.

We also find that the two ablated models have the same OVL scores in the subjective evaluation, even though Ablation 2 has never seen any pop song data during training. To investigate the reason behind this, we first examine the piano covers generated by Ablation 2. Figure 4 shows a snippet of a cover generated by this model. We note that it tends to generate repeated short notes, resulting in an unnatural-sounding performance. However, Figure 3 demonstrates the OVL scores across different music genres. Interestingly, we see that Ablation 2 outperforms Ablation 1 for the Cpop, Jpop, and Anime genres. Additionally, as shown in Table 2, the former ablation also achieves higher SI and lower FL scores than the latter. From this

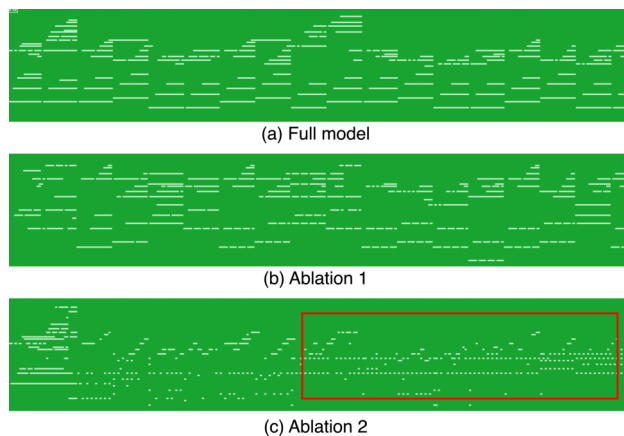


Figure 4. The pianoroll representation of a snippet from an example generated by the models. We observe that Ablation 2, which trained on piano-only data, tends to generate repeated short notes.

observation, we suggest that (i) for short audio clips (less than 40 seconds), human raters may place more emphasis on initial melodic accuracy when judging the overall perceived quality, even if Ablation 1 tends to generate more coherent and natural-sounding results; (ii) Ablation 1 does not effectively learn to precisely capture the melodic contour from the reference song condition due to the inherent alignment errors present in the weakly-aligned song-to-piano paired data it was trained on.

6. CONCLUSION

In this paper, we have presented PiCoGen2, which applies the concept of transfer learning to the piano cover generation task. We propose a training strategy that involves two stages: pre-training on piano-only data to learn fundamental piano performance skills, followed by fine-tuning on weakly-aligned song-to-piano paired examples for the cross-domain translation. A comprehensive set of experiments validate the effectiveness of the proposed transfer learning approach and the use of weakly-aligned data.

As we still require weakly-aligned data, future work can be done to tackle cover generation without relying on data alignment at all. Moreover, it is useful to have a systematic analysis to evaluate the quality of piano covers and identify the key factors influencing the result, e.g., by studying the performance difference between PiCoGen [31] and PiCoGen2. It is also interesting to generate other covers, such as orchestral covers, and to develop better objective metrics.

7. ACKNOWLEDGMENT

The work is supported by a grant from the National Science and Technology Council of Taiwan (NSTC 112-2222-E-002-005-MY2). We are also grateful to the reviewers and meta-reviewer for helpful comments that help improve the quality of the paper.

8. REFERENCES

- [1] H. Takamori, T. Nakatsuka, S. Fukayama, M. Goto, and S. Morishima, "Audio-based automatic generation of a piano reduction score by considering the musical structure," in *Proc. MultiMedia Modeling Conference (MMM)*, 2019.
- [2] J. Choi and K. Lee, "Pop2piano: Pop audio-based piano cover generation," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023.
- [3] J. Gardner, I. Simon, E. Manilow, C. Hawthorne, and J. Engel, "MT3: Multi-task multitrack music transcription," in *Proc. International Conference on Learning Representations (ICLR)*, 2022.
- [4] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proceedings of the IEEE*, 2020.
- [5] F. Zalkow and M. Müller, "Using weakly aligned score-audio pairs to train deep chroma models for cross-modal music retrieval," in *Proc. International Society for Music Information Retrieval (ISMIR)*, 2020.
- [6] C. Donahue, J. Thickstun, and P. Liang, "Melody transcription via generative pre-training," in *Proc. International Society for Music Information Retrieval (ISMIR)*, 2022.
- [7] A. Elowsson and A. Friberg, "Algorithmic composition of popular music," in *International Conference on Music Perception and Cognition (ICMPC)*, 2012.
- [8] E. Nakamura and S. Sagayama, "Automatic piano reduction from ensemble scores based on merged-output hidden markov model," in *International Conference on Mathematics and Computing (ICMC)*, 2015.
- [9] Z. Wang, K. Chen, J. Jiang, Y. Zhang, M. Xu, S. Dai, G. Bin, and G. Xia, "POP909: A pop-song dataset for music arrangement generation," in *Proc. International Society for Music Information Retrieval (ISMIR)*, 2020.
- [10] H.-W. Dong, W.-Y. Hsiao, L.-C. Yang, and Y.-H. Yang, "MuseGAN: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment," in *Proc. AAAI Conference on Artificial Intelligence (AAAI)*, 2018.
- [11] C.-Z. A. Huang, A. Vaswani, J. Uszkoreit, I. Simon, C. Hawthorne, N. Shazeer, A. M. Dai, M. D. Hoffman, M. Dinculescu, and D. Eck, "Music Transformer: Generating music with long-term structure," in *Proc. International Conference on Learning Representations (ICLR)*, 2019.
- [12] Y.-S. Huang and Y.-H. Yang, "Pop music Transformer: Beat-based modeling and generation of expressive pop piano compositions," in *Proc. ACM Multimedia (ACM MM)*, 2020.
- [13] W.-Y. Hsiao, J.-Y. Liu, Y.-C. Yeh, and Y.-H. Yang, "Compound Word Transformer: Learning to compose full-song music over dynamic directed hypergraphs," in *Proc. AAAI Conference on Artificial Intelligence (AAAI)*, 2021.
- [14] J. Huang, K. Chen, and Y.-H. Yang, "Emotion-driven piano music generation via two-stage disentanglement and functional representation," in *Proc. International Society for Music Information Retrieval (ISMIR)*, 2024.
- [15] D.-V.-T. Le, L. Bigo, M. Keller, and D. Herremans, "Natural language processing methods for symbolic music generation and information retrieval: a survey," *arXiv preprint arXiv:2402.17467*, 2024.
- [16] C. Hawthorne, E. Elsen, J. Song, A. Roberts, I. Simon, C. Raffel, J. Engel, S. Oore, and D. Eck, "Onsets and Frames: Dual-objective piano transcription," in *Proc. International Society for Music Information Retrieval (ISMIR)*, 2018.
- [17] E. Benetos, S. Dixon, Z. Duan, and S. Ewert, "Automatic music transcription: An overview," in *Proc. IEEE Signal Processing Magazine*, 2019.
- [18] K. Toyama, T. Akama, Y. Ikemiya, Y. Takida, W.-H. Liao, and Y. Mitsufuji, "Automatic piano transcription with hierarchical frequency-time Transformer," in *Proc. International Society for Music Information Retrieval (ISMIR)*, 2023.
- [19] C. Hawthorne, I. Simon, R. Swavely, E. Manilow, and J. Engel, "Sequence-to-sequence piano transcription with Transformers," in *Proc. International Society for Music Information Retrieval (ISMIR)*, 2021.
- [20] Q. Kong, B. Li, X. Song, Y. Wan, and Y. Wang, "High-resolution piano transcription with pedals by regressing onset and offset times," *IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP)*, vol. 29, pp. 3707–3717, 2021.
- [21] R. P. Paiva, T. Mendes, and A. Cardoso, "An auditory model based approach for melody detection in polyphonic musical recordings," in *Proc. Computer Music Modeling and Retrieval (CMMR)*, 2004.
- [22] ———, "On the detection of melody notes in polyphonic audio," in *Proc. International Society for Music Information Retrieval (ISMIR)*, 2005.

- [23] M. P. Ryyänänen and A. P. Klapuri, “Automatic transcription of melody, bass line, and chords in polyphonic music,” *Computer Music Journal*, 2008.
- [24] J. Weil, T. Sikora, J.-L. Durrieu, and G. Richard, “Automatic generation of lead sheets from polyphonic music signals,” in *Proc. International Society for Music Information Retrieval (ISMIR)*, 2009.
- [25] K. Weiss, T. M. Khoshgoftaar, and D. Wang, “A survey of transfer learning,” *Journal of Big data*, 2016.
- [26] P. Hamel, M. Davies, K. Yoshii, and M. Goto, “Transfer learning in MIR: Sharing learned latent representations for music audio classification and similarity,” in *Proc. International Society for Music Information Retrieval (ISMIR)*, 2013.
- [27] A. Van Den Oord, S. Dieleman, and B. Schrauwen, “Transfer learning by supervised pre-training for audio-based music classification,” in *Proc. International Society for Music Information Retrieval (ISMIR)*, 2014.
- [28] K. Choi, G. Fazekas, M. Sandler, and K. Cho, “Transfer learning for music classification and regression tasks,” in *Proc. International Society for Music Information Retrieval (ISMIR)*, 2017.
- [29] K. Choi, G. Fazekas, and M. Sandler, “Towards playlist generation algorithms using RNNs trained on within-track transitions,” in *Proc. Workshop on Surprise, Opposition, and Obstruction in Adaptive and Personalized Systems (SOAP)*, 2016.
- [30] D. Liang, M. Zhan, and D. P. Ellis, “Content-aware collaborative music recommendation using pre-trained neural networks,” in *Proc. International Society for Music Information Retrieval (ISMIR)*, 2015.
- [31] C.-P. Tan, S.-H. Guan, and Y.-H. Yang, “PiCoGen: Generate piano covers with a two-stage approach,” in *Proc. International Conference on Multimedia Retrieval (ICMR)*, 2024.
- [32] Z. Wang, D. Xu, G. Xia, and Y. Shan, “Audio-to-symbolic arrangement via cross-modal music representation learning,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.
- [33] M. Müller, Y. Özer, M. Krause, T. Prätzlich, and J. Driedger, “Sync toolbox: A python package for efficient, robust, and accurate music synchronization,” *Journal of Open Source Software*, 2021.
- [34] J. Zhao, G. Xia, and Y. Wang, “Beat Transformer: Demixed beat and downbeat tracking with dilated self-attention,” in *Proc. International Society for Music Information Retrieval (ISMIR)*, 2022.
- [35] S.-L. Wu and Y.-H. Yang, “Compose & Embellish: Well-structured piano performance generation via a two-stage approach,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023.
- [36] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Défossez, “Simple and controllable music generation,” in *Proc. Conference on Neural Information Processing Systems (NeurIPS)*, 2023.
- [37] S.-L. Wu and Y.-H. Yang, “The Jazz Transformer on the front line: Exploring the shortcomings of AI-composed music through quantitative measures,” in *Proc. International Society for Music Information Retrieval (ISMIR)*, 2020.
- [38] —, “MuseMorphose: Full-song and fine-grained piano music style transfer with one Transformer VAE,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP)*, vol. 31, pp. 1953–1967, 2023.
- [39] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, D. P. Ellis, and C. C. Raffel, “mir_eval: A transparent implementation of common MIR metrics,” in *Proc. International Society for Music Information Retrieval (ISMIR)*, 2014.
- [40] A. Andonian, Q. Anthony, S. Biderman, S. Black, P. Gali, L. Gao, E. Hallahan, J. Levy-Kramer, C. Leahy, L. Nestler, K. Parker, M. Pieler, J. Phang, S. Purohit, H. Schoelkopf, D. Stander, T. Songz, C. Tigges, B. Thérien, P. Wang, and S. Weinbach, “GPT-NeoX: Large scale autoregressive language modeling in PyTorch,” 2023. [Online]. Available: <https://www.github.com/eleutherai/gpt-neox>
- [41] P. H. Samuel Bianchini, J. Lee, J. Green, P. Lopez-Cabanillas, D. Henningsson, T. Moebert, J.-J. Ceresa, and M. Weseloh, “FluidSynth,” 2001. [Online]. Available: <https://www.fluidsynth.org/>
- [42] R. Hennequin, A. Khlif, F. Voituret, and M. Moussallam, “Spleeter: a fast and efficient music source separation tool with pre-trained models,” *Journal of Open Source Software*, 2020.
- [43] A. L. Uitdenbogerd and J. Zobel, “Manipulation of music for melody matching,” in *Proc. ACM Multimedia (ACM MM)*, 1998.

DIFF-MST: DIFFERENTIABLE MIXING STYLE TRANSFER

Soumya Sai Vanka^{1†} Christian Steinmetz^{1†} Jean-Baptiste Rolland²
Joshua Reiss¹ György Fazekas¹

¹ Centre for Digital Music, Queen Mary University of London, UK

² Steinberg Media Technologies GmbH, Germany

s.s.vanka@qmul.ac.uk, c.j.steinmetz@qmul.ac.uk

ABSTRACT

Mixing style transfer automates the generation of a multi-track mix for a given set of tracks by inferring production attributes from a reference song. However, existing systems for mixing style transfer are limited in that they often operate only on a fixed number of tracks, introduce artifacts, and produce mixes in an end-to-end fashion, without grounding in traditional audio effects, prohibiting interpretability and controllability. To overcome these challenges, we introduce **Diff-MST**, a framework comprising a differentiable mixing console, a transformer controller, and an audio production style loss function. By inputting raw tracks and a reference song, our model estimates control parameters for audio effects within a differentiable mixing console, producing high-quality mixes and enabling post-hoc adjustments. Moreover, our architecture supports an arbitrary number of input tracks without source labelling, enabling real-world applications. We evaluate our model’s performance against robust baselines and showcase the effectiveness of our approach, architectural design, tailored audio production style loss, and innovative training methodology for the given task. We provide code and listening examples online¹.

1. INTRODUCTION

Music mixing involves technical and creative decisions that shape the emotive and sonic identity of a song [1]. The process involves creating a cohesive mix of the given tracks using audio effects to achieve balance, panorama, and aesthetic value [2]. Given the complexity of the task, mastering the task of mixing often requires many years of practice. To address this, several solutions have been proposed to provide assistance or automation [3,4]. Automatic mixing systems have been designed using knowledge engineering [5,6], machine learning, and more recently deep learning methods [7–11]. Automatic mixing systems can

[†]These authors contributed equally to the work.

¹<https://sai-soum.github.io/projects/diffmst/>

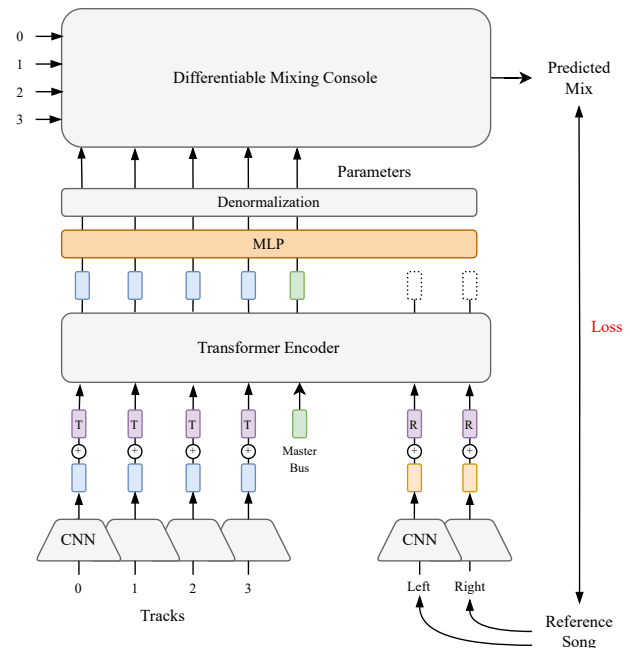


Figure 1. Diff-MST, a differentiable mixing style transfer framework featuring a differentiable multitrack mixing console, a transformer-based controller that estimates control parameters for this mixing console, and an audio production style loss function that measures the similarity between the estimated mix and reference mixes.

be further subdivided into direct transformation systems and parameter estimation systems, as shown in Figure 2. Direct transformation systems operate on tracks and predict a mix directly, in an end-to-end fashion, with the loss calculated between the ground truth mix and the predicted mix. On the other hand, parameter estimation systems take input tracks and predict control parameters for a dedicated mixing console. In such systems, the loss can either be calculated on the predicted control parameters (parameter loss) based on the availability of ground truth, or on the predicted audio against the ground truth mix (audio loss). Parameter loss, calculated on the parameters, may not be optimal for multiparameter signal processing blocks since various combinations of parameters could potentially produce similar outcomes. [7, 11] utilizes a deep learning-based direct transformation system for mixing, while [8] employs a parameter estimation-based deep learning approach. However, many of these systems are constrained

to a small number of input tracks or struggle to generalize effectively to real-world mixing scenarios. Furthermore, most of these approaches generate a mix without accounting for the desired sound and emotion. Due to the subjective nature of the task, an end-to-end approach without user control is less desirable in professional practice [12].

1.1 Mixing Style Transfer

In professional practice, the audio engineer often uses reference songs and guidelines provided by the client to make mixing decisions [13]. This encourages the development of automatic mixing systems that are aware of the intention of the mixing engineer. In our context, mixing style transfer refers to mixing in the style of given reference songs [14]. This pertains to capturing the global sound, dynamics and spatialisation of the reference song. Recently, deep learning systems have been proposed for audio production style transfer. While some approaches have considered estimating the control parameters for audio effects [15], they are so far limited to controlling only a single or small set of effects with a singular input. Whereas [16] have implemented an end-to-end style transfer system between two mixed songs which limits controllability and full raw tracks mixing. In this work, we introduce a novel deep learning-based approach to mixing multitrack audio material using a reference song, which utilises a differentiable mixing console to predict parameter values for gain, pan, 4-band equalization, compressor, and a master bus. Our proposed system is differentiable, interpretable and controllable, and can learn the mixing style from the given reference song. The contributions of this work can be summarised as follows:

1. A framework for mixing style transfer that enables control of audio effects mapping the production style from a reference onto a set of input tracks.
2. A differentiable multitrack mixing console consisting of gain, parametric equalisation, dynamic range compression, stereo panning, and master bus processing using `dasp-pytorch`², which enables end-to-end training.
3. Demonstration of the benefits of our system, including generalisation to an arbitrary number of input tracks, no requirement for labelling of inputs or enforcement of specific taxonomies, high-fidelity processing without artifacts, and greater efficiency.

2. METHOD

2.1 Problem Formulation

We can formulate the mixing style transfer task as follows. Let T be a matrix of N mono input raw tracks $\{t_1, t_2, t_3, \dots, t_N\}$ and M_r be the matrix of stereo reference mix containing two channels. A shared weight encoder f_{θ_r} and f_{θ_t} are employed to extract information from

the reference and input tracks respectively. This information is then aggregated and fed into a transformer controller network comprising a transformer encoder and a multi-layer perceptron (MLP) g_ϕ . The primary task of this network is to estimate the parameter matrix P , which consists of N parameter vectors p , each responsible for configuring the chain of audio effects for a respective track in T . Subsequently, the differentiable mixing console $h(T, P)$ processes the input tracks T using the parameters P to generate a predicted mix M_p that mirrors the style of the reference mix M_r .

$$P = g_\phi(f_{\theta_t}(T), f_{\theta_r}(M_r)) \quad (1)$$

$$M_p = h(T, P) \quad (2)$$

2.2 Differentiable Mixing Style Transfer System

We propose a differentiable mixing style transfer system (Diff-MST) that takes raw tracks and a reference mix as input and predicts mixing console parameters and a mix as output. As shown in Figure 1, our system employs two encoders, one to capture a representation of the input tracks and another to capture elements of the mixing style from the reference. A transformer-based controller network analyses representations from both encoders to predict the differentiable mixing console (DMC) parameters. The DMC generates a mix for the input tracks using the predicted parameters in the style of the given reference song. Given that our system oversees the operations of the DMC rather than directly predicting the mixed audio, we circumvent potential artefacts that may arise from neural audio generation techniques [17, 18]. This also creates an opportunity for further fine-tuning and control by the user.

2.3 Differentiable Mixing Console (DMC)

The process of multitrack mixing involves applying a chain of audio effects, also known as a channel strip, on each channel of a mixing console. The audio engineer may use these devices to reduce masking, ensure a balance between the sources, and address noise or bleed. Incorporating this prior knowledge of signal processing in the design of our mixing system, we propose an interpretable and controllable differentiable mixing console (DMC). Our console applies a chain of audio effects comprising gain, parametric equaliser (EQ), dynamic range compressor (DRC), and panning to each of the tracks to produce wet tracks. The sum of wet tracks is then sent to a master bus on which we insert stereo EQ and a DRC. This produces a mastered mix of the given tracks. We incorporate a master bus in our console as it is usual to use a mastered song as a reference in workflows. Therefore, having a master bus in the mixing console chain allows for easier optimisation of the system. To enable gradient descent and training in a deep learning framework, we require the mixing console to be differentiable. To achieve this, we use differentiable effects from the `dasp-pytorch`². The pipeline of the DMC is presented in Figure 3.

²<https://github.com/csteinmetz1/dasp-pytorch/>

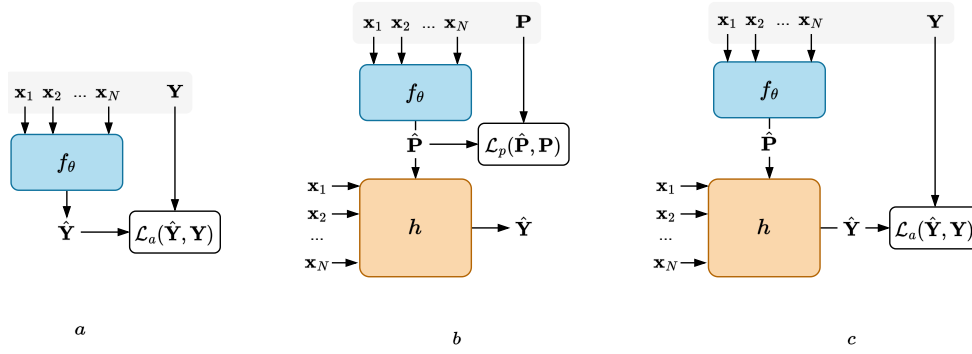


Figure 2. Formulations for deep learning-based automatic mixing systems [4]. (a) Direct transformation (b) Parameter estimation on parameter loss (c) Parameter estimation on audio loss. Here, x_i for $i \in [1, N]$ are the N input tracks, f_θ is the transformation, h is the dedicated mixing console, Y and \hat{Y} are the ground truth and predicted mix, P and \hat{P} are the ground truth and predicted control parameters and L_a and L_p are the audio and parameter loss respectively.

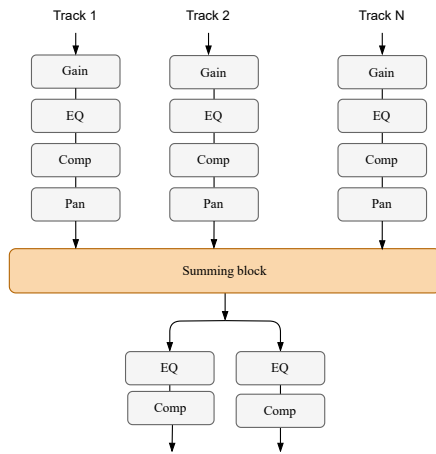


Figure 3. Differentiable Mixing console

2.4 Spectrogram Encoder

The encoder consists of a convolutional network based on the magnitude spectrum. It computes spectrograms by employing a short-time Fourier transform with a Hann window of size $N = 2048$ and a hop size of $H = 512$. The generated magnitude spectrogram is then processed through the convolutional layers. The resultant convolutional encodings are subsequently fed into a linear layer, producing a final embedding of size 512. The model includes separate shared-weight encoders: f_{θ_r} for the reference mix and f_{θ_t} for the input tracks. Each channel of stereo audio is treated as an individual track. Consequently, the stereo mix and any other stereo input tracks are loaded as separate tracks. Embeddings are computed by passing T and M_r through the encoder.

2.5 Transformer Controller

The controller features a transformer encoder and a shared-weight MLP. The transformer encoder generates style-aware embeddings using self-attention across the output of the spectrogram encoder f_{θ_r} and f_{θ_t} and a master bus embedding which is learned during training. The MLP predicts the control parameters corresponding to the channel strip for each track, and the master bus embeddings are

used to predict the master bus control parameters. A shared weight MLP is used to predict channel strip parameters for each channel. We generate the predicted mix M_p by passing the control parameters through the DMC along with the tracks. This architecture enables our system to be invariant to the number of input tracks as shown in Figure 1.

2.6 Audio Production Style Loss

The style of a mix can be broadly captured using features that describe its dynamics, spatialisation and spectral attributes [13]. We propose two different losses to train and optimise our models.

Audio Feature (AF) loss: This loss is composed of traditional Music Information Retrieval (MIR) audio feature transforms [19]. The T transforms include the root mean square (RMS) and crest factor (CF), stereo width (SW) and stereo imbalance (SI) and bark spectrum (BS) corresponding to the dynamics, spatialisation and spectral attributes respectively. We optimise our system by calculating the weighted average of the mean squared error on the audio features that minimises the distance between M_p and M_r . We compute the audio feature transforms T along with the weights w as follows:

$$T_1(\mathbf{x}) = \text{RMS}(\mathbf{x}) = \sqrt{\frac{1}{N} \sum_{i=1}^N x_i^2} \quad ; w_1 = 0.1 \quad (3)$$

$$T_2(\mathbf{x}) = \text{CF}(\mathbf{x}) = 20 \log_{10} \left(\frac{\max(|x_i|)}{\text{RMS}(\mathbf{x})} \right) \quad ; w_2 = 0.001 \quad (4)$$

$$T_3(\mathbf{x}) = \text{BS}(\mathbf{x}) = \log(\mathbf{FB} \cdot |\text{STFT}(\mathbf{x})| + \epsilon) \quad ; w_3 = 0.1 \quad (5)$$

$$T_4(\mathbf{x}) = \text{SW}(\mathbf{x}) = \frac{\frac{1}{N} \sum_{i=1}^N (x_{Li} - x_{Ri})^2}{\frac{1}{N} \sum_{i=1}^N (x_{Li} + x_{Ri})^2} \quad ; w_4 = 1.0 \quad (6)$$

$$T_5(\mathbf{x}) = \text{SI}(\mathbf{x}) = \frac{\frac{1}{N} \sum_{i=1}^N x_{Ri}^2 - \frac{1}{N} \sum_{i=1}^N x_{Li}^2}{\frac{1}{N} \sum_{i=1}^N x_{Ri}^2 + \frac{1}{N} \sum_{i=1}^N x_{Li}^2} \quad ; w_5 = 1.0 \quad (7)$$

where N represents the sequence length, x is the input tensor, \mathbf{FB} is the filterbank matrix, $\text{STFT}(x)$ represents the short-time Fourier transform of x , and ϵ is a small constant of value 10^{-8} added for numerical stability. x_{L_i} and x_{R_i} represent the input tensor corresponding to the left and right channels, respectively. The net loss is computed as follows:

$$\text{Loss}(\mathbf{M}_p, \mathbf{M}_r) = \frac{1}{2} \sum_{i=1}^2 \sum_{j=1}^5 w_j \cdot \text{MSE}(T_j(\mathbf{M}_{p_i}), T_j(\mathbf{M}_{r_i})) \quad (8)$$

where w_j is the weight associated with j^{th} transform T_j and MSE corresponds to mean squared error. The weights for the transforms were determined through empirical testing to balance the scale of various losses.

MRSTFT loss: The multi-resolution short-time Fourier transform loss [20, 21] is the sum of L_1 distance between STFT of ground truth and estimated waveforms measured in both log and linear domains at multiple resolutions, with window sizes $W \in [512, 2048, 8192]$ and hop sizes $H = W/2$. This is a full-reference metric meaning that the two input signals must contain the same content.

3. EXPERIMENT DESIGN

The task requires a dataset with multitrack audio, style reference, and the ground truth mix of the multitrack in the style of the reference for training. However, due to the lack of suitable datasets, we deploy a self-supervised training strategy to enable learning of the control of audio effects without labelled or paired training data. We achieve this by training our model under two different regimes which mainly vary in data generation and loss function.

Method 1: We extend the data generation technique used in [15] to a multitrack scenario as shown in Figure 4. We first randomly sample a $t = 10$ s segment from input tracks and generate a random mix of these input tracks by using random DMC parameters. We then split the segment of the randomly mixed audio and the input tracks into two halves, namely, M_{rA} and M_{rB} and T_A and T_B of $t/2$ s each, respectively. The model is input with T_B as input tracks and M_{rA} as the reference song. The predicted mix M_p is compared against M_{rB} as the ground truth for backpropagation and updating of weights. Using different sections of the same song for input tracks and reference song encourages the model to focus on the mixing style while being content-invariant. This method allows the use of MRSTFT loss for optimisation as we have the ground truth available. The predicted mix is loudness normalised to -16.0 dBFS before computing the loss.

Method 2: We sample a random number of input tracks between 4-16 for song A from a multitrack dataset and use a pre-mixed real-world mix of song B from a dataset consisting of full songs as the reference. We train the model using AF loss mentioned in Section 2.6 computed between M_p and M_r . This method also allows us to train the model

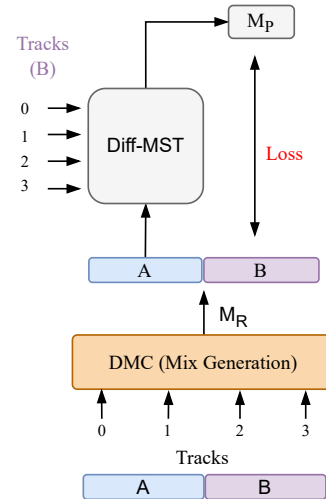


Figure 4. First training strategy from Section 3.

without the availability of a ground truth. Unlike Method 1, this approach exposes the system to training examples more similar to real-world scenarios where the input tracks and the reference song come from a different song. However, due to random sampling, some input track and reference song combinations may not be realistic.

3.1 Datasets

Multitrack: For both training methods, we utilise multitrack from MedleyDB [22, 23] and Cambridge.mt³ which contains a total of 196 and 535 songs respectively, sampled at $f_s = 44100$ Hz. For both datasets, we generate a train/test/validation split of 4:1:1. During training, songs are picked at random from the training split of both datasets. Thereafter, we randomly sample a section of the song as input tracks. We find a random offset for sampling multitrack by finding a section of the mix $x[i]$ that has mean energy above the threshold, $\frac{1}{N} \sum_{i=1}^N |x[i]|^2 \geq 0.001$. During training, each channel corresponding to a stereo raw track is treated as a separate mono track. We check the mean energy of each track to avoid loading silent tracks. All input tracks are loudness normalised to -48.0 dBFS.

Reference Songs: For Method 1 we generate a random mix using random parameters and input tracks as mentioned in Section 3 and loudness normalise the random mix to -16 dBFS. For Method 2, we use real-world songs from MTG-Jamendo which contains more than 55k songs in MP3 format [24]. We pick a random segment $y[i]$ of a random song from the dataset as a reference and check for mean energy above the threshold, $\frac{1}{N} \sum_{i=1}^N |xy[i]|^2 \geq 0.001$. We loudness normalise the reference to -16 dBFS and load stereo information on separate channels.

3.2 Training Details

Our model contains 190M trainable parameters, 76.5M corresponding to the track and mix encoder, and 37.9M

³<https://cambridge-mt.com/>

for the transformer controller. We train five variations of our model differing in the number of tracks, methodology and loss function used. To remedy the bottleneck of reading multitrack audio data from disk, we load data into RAM every epoch from both the training and validation sets respectively. The number of training steps per epoch is comprised of passing over these examples 20 times for training and 4 times for validation, sampling random examples at each step. This provides a tradeoff between training speed and data diversity. We train all our models with a batch size of 2 and a learning rate of 10^{-5} with the *Adam* optimiser. We accumulate gradients over 4 batches and use `pytorch` for training.

Diff-MST-MRSTFT: We generate data using the method 1 described in Section 3 and calculate MRSTFT loss for weight update and backpropagation. We train two variations of the model with a maximum of 8 tracks and 16 tracks as input, each for 1.16 M steps.

Diff-MST-MRSTFT+AF: We fine-tune both versions of the pre-trained Diff-MST-MRSTFT using the synthetically generated data of method 1 in Section 3 with AF loss described in Section 2.6 for 20k steps.

Diff-MST-AF: We follow the training strategy mentioned in method 2 of Section 3 and use real-world songs as the reference. We train this model for 1.16 M steps using the AF loss described in Section 2.6. We train with a varying number of tracks with an upper limit of 16.

3.3 Baselines

We compare the performance of our model against three baselines: an equal loudness mix (lowest anchor), the mix generated using the pre-trained mixing style transfer (MST) model by [16] (state-of-the-art), and two human mixes. We picked three songs from the Cambridge online multitrack repository belonging to the genres of electronic, pop, and metal for our main evaluation. Each of the songs contains between 12 and 22 input tracks. We selected references from popular songs.

Equal Loudness: We loudness normalise the tracks to -48.0 dBFS and take the mean among the tracks to generate the mix which is then normalised. This generates a loudness-normalised sum of input tracks. We consider this system to be the lowest anchor as it does not consider any style information or mixing transformations.

MST [16]: The method uses a pre-trained source separation model to generate stems from input and reference mix and perform stem-to-stem style transfer using a contrastive learning-based pre-trained audio effect encoder. The stems are mixed using a TCN-based model conditioned on style embeddings. Since the model performs a mix-to-mix transformation, we make use of the equal loudness mix of input tracks as the input to be transformed by the model. This allows us to extend the system to perform mixing

style transfer for any number of input tracks. This puts the system at a disadvantage as it is trained to work for mix-to-mix scenarios where good-quality mixes are used as input, leading to better-quality extracted stems.

Human Mixes: We asked two audio engineers with professional practice to mix the three songs using the corresponding references. Each of them mixed all three songs until the end of the first chorus.

4. OBJECTIVE EVALUATION

We evaluate the performance of our model against three baselines listed in Section 3.3. For the first evaluation, we compare the mixes generated by all five of our systems described in Section 3.2 and the baselines for three songs belonging to the genres of pop, electronic and metal. We manually picked the songs for the input tracks and the references for each of these cases. A 10-second section ranging between the middle of the first verse to the middle of the first chorus was used for evaluation in Table 1. We loudness normalise the reference mix to -16 dBFS and the predicted mix to -22 dBFS before predicting the metrics.

We report the average AF loss and individual weighted audio feature transforms from Section 2.6 for all three songs. Our Diff-MST system trained on real-world songs as reference using AF loss performs the best, closely followed by the MST [16], human engineer mix, and the mix from our Diff-MST-MRSTFT+AF-16 system.

For the second evaluation, we compute average metrics across 100 randomly sampled examples with multitrack taken from the unseen set of Cambridge multitrack and reference songs from MUSED18 [25]. We compare the performance of our systems and the baselines MST [16] and the equal loudness system as shown in Table 2. We report individual weighted audio features from the AF loss along with average loss and Fréchet Audio distance (FAD) [26]. The FAD metric is employed to gauge the efficacy of music enhancement approaches or models by comparing the statistical properties of embeddings generated by their output to those of embeddings generated from a substantial collection of clean music. In this context, we analyze the distributions of real-world songs against the mixes generated by various systems using the VGGish model. Again, Diff-MST-AF-16 outperforms other approaches at capturing the dynamics, spatialisation and spectral attributes of the reference songs.

5. DISCUSSION

Overall, the results indicate the effectiveness of our approach, architecture choice, custom audio production style loss, and novel training regime for the task. The reported metrics for both evaluations show improved performance when trained on a larger number of tracks. Furthermore, we also see that the systems trained or fine-tuned using AF loss generally perform better than those trained with MRSTFT loss, specifically in improving the spatialisation and dynamics of the mixes, thus showing the efficacy of

Method	RMS ↓	CF ↓	SW ↓	SI ↓	BS ↓	AF Loss ↓
Equal Loudness	3.11	0.51	3.16	0.21	33.3	33.389
MST [16]	3.15	0.45	4.64	0.13	0.09	<u>0.185</u>
Diff-MST						
MRSTFT-8	3.63	1.44	1.97	4.29	0.17	0.379
MRSTFT-16	3.40	0.98	1.91	1.99	0.19	0.328
MRSTFT+AF-8	3.12	0.86	1.29	0.76	0.13	0.237
MRSTFT+AF-16	3.15	0.43	0.89	2.20	0.11	<u>0.186</u>
AF-16	2.39	0.07	1.60	0.97	0.13	0.168
Human 1	3.02	0.26	2.05	0.46	0.17	0.218
Human 2	3.21	0.14	3.63	2.29	0.11	<u>0.180</u>

Table 1. Average of metrics computed across the same section of three songs from three different genres. RMS is reported in e-04, CF in e-01, SW in e-02, and SI in e-02. We have provided audio examples as supplementary material.

Method	RMS ↓	CF ↓	SW ↓	SI ↓	BS ↓	AF loss ↓	FAD ↓
Equal Loudness	2.31e-04	2.11	6.03	1.41	32.7	6.55e+00	17.6
MST [16]	4.07e-04	1.72	5.84	0.89	0.31	<u>7.85e-02</u>	17.9
Diff-MST							
MRSTFT-8	3.08e+06	3.91	4.55	3.38	7.06	6.15e+05	51.3
MRSTFT-16	2.23e+03	4.07	5.00	1.97	1.81	4.47e+02	65.9
MRSTFT+AF-8	2.00e+05	1.79	4.58	2.86	6.89	4.00e+04	48.3
MRSTFT+AF-16	2.46e+00	1.14	4.29	3.44	0.92	6.92e-01	51.1
AF-16	4.24e-04	0.67	4.78	0.22	0.11	3.26e-02	15.1

Table 2. Average of metrics using unseen tracks from Cambridge dataset and mixes from MUSDB18 [25]. CF in e-02, SW in e-02, SI in e-02.

our hand-crafted audio feature-based loss function. The significant difference in the Bark spectrum values between the equal loudness and our system’s mixes suggests that mixes generated using our system have undergone significant spectral processing, resulting in an increased spectral similarity between the reference song and the predicted mix. The metrics indicate inferior performance for the Diff-MST-MRSTFT-8/16 model compared to all our proposed models. This may be attributed to the training data, which is generated using random mixing console parameters, often resulting in mixes that sound unrealistic. However, fine-tuning with AF loss during the last steps notably enhances performance. This improvement could be attributed to AF loss compelling the model to enhance dynamics and spatialization, as evidenced by the reported metrics. We observe a notable enhancement in performance through training on real-world songs, underscoring the significance of high-quality real-world data. Although the system demonstrates promising outcomes, it is not without its limitations. While we note higher metric values for certain features on the human mixes, this can be explained by the fact that human engineers often strive to capture the overall essence of the reference song. However, they may also incorporate creative elements leading to spatialization and dynamics that diverge significantly from the reference. Our metrics serve to quantify the similarity between the reference song and the predicted mix, which is suitable for the task at hand but may fall short in assessing the creative or unconventional decisions made by human

engineers during the mixing process. Additionally, while FAD indicates the predicted audio quality, it might not capture the intricate nuances involved in the mixing process, such as frequency masking and achieving balance and spatialization.

Moreover, we noticed a decline in the system’s mixing capabilities as the number of input tracks increased beyond what it was trained on. Additionally, our mixing console lacks a crucial reverb module essential for comprehensive mixing tasks. Determining the optimal method for processing the entire song poses a challenge, as inferring over the entire song length may result in overly sparse embeddings. Our current system also falls short in modelling mixing context in all possible senses as discussed in [27]. However, we address this challenge by incorporating a reference input, typically selected by the mixing engineer or client. The reference song serves as a proxy for some of the contextual information that engineers typically rely on when making mixing decisions. Lastly, while real-world mixing often entails dynamic adjustments to effect parameters over the course of a song, our system is presently constrained to static mixing configurations.

6. CONCLUSION

In this work, we proposed a framework for mixing style transfer for multitrack music using a differentiable mixing console. Our system is rooted in strong inductive bias, taking inspiration from real-world mixing consoles and channel strips and predicts control parameters for these signal processing blocks allowing interpretability and controllability. Our system supports inputting any number of raw tracks, without source labelling. Furthermore, we circumvent possibilities for audio degradation and artifacts with our design choice for a parameter estimation-based system. Objective evaluations demonstrate that our Diff-MST-MRSTFT+AF-16 system surpasses all baseline methods. The reported metrics give us an insight into the impact of architectural and training design choices. We show that training on a larger number of input tracks improves the performance substantially while running inference on real-world examples that generally contain a larger number of input tracks. We also demonstrate the benefits of training on real-world quality audio examples. While our research has produced promising results based on objective metrics, it is important to acknowledge our evaluation’s constraints, as we have not conducted subjective assessments via listening tests. While objective metrics offer valuable insights into the model’s performance, integrating subjective evaluations would provide a more comprehensive understanding of its efficacy in practical applications. Future work includes conducting an extensive subjective evaluation alongside assessing the usability of a prototype of the system that is integrated into the real-world workflow in the digital audio workstation (DAW). Further, work towards developing a robust understanding and objective metrics for mix similarity and mixing style is imperative for enhancing these systems.

7. ACKNOWLEDGMENTS

We express our sincere gratitude to the ISMIR reviewers for providing valuable feedback on our work. Further, we extend our thanks to Steinberg’s research and development team for their unwavering support and honest feedback throughout this project.

This work is funded and supported by UK Research and Innovation [grant number EP/S022694/1] and Steinberg Media Technologies GmbH under the AI and Music Centre for Doctoral Training (AIM-CDT) at the Centre for Digital Music, Queen Mary University of London, London, UK.

8. ETHICAL STATEMENT

We utilized open-source multitrack data from MedleyDB [22, 23] and the web forum Cambridge.mt³ as well as full songs from MTG-Jamendo [24] to train our models. MedleyDB and MTG-Jamendo are available under the licenses CC-BY-NC-SA and Apache 2.0, respectively. Cambridge.mt is an educational web platform managed by Mike Senior, a professional mixing engineer, where artists and professional engineers consensually share audio files for multitracks and corresponding mixes. The terms and conditions permit educational and non-commercial research usage.

The design of our system integrates user-centric principles and has been built upon extensive qualitative research involving professional engineers [13]. Moreover, the design of this system is grounded in traditional mixing methods and expert knowledge, incorporating context and ensuring controllability and interpretability. In professional environments, our system can provide technical assistance for mixing, allowing more time for creative expression. Additionally, our system aims to make music mixing more accessible for beginners and non-specialists, promoting the democratisation of music production. The system can be used as a tool for learning basic mixing skills using the reference method. This system has the potential to support musicians and bands to create and distribute their music affordably, resulting in diverse representation within the music industry. However, there are potential drawbacks. Automated mixing systems might reduce the need for professional audio engineers in budget productions, impacting their job opportunities. Moreover, the widespread use of these tools may lead to homogenization in music production, resulting in algorithmically driven mixes overshadowing unique stylistic traits.

9. REFERENCES

- [1] M. Miller, *Mixing Music*. London, UK: Dorling Kindersley Ltd, 2016.
- [2] R. Izhaki, *Mixing Audio: Concepts, Practices, and Tools*. New York, USA: Routledge, 2017.
- [3] B. De Man, J. D. Reiss, and R. Stables, “Ten years of automatic mixing,” in *3rd Workshop on Intelligent Music Production*, September 2017.
- [4] C. J. Steinmetz, S. S. Vanka, M. A. Martínez Ramírez, and G. Bromham, *Deep Learning for Automatic Mixing*. Bengaluru, India: Proc. of the 23rd Int. Society for Music Information Retrieval Conf. (ISMIR), December 2022. [Online]. Available: <https://dl4am.github.io/tutorial>
- [5] A. Tom, J. D. Reiss, and P. Depalle, “An automatic mixing system for multitrack spatialization for stereo based on unmasking and best panning practices,” in *146th Audio Engineering Society Convention*. Dublin, Ireland, UK: Audio Engineering Society, 2019.
- [6] D. Moffat and M. Sandler, “Machine learning multitrack gain mixing of drums,” in *147th Audio Engineering Society Convention*. Dublin, Ireland, UK: Audio Engineering Society, 2019.
- [7] M. A. Martínez-Ramírez, D. Stoller, and D. Moffat, “A deep learning approach to intelligent drum mixing with the Wave-U-Net,” *Journal of the Audio Engineering Society*, vol. 69, pp. 142–151, March 2021.
- [8] C. J. Steinmetz, J. Pons, S. Pascual, and J. Serrà, “Automatic multitrack mixing with a differentiable mixing console of neural audio effects,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021.
- [9] J. T. Colonel and J. Reiss, “Reverse engineering of a recording mix with differentiable digital signal processing,” *The Journal of the Acoustical Society of America*, vol. 150, no. 1, pp. 608–619, 2021.
- [10] C. J. Steinmetz, “Learning to mix with neural audio effects in the waveform domain,” Master’s thesis, Universitat Pompeu Fabra, September 2020.
- [11] M. A. Martínez-Ramírez, W.-H. Liao, G. Fabbro, S. Uhlich, C. Nagashima, and Y. Mitsufuji, “Automatic music mixing with deep learning and out-of-domain data,” in *Proc. of the 23rd Int. Society for Music Information Retrieval Conf. (ISMIR)*, Bengaluru, India, 2022.
- [12] S. Vanka, M. Safi, J.-B. Rolland, and G. Fazekas, “Adoption of AI technology in music mixing workflow: An investigation,” in *154th Audio Engineering Society Convention*. Audio Engineering Society, 2023.
- [13] S. S. Vanka, M. Safi, J.-B. Rolland, and G. Fazekas, “The role of communication and reference songs in the mixing process: Insights from professional mix engineers,” *Journal of the Audio Engineering Society*, vol. 72, no. 1/2, pp. 5–15, 2024.
- [14] S. Vanka, J.-B. Rolland, and G. Fazekas, “Intelligent music production: Music production style transfer and analysis of mix similarity,” in *16th Digital Music Research Network (DMRN+ 16) Workshop*, London, UK, 2021.

- [15] C. J. Steinmetz, N. J. Bryan, and J. D. Reiss, “Style transfer of audio effects with differentiable signal processing,” *Journal of the Audio Engineering Society*, vol. 70, no. 9, pp. 708–721, September 2022.
- [16] J. Koo *et al.*, “Music mixing style transfer: A contrastive learning approach to disentangle audio effects,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Rhodes Island, Greece: IEEE, April 2023, pp. 1–5.
- [17] J. Pons, S. Pascual, G. Cengarle, and J. Serra, “Upsampling artifacts in neural audio synthesis,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 3005–3009.
- [18] J. Engel, L. H. Hantrakul, C. Gu, and A. Roberts, “DDSP: Differentiable digital signal processing,” in *International Conference on Learning Representations*, 2020.
- [19] B. Man, B. Leonard, R. King, J. D. Reiss *et al.*, “An analysis and evaluation of audio features for multi-track music mixtures,” in *Proc. of the 15th Int. Society for Music Information Retrieval Conf. (ISMIR)*, Taipei, Taiwan, 2014.
- [20] X. Wang, S. Takaki, and J. Yamagishi, “Neural source-filter waveform models for statistical parametric speech synthesis,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 402–415, 2019.
- [21] C. J. Steinmetz and J. D. Reiss, “auraloss: Audio focused loss functions in pytorch,” in *Digital music research network one-day workshop (DMRN+ 15)*, London, UK, December 2020.
- [22] R. M. Bittner, J. Salamon, M. Tierney, M. Mauch, C. Cannam, and J. P. Bello, “MedleyDB: A multitrack dataset for annotation-intensive mir research.” in *Proc. of the 15th Int. Society for Music Information Retrieval Conf. (ISMIR)*, vol. 14, Taipei, Taiwan, 2014, pp. 155–160.
- [23] R. M. Bittner *et al.*, “MedleyDB 2.0: New data and a system for sustainable data collection,” in *Proc. of the 17th Int. Society for Music Information Retrieval Conf. (ISMIR)*, New York, USA, 2016.
- [24] D. Bogdanov, M. Won, P. Tovstogan, A. Porter, and X. Serra, “The MTG-Jamendo dataset for automatic music tagging,” in *Machine Learning for Music Discovery Workshop, International Conference on Machine Learning*, Long Beach, CA, United States, 2019.
- [25] Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimitakis, and R. Bittner, “The MUSDB18 corpus for music separation,” December 2017.
- [26] K. Kilgour, M. Zuluaga, D. Roblek, and M. Sharifi, “Fréchet Audio Distance: A Reference-Free Metric for Evaluating Music Enhancement Algorithms,” in *Proc. Interspeech 2019*, 2019, pp. 2350–2354.
- [27] M. N. Lefford, G. Bromham, G. Fazekas, and D. Mof-fat, “Context-aware intelligent mixing systems,” *Journal of the Audio Engineering Society*, vol. 69, pp. 128–141, March 2021.

SEMI-SUPERVISED CONTRASTIVE LEARNING OF MUSICAL REPRESENTATIONS

Julien Guinot^{1,2} Elio Quinton² György Fazekas¹

¹ Centre for Digital Music, Queen Mary University of London, U.K.

² Music & Audio Machine Learning Lab, Universal Music Group, London, U.K.

j.guinot@qmul.ac.uk

ABSTRACT

Despite the success of contrastive learning in Music Information Retrieval, the inherent ambiguity of contrastive self-supervision presents a challenge. Relying solely on augmentation chains and self-supervised positive sampling strategies can lead to a pretraining objective that does not capture key musical information for downstream tasks. We introduce semi-supervised contrastive learning (SemiSupCon), a simple method for leveraging musically informed labeled data (supervision signals) in the contrastive learning of musical representations. Our approach introduces musically relevant supervision signals into self-supervised contrastive learning by combining supervised and self-supervised contrastive objectives in a simpler framework than previous approaches. This framework improves downstream performance and robustness to audio corruptions on a range of downstream MIR tasks with moderate amounts of labeled data. Our approach enables shaping the learned similarity metric through the choice of labeled data that (1) infuses the representations with musical domain knowledge and (2) improves out-of-domain performance with minimal general downstream performance loss. We show strong transfer learning performance on musically related yet not trivially similar tasks - such as pitch and key estimation. Additionally, our approach shows performance improvement on automatic tagging over self-supervised approaches with only 5% of available labels included in pretraining.

1. INTRODUCTION

Self-supervised learning (SSL) has emerged as a powerful paradigm for learning structured representations of data without the need for costly and time-consuming labeling. SSL approaches have achieved competitive performance on downstream tasks with minimal labeled data in many domains [1–8]. In the field of Music Information Retrieval (MIR), the complexity of labeling for many

tasks - due to the high technicality and subjectivity - underscores the importance of such self-supervised methods [5,8–14]. Instance-discriminative SSL specifically, such as contrastive learning, has proven to be effective in learning meaningful representations for a multitude of downstream tasks [8,9,15]. However, major design choices such as positive mining strategies and augmentations are crucial to downstream performance [8,16–19], and selecting a strategy for a given task remains a challenge, prompting the reintroduction of supervision within the SSL framework. In MIR, the key notion of “similarity” in contrastive learning can derive from a variety of musical attributes. Guiding the model towards a musically informed similarity metric is an objective that may be achieved by leveraging supervised labeled data, i.e. *supervision signals*.

In this work, we propose a novel semi-supervised contrastive learning method, SemiSupCon. Our method leverages both unlabeled and labeled data for contrastive learning, an extension of Contrastive Learning of Musical Representations (CLMR) in the music domain [8] and SupCon [20] in Computer Vision. Our approach differs from previous attempts at combining self-supervised contrastive learning with an auxiliary supervision signal in that it is the first to our knowledge to implement a fully-contrastive semi-supervised learning pipeline. The simple machinery of this method allows for leveraging new supervision signals beyond labels within the contrastive objective.

Briefly, the contributions of this work are the following: (1) We propose an architecturally simple extension of self-supervised and supervised contrastive learning to the semi-supervised case with the ability to make use of a variety of supervision signals. (2) We show the ability of our method to shape the representations according to the support supervision signal used for the learning task with minimal performance loss on other tasks. (3) We propose a representation learning framework with low-data regime potential and higher robustness to data corruption. Our implementation and experiments are made publicly available at <https://github.com/Pliploop/SemiSupCon>

2. RELATED WORK

Self-supervised learning aims to learn representations that capture the semantic structure of data without labels in order to utilize these representations on downstream tasks. Among self-supervised learning approaches, Contrastive



Learning teaches a model to identify augmented samples originating from the same data point amongst distractor negative samples [1, 8]. Beyond its success in neighboring fields, MIR and audio representation learning have largely benefited from Contrastive Learning approaches [2, 8, 9, 21–23]. From the implementation of CLMR, several works have expanded on contrastive learning for music, with competitive results on many downstream tasks and in multiple modalities [9, 10, 13, 24, 25]. One of the key challenges of contrastive learning is establishing an effective positive mining strategy to select positive and negative samples [16–18]. Previous studies show that both the positive mining strategy and the augmentation chain are crucial toward the performance on a given downstream task [16–19] - an inappropriate sampling strategy can lead to treating similar samples as negatives, to the detriment of downstream performance [26–28]. In MIR specifically, even the temporal proximity of two positive segments within an audio clip is influential on downstream performance depending on the task, as shown in [18]. Previous works have attempted to design domain-appropriate strategies for music and audio contrastive learning, including auxiliary similarity metrics [24, 29–31], weak supervision [15, 32–34], as well as music-specific preprocessing and augmentations [8, 10, 25].

Self-supervision is inherently limited by the ability of the positive mining strategy to select semantically relevant positives. Some approaches have attempted to reintroduce supervision signals for positive mining within the contrastive objective to reduce noise induced by self-supervised pseudolabels. SupCon [20] introduces supervised contrastive learning, which uses class labels to mine positives. Other approaches have extended contrastive learning to the semi-supervised regime by leveraging both labeled and unlabeled data. However, these approaches often use complex machinery, such as auxiliary classification modules or multiple losses [29, 35–37], making them inflexible and difficult to balance with regard to the supervision signal. Recently, in MIR, Akama *et. al* [29] employ contrastive learning as an auxiliary loss for automatic tagging, with improved results over supervision alone.

3. METHODS

3.1 Self-Supervised contrastive learning

In the SSL setting for contrastive learning [1, 8], each sample in a N -sample batch is augmented into two views through a stochastic augmentation chain. Let B be a batch of these augmented views x_i . Indices $i \in I = \{1, 2, \dots, 2N\}$ represent the index of a data point in the batch (anchor). $p(i)$ is the index of the augmented data point originating from the same original sample as the anchor (positive sample). $N(i)$ is the set of negatives: data points in the augmented batch excluding the anchor and positives: $N(i) = I \setminus \{i, p(i)\}$. Let z_i be the embedded representation of the data point by an encoder $E : x \mapsto E(x) \in \mathbb{R}^{d_E}$ and a projection head $g : E(x) \mapsto g(E(x)) = z_i \in \mathbb{R}^{d_g}$ into the contrastive latent space. In the SSL setting, the objective function for the contrastive method is the nor-

malised temperature-scaled cross-entropy loss [1] between samples i and $p(i)$ for all pairs in the batch:

$$\mathcal{L}_{ssl}^i = -\log \frac{\exp(\text{sim}(z_i, z_{p(i)})/\tau)}{\sum_{n \in N(i) \cup \{p(i)\}} \exp(\text{sim}(z_i, z_n)/\tau)} \quad (1)$$

Where τ is a temperature hyperparameter, sim is a similarity function - usually, cosine similarity [1, 8]. For the sake of brevity we notate $\sigma_{i,j} = \exp(\text{sim}(z_i, z_j)/\tau)$ in the rest of this work.

3.2 Supervised contrastive learning

In the supervised setting [20], the set of *supervised* positives $P_s(i)$ are now defined by the label information in the set of labels y_i : $P_s(i) = \{p \in I | y_p = y_i\} \setminus i$. As in [20], the supervised contrastive loss objective is given by:

$$\mathcal{L}_{sl}^i = \frac{-1}{|P_s(i)|} \sum_{p \in P_s(i)} \log \frac{\sigma_{i,p}}{\sum_{n \in N(i) \cup P_s(i)} \sigma_{i,n}} \quad (2)$$

The contrastive matrix \mathbf{M} is constructed by leveraging class information obtained by mining the labels, i.e. if two samples x_i and x_j are in the same category then $\mathbf{M}_{i,j} = 1$.

3.3 Semi-supervised Contrastive Learning

Let \mathcal{U} be a set of unlabeled samples, and \mathcal{S}^* be a set of labeled samples. We sample a proportion p_s of the labeled dataset for training such that $|\mathcal{S}| = p_s |\mathcal{S}^*|$. Let $\mathcal{A} = \mathcal{U} \cup \mathcal{S}$ be the set of all data points seen during training. During training, we use both labeled and unlabeled data points by sampling batches B comprised of proportions b_s (resp. $1 - b_s$) of labeled (resp. unlabeled) samples. $P_s(i) = \emptyset$ if i is the index of an unlabeled data point. We now define our semi-supervised contrastive loss, with $P_A(i) = P_s(i) \cup P_u(i)$, where $P_u(i)$ is the set of self-supervised positives ($\{p(i)\}$ in Eq. 1):

$$\mathcal{L}_{sem}^i = \frac{-1}{|P_A(i)|} \sum_{p \in P_A(i)} \log \left(\frac{\sigma_{i,p}}{\sum_{n \in N(i) \cup P_A(i)} \sigma_{i,n}} \right) \quad (3)$$

With the inclusion of both sets of positives, we generalize to both labeled and unlabeled data in our representation learning task: Note that if $\mathcal{U} = \emptyset$ or $\mathcal{S} = \emptyset$, the semi-supervised contrastive loss reverts back to the fully-supervised loss or the fully self-supervised loss (as $P_s(i) = \emptyset$) respectively. The approach is shown Figure 1.

This approach differs from simply adding the supervised and self-supervised contrastive losses together, as our objective maintains the number of samples to discriminate against in the self-supervised setting by leveraging labeled data as negatives for the self-supervised samples.

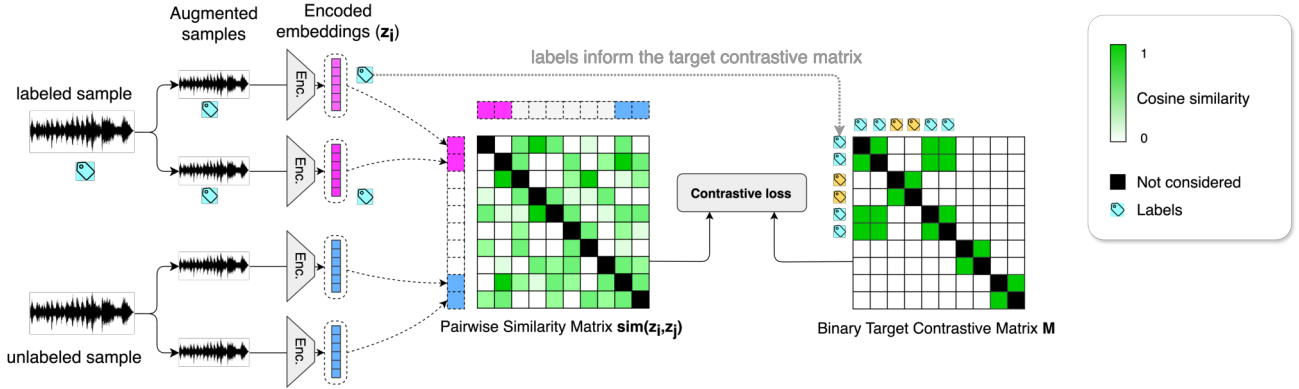


Figure 1: Semi-Supervised Contrastive Learning. The sparsely labeled dataset contains a mix of unlabeled data and labeled data. Given a batch, available labels (blue and yellow tags) are used to augment \mathbf{M} . Unlabeled samples degenerate back to the self-supervised case. Loss is computed between the pairwise similarity matrix from the encoded embeddings and the target matrix using Equation 3

3.3.1 Extension to other supervision signals

The range of supervision signals this method can leverage are limited only by the ability to construct the target contrastive matrix. In this, SemiSupCon can make use of support data beyond single label multiclass tasks. To demonstrate this, we devise two strategies for training on MagnaTagATune [38], which are studied in Section 5.4. For a multi-label signal, if $C \in \mathbb{N}$ labels coincide between two samples, we set the corresponding index in the target contrastive matrix $\mathbf{M}_{i,j} = 1$. The criterion C is a hyperparameter which is studied in Section 5.4. By default we use $C = 1$, i.e., if any labels coincide between two samples they are considered as positives.

Further, we can construct a target continuous similarity metric factor $\alpha_{i,j}$ which denotes the degree of “semantic similarity” between the samples by weighing the common classes by the total number of labels:

$$\alpha_{i,j} = \frac{2C_{i,j}}{(C_i + C_j)}$$

$C_{i,j}$ is the number of common classes for x_i and x_j , C_i and C_j are the number of classes of x_i and x_j . The similarity term $\sigma_{i,j}$ is then weighted by $\alpha_{i,j}$ in Eq. 3.

4. EXPERIMENTS AND RESULTS

4.1 Datasets

For our experiments, we use The Free Music Archive (FMA) dataset [39] as a self-supervised dataset, i.e., we do not use its labels. To match the scale of the supervised datasets, we elect to use the *medium* subset, containing 25000 clips of 30 seconds of audio.

We utilize several labeled datasets as support labeled data for training and for evaluation to demonstrate the cross-domain usefulness of SemiSupCon. For automatic tagging and most of our experiments, we use MagnaTagATune (MTAT) [38] as labeled data as a proxy evaluation of general music understanding. We reproduce the canonical 12:3:1 train-test-validation splits [8]. We use MTG-Jamendo (all subsets, including the top 50 tags,

genre, mood/theme, and instrument) [40] as another tagging dataset. We use NSynth [41] for pitch and instrument classification of short snippets, and MedleyDB [42, 43] for instrument classification with longer audio clips than NSynth. We use Giantsteps [44] as a key classification dataset - as in [45], we use the original dataset as our training set and the MTG-Giantsteps dataset as our test set. For genre classification, we use the fault-filtered GTZAN dataset [46, 47]. We use the VocalSet dataset [48] for two additional tasks: singer identification and technique classification. Finally, we regress Arousal (A) and Valence (V) on EmoMusic [49] as a downstream evaluation task only, with the same train-test split as [45].

4.2 Model input, augmentation chain

As in [8, 9], we crop 2.7 second segments of mono 22050kHz audio as input to the encoders, SampleCNN [50] or TUNE+ [9]. We sample and augment 2 adjacent non-overlapping segments as positives. The dimensions of the encoders and the 2-layer ReLU-nonlinear projection head are $d_E = 512$ and $d_g = 128$. We implement a stochastic augmentation chain similar to CLMR [8], TUNE [9], and [10]. In order, we apply (Table 1):

Augmentation	probability	parameter	Min/Max	unit
Gain	0.4	Gain	-15 [‡] / 5 [‡]	dB
Polarity inv.	0.6	-	-	-
Colored Noise	0.6	Signal/noise ratio	3 [‡] / 30 [‡]	dB
		Spectral decay	-2 [‡] / 2 [‡]	dB/octave
<i>Filtering</i>	(One of)			
Low pass	0.3	Cutoff	0.15 [‡] / 7 [‡]	kHz
High pass	0.3	Cutoff	0.2 [‡] / 2.4 [‡]	kHz
Band pass	0.3	center frequency	0.2 / 4 [‡]	kHz
		Bandwidth fraction	0.5 [‡] / 2	-
Band cut	0.3	center frequency	0.2 / 4 [‡]	kHz
		Bandwidth fraction	0.5 [‡] / 2	-
Pitch shifting	0.6	transpose	-4 [‡] / 4 [‡]	semitones
Delay	0.6	reflection time	100 [‡] / 500	ms
		reflections	1 [‡] / 3 [‡]	-
		attenuation	-6 [‡] / -3 [‡]	dB/reflection
		wet/dry factor	0.25 [‡] / 1	-

Table 1: Training augmentation chain. Only one amongst the four frequency filters is applied at once. Ranges denoted with [‡] (resp. [‡]) are subject to increasing (resp. decreasing) in Subsection 5.3

Ours	$b_s = p_s$	AUROC \uparrow		AP \uparrow	
		SampleCNN (\ddagger)	TUNe+ (\star)	\ddagger	\star
Self-Supervised	0	88.8	88.9	41.6	41.6
	0.05	89.4	89.4	42.5	42.1
	0.1	89.5	89.4	42.2	42.2
	0.25	89.5	89.4	42.5	42.8
Semi-Supervised	0.5	89.7	89.5	42.9	43.3
	0.75	89.9	89.8	43.3	43.5
	0.5/1	89.8	89.8	43.1	43.0
Supervised	1	90.3	90.1	44.3	44.6
<i>Literature</i>					
SampleCNN [8]	-	89.3* (88.6 \dagger [8])		41.2* (34.4 \dagger [8])	
CLMR _{FMA} [8]	-	86.6 \dagger		31.2 \dagger	
TUNe+ [9]	-	89.2 \dagger		36.6 \dagger	
MERT [5]	-	91.0 \dagger		39.3 \dagger	

Table 2: Performance on automatic tagging. Results denoted by \dagger are reported in their original paper. In our experiment, we constrain $p_s = b_s$ except for one run where $p_s = 1, b_s = 0.5$. We trained our own end-to-end supervised SampleCNN with the same compute budget as SemiSupCon and report results with *.

4.3 Training and evaluation details

For our baseline models, we adopt a training setup similar to TUNe [9] and CLMR [8]. Models are trained for 200k steps on semi-supervised batches sampled from MagnaTagATune as labeled data and FMA-Medium as unlabeled data [38,39] using the Pytorch Adam optimiser with a learning rate of $1e^{-4}$. For ablation and variation studies, we train our models for 50k steps unless otherwise stated. All models are trained with $\tau = 0.1$ with a non-augmented batch size of 96 on a single RTX A5000 GPU unless otherwise specified. We report steps instead of epochs to standardise the amount of data seen during training.

To evaluate pretrained models, we freeze the encoder and discard the projection head. Frozen representations are fed into a 2-layer ReLU-nonlinear MLP for probing on downstream tasks. For probing, we use the Adam optimizer with a learning rate 0.0003 and an early stopping mechanism conditioned on validation loss. For automatic tagging tasks, we report area under receiver-operator curve (AUROC) and mean Average Precision (AP). For classification tasks, we report top-1 accuracy except for key classification: the metric for this task is a weighted score taking into account reasonable errors [45] - We use the `mir_eval` [51] implementation for evaluation. For emotion regression we report R^2 values between predicted and actual values.

5. RESULTS

5.1 Automatic tagging with semi-supervised contrastive learning

We train a self-supervised baseline, a supervised contrastive baseline with and without augmentations, an end-to-end supervised baseline using the sampleCNN architecture, and five variants of our semi-supervised approach with different proportions of labeled data ($p_s \in [0.01, 0.05, 0.1, 0.2, 0.5]$) for Automatic Tagging on MTAT. MTAT labels augment the contrastive matrix \mathbf{M}

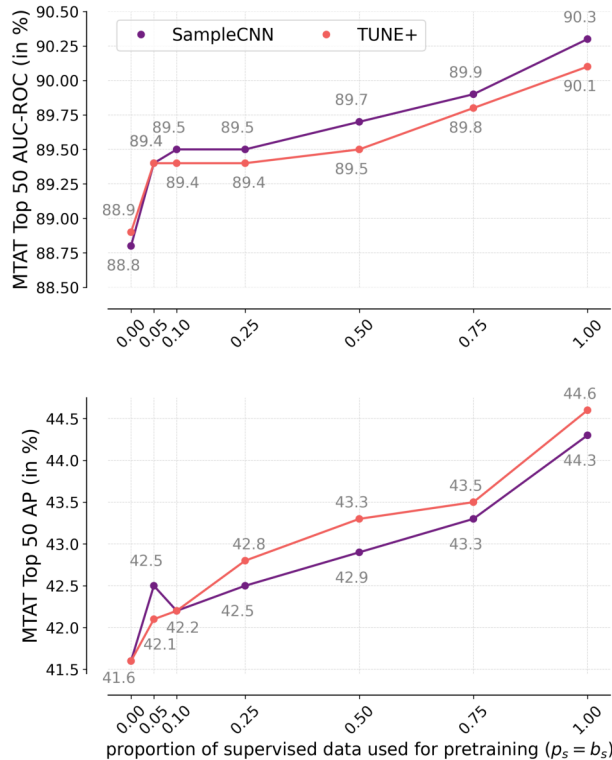


Figure 2: Evolution of AUROC and AP on MTAT probing with proportion of supervised MTAT data used for training.

with positives in the case of supervised or semi-supervised pretraining. We vary the in-batch and global proportion of supervised data b_s and p_s simultaneously. We report results on the same task in the literature in Table 2 for comparable datasets and training scales.

When trained for 200k steps, the supervised contrastive model is competitive with larger self-supervised approaches. Furthermore, it outperforms both our implementation and the results claimed in CLMR for self-supervised contrastive and end-to-end supervised models. Figure 2 shows the influence of $p_s = b_s$ on AUROC and AP. As the proportion of supervised data increases, so does the performance on the downstream evaluation. Including only 5% of labeled data leads to an increase in performance from 88.8 to 89.4 in AUROC. For our experiment with $p_s = 1$ and $b_s = 0.5$, both architectures perform worse than $p_s = b_s = 0.75$, as the model has seen 100k steps of supervised data versus 150k.

5.2 Influence of pretraining labeled dataset

In this experiment, we pre-train multiple semi-supervised models using datasets described in Section 4.1 as support labeled data and FMA as unlabeled data - one model per dataset, each for 50000 steps. We then freeze all models and train shallow MLP probes on all downstream tasks for each model. We train a self-supervised baseline for comparison. Semi-supervised approaches are trained with $b_s = 0.5$ and $p_s = 1$. Table 3 shows these results.

Semi-supervised training on the target dataset always surpasses the self-supervised baseline by a significant mar-

Target Dataset	MTAT		Jamendo			NSynth		Giantsteps	GTZAN	VocalSet		MedleyDB	Emo		
Subset	50	All	50	Genre	Mood	Inst.	Pitch	Inst.	Key	Genre	Tech.	Singer	Inst.	A/V	
Metrics	AUROC						Acc.	Acc _w	Acc.				R _V ² / R _A ²		
Self-Supervised															
FMA	88.4	86.2	80.1	83.3	74.0	71.6	36.8	51.7	13.5	65.5	53.8	71.1	56.5	46.7/71.5	
Semi-Supervised $b_s = 0.5$															
MTAT	50	89.3	86.8	80.0	83.4	73.8	73.3	34.5	46.9	11.3	65.5	53.2	70.0	67.3	44.3/65.9
	All	89.1	87.5	80.3	83.2	74.1	73.0	34.0	51.0	14.9	68.2	52.4	72.9	72.8	41.6/76.2
Jamendo	50	88.6	86.6	81.5	83.4	74.6	72.5	33.8	50.0	14.7	74.1	52.1	71.7	62.0	50.1/ 77.9
	Genre	88.6	86.3	80.5	84.0	74.6	71.5	33.4	50.2	14.6	72.8	52.0	74.6	66.3	48.2/70.3
	Mood	88.3	86.6	81.0	83.0	74.7	72.3	38.2	47.7	14.9	71.3	53.5	71.4	60.9	48.0/73.0
	Instrument	88.4	86.3	80.8	83.1	74.0	71.6	37.2	52.5	14.9	69.3	54.5	67.9	63.0	52.4 /70.6
NSynth	Pitch [†]	88.3	86.3	79.7	82.6	73.5	72.0	79.0	48.6	20.1	65.5	56.9	75.6	64.1	37.5/66.6
	Inst.	88.6	85.7	79.6	82.7	73.3	71.7	26.6	59.6	16.0	67.2	57.3	72.3	66.3	40.3/75.0
GiantSteps	Key [†]	87.7	85.0	79.0	82.1	73.0	70.5	50.8	51.3	22.3	69.3	54.1	71.4	61.2	39.6/63.6
GTZAN	Genre	88.8	86.8	80.9	83.9	74.1	71.5	38.6	46.9	16.3	74.0	53.4	71.7	66.3	28.7/56.4
VocalSet	Technique	88.7	86.7	79.6	82.5	73.3	71.0	46.0	53.5	12.1	63.5	63.0	77.8	67.3	41.5/70.1
	Singer	88.9	86.2	80.1	82.6	73.6	72.8	45.2	52.4	15.3	67.2	54.0	87.1	69.6	54.3/74.6
MedleyDB	Instrument	88.6	87.0	80.2	82.6	73.6	73.8	32.0	48.8	13.2	62.1	58.6	74.3	62.0	41.6/74.8
SOTA		92.7	95.4	84.3	88.0	78.6	78.8	94.4	78.2	74.3	86.9	76.9	87.5	-	61.7/76.3
		[12]	[34]	[13]	[14]	[13]	[52]	[5]	[53]	[54]	[55]	[5,45]	[5,45]		[5,14]
CLMR [45]		89.5		81.3	84.6	73.5	73.5	47.0	67.9	14.8	65.2	58.1	49.9	-	44.4/70.3

Table 3: Results for cross-task evaluation. Models are trained for 50k steps on FMA [39] as the self-supervised dataset and support supervised datasets (rows), and evaluated on target datasets (columns). Giantsteps[†], NSynth[†] are trained without pitch shifting augmentation. Results in bold are the best results obtained for evaluation on a target dataset. SOTA results are included for illustration purposes, but do not necessarily leverage comparable methodologies.

gin when evaluating on the same dataset - with minimal loss of performance on other downstream tasks.

Some complementary tasks improve performance on other downstream datasets, proving semi-supervised contrastive learning a viable transfer learning strategy. Expectedly, training on genre tagging data increases out-of-domain performance on genre classification, instrument tagging on instrument classification, etc. Training on mood data from MTG-Jamendo provides a performance boost on emotion regression. A notable example is the improvement in performance on NSynth pitch when training with key data as support labeled data and *vice versa*. This demonstrates an improvement in the understanding of *pitch* by the model on tasks which are musically related but not trivial transfer learning instances. Most importantly, this occurs without performance loss on general music understanding, *i.e.* automatic tagging. Other musically grounded examples are pitch pretraining improving instrument classification performance and instrument pretraining improving emotion regression performance.

5.3 Robustness to in-domain data corruption

In this section, we evaluate the robustness of our semi-supervised, supervised, and self-supervised contrastive approaches to audio corruptions compared to the end-to-end baseline. We train the probing head without augmentation until convergence and evaluate the model *with* augmentations applied. We design different severity degrees of our augmentation chain (See Subsection 4.2) by applying a modifier to the application probabilities: for severity $s \in [0, 1...4]$, we scale probabilities of application of each augmentation by $s/2$ such that $s = 2$ is the chain applied during training. We sensibly multiply or divide the min and max values of each augmentation hyperparameter (see Table 1) by $s/2$. We then evaluate all models with these augmentation chains on MagnaTagATune. The results are

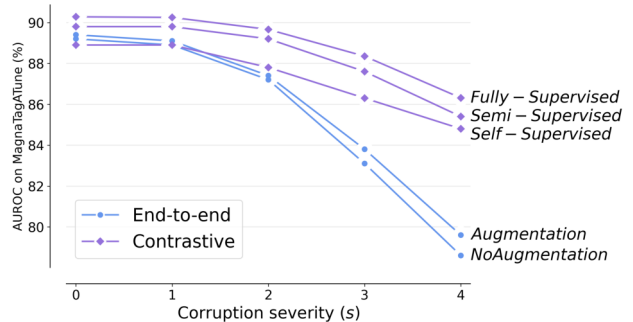


Figure 3: Effect of corruption severity on downstream performance. Contrastive models are more robust than Cross-entropy trained models.

shown in Figure 3. Contrastive approaches are more robust to in-domain corruption than end-to-end approaches - hypothetically because we train contrastive models to be invariant to such transformations through the augmentation chain - which is not an objective of the end-to-end supervised approach.

5.4 Multilabel positive mining strategy

In this experiment, we test multiple label-based positive mining strategies. First by varying the number of common labels for mining positives - *i.e.* $C \in \{1, 2, 4, 6\}$. Further, we explore the “semantic weighing” strategy described in Section 3.3.1, in which the target similarity between two tracks is weighed by the number of common labels and the total number of labels. We test these strategies on both semi-supervised and supervised contrastive models. Results are reported in Table 4.

For supervised approaches, the continuous target produced by semantic weighing produces the best results, on par with 4x training steps with a criterion $C = 1$ (as shown in Table 2). In the supervised case, as the criterion in-

Positive strategy Class criterion	Supervised		Semi-Supervised	
	AUROC	AP	AUROC	AP
$C = 1$	90.1	44.2	89.3	41.3
$C = 2$	90.1	43.9	89.0	41.6
$C = 4$	89.3	42.8	89.0	41.3
$C = 6$	88.9	42.3	89.0	41.5
Weighing	90.6	45.3	88.9	41.6

Table 4: Multilabel positive mining strategy as described in Section 3.3.1.

creases, performance deteriorates. We hypothesise that this could be because it is an *easier* task for the model to discern that two tracks with many common tags are similar (higher C), as they likely share many attributes, therefore providing a weaker training signal. Understanding what links two tracks from a single tag is more challenging and appears to yield more robust representations. The continuous “relative similarity” target created by the weighing strategy is a more nuanced task and appears to be a stronger supervision signal. This guides the model towards more robust representations, which explains the higher performance. In the semi-supervised case, we speculate that the binary self-supervision signal overpowers the continuous target as a less nuanced objective with harsher penalties for failure. These penalties could overpower softer penalties from the continuous target in the loss, preventing optimal convergence. Future work should focus on understanding and reconciling these aspects of the semi-supervised approach to leverage other continuous signals.

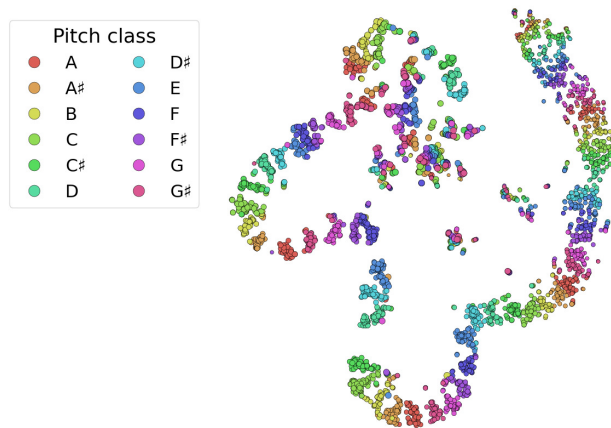
5.5 Qualitative analysis

The results reported in Section 5.2 show that performance on downstream tasks improves when labels from a related task are used for model training, with minimal loss of performance on other tasks. We hypothesise that the internal latent representations are given structure relative to the supervision signal while maintaining the semantic structure given by the self-supervision signal. To illustrate this, we perform t-SNE dimension reduction on embeddings produced by the semi-supervised model from Table 3 trained with NSynth (Figure 4a) as support labeled data and fully self-supervised (Figure 4b) evaluated on the test set of NSynth-pitch.

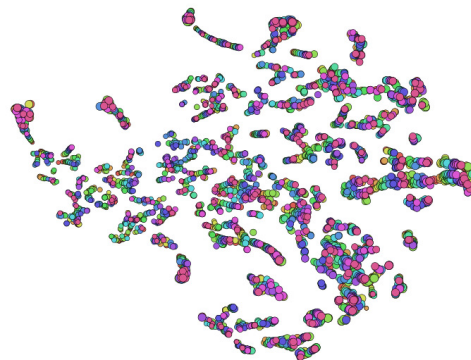
In the set of Figures 4, the latent spaces for the NSynth test set produced by these two models are shown. When pretrained on NSynth-pitch, the latent space is highly organized. Separability by class is much clearer than when pretrained on FMA. We notice that several musical structures emerge in this latent space. Notably, octaves go from low to high clockwise. Pitches that are “similar” are close together, i.e., semitones and octaves of the same pitch class.

6. CONCLUSION AND FUTURE WORK

We presented SemiSupCon, a simple method for leveraging both supervision and self-supervision signals in contrastive representation learning. By leveraging reduced amounts of labeled data during pretraining, SemiSupCon outperforms end-to-end comparable supervised baselines



(a) Latent embeddings of the NSynth-pitch test set from a semi-supervised model trained on FMA+NSynth-pitch



(b) Latent embeddings of the NSynth-pitch test set from a self-supervised model trained on FMA

Figure 4: Exploration of the NSynth pitch latent space. Octaves are denoted by size and pitch class by the color of the dot. Each dot is a full audio sample

on downstream tasks. We find that SemiSupCon is more robust to data corruption at inference compared to end-to-end supervised methods. Additionally, SemiSupCon can utilize various supervision signals with minimal performance loss on out-of-domain tasks and achieve performance transfer on similar tasks. While performance gains might seem moderate on automatic tagging for instance, other downstream tasks show more distinct improvements. Furthermore, the contrastive objective can lead to explicitly structured latent spaces with emergent musical structures - enhancing the musical interpretability of latent spaces by design of the support supervision signal - i.e. labeling small amounts of data.

Future work will focus on exploring additional supervision signals and tasks such as perceptual metrics, tempo estimation, and chord estimation. Other avenues include leveraging the low-data proficiency of SemiSupCon for human-in-the-loop representation learning. The architecture of SemiSupCon being very flexible, it can be further adapted to multimodal approaches or hierarchical representation learning. A more comprehensive exploration of the influence of the proportion of labeled data and the exact effect of labels and contrastive matrix sparsity on downstream performance will also be undertaken.

7. ACKNOWLEDGMENT

This work is supported by the EPSRC UKRI Centre for Doctoral Training in Artificial Intelligence and Music (EP/S022694/1) and Universal Music Group.

8. REFERENCES

- [1] T. Chen, S. Kornblith, M. Norouzi *et al.*, “A simple framework for contrastive learning of visual representations,” in *International conference on machine learning (ICML)*. PMLR, 2020, pp. 1597–1607.
- [2] A. v. d. Oord, Y. Li, and O. Vinyals, “Representation learning with contrastive predictive coding,” *arXiv preprint arXiv:1807.03748*, 2018.
- [3] J.-B. Grill, F. Strub, F. Altché *et al.*, “Bootstrap your own latent—a new approach to self-supervised learning,” *Advances in neural information processing systems*, vol. 33, pp. 21 271–21 284, 2020.
- [4] X. Chen and K. He, “Exploring simple siamese representation learning,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, 2021, pp. 15 750–15 758.
- [5] L. Yizhi, R. Yuan, G. Zhang, Y. Ma, X. Chen, H. Yin, C. Xiao, C. Lin, A. Ragni, E. Benetos *et al.*, “MERT: Acoustic music understanding model with large-scale self-supervised training,” in *The Twelfth International Conference on Learning Representations (ICLR)*, 2023.
- [6] Y. Gong, C.-I. Lai, Y.-A. Chung *et al.*, “SSAST: Self-supervised audio spectrogram transformer,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 10, 2022, pp. 10 699–10 709.
- [7] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhota, R. Salakhutdinov, and A. Mohamed, “HuBERT: Self-supervised speech representation learning by masked prediction of hidden units,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3451–3460, 2021.
- [8] J. Spijkervet and J. A. Burgoyne, “Contrastive Learning of Musical Representations,” in *Proc. of International Society for Music Information Retrieval (ISMIR)*, 2021, pp. 673–681.
- [9] M. A. V. Vásquez and J. A. Burgoyne, “Tailed U-Net: Multi-scale music representation learning,” in *Proc. of International Society for Music Information Retrieval (ISMIR)*, 2022, pp. 67–75.
- [10] H. Zhao, C. Zhang, B. Zhu *et al.*, “S3T: Self-supervised pre-training with swin transformer for music classification,” in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 606–610.
- [11] Y. Li, R. Yuan, G. Zhang, Y. MA, C. Lin, X. Chen, A. Ragni, H. Yin, Z. Hu, H. He *et al.*, “MAP-Music2Vec: A simple and effective baseline for self-supervised music audio representation learning,” in *Proc. of International Society for Music Information Retrieval (ISMIR)*, 2022.
- [12] M. Won, Y.-N. Hung, and D. Le, “A foundation model for music informatics,” in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 1226–1230.
- [13] M. C. McCallum, F. Korzeniowski, S. Oramas, F. Gouyon, and A. Ehmann, “Supervised and unsupervised learning of audio representations for music understanding,” in *Proc. of International Society for Music Information Retrieval (ISMIR)*, 2022, pp. 256–263.
- [14] R. Castellon, C. Donahue, and P. Liang, “Codified audio language modeling learns useful representations for music information retrieval,” in *Proc. of International Society for Music Information Retrieval (ISMIR)*, 2021, pp. 88–96.
- [15] I. Manco, E. Benetos, E. Quinton, and G. Fazekas, “Contrastive audio-language learning for music,” in *Proc. of International Society for Music Information Retrieval (ISMIR)*, 2022, pp. 640–649.
- [16] T. Xiao, X. Wang, A. A. Efros, and T. Darrell, “What should not be contrastive in contrastive learning,” in *The 9th International Conference on Learning Representations (ICLR)*, 2021.
- [17] Y. Tian, C. Sun, B. Poole, D. Krishnan, C. Schmid, and P. Isola, “What Makes for Good Views for Contrastive Learning?” in *Advances in Neural Information Processing Systems*, vol. 33. Curran Associates, Inc., 2020, pp. 6827–6839.
- [18] J. Choi, S. Jang, H. Cho *et al.*, “Towards proper contrastive self-supervised learning strategies for music audio representation,” in *2022 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2022, pp. 1–6.
- [19] M. C. McCallum, M. E. Davies, F. Henkel, J. Kim, and S. E. Sandberg, “On the effect of data-augmentation on local embedding properties in the contrastive learning of music audio representations,” *arXiv preprint arXiv:2401.08889*, 2024.
- [20] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, “Supervised contrastive learning,” *Advances in neural information processing systems*, vol. 33, pp. 18 661–18 673, 2020.
- [21] H. Al-Tahan and Y. Mohsenzadeh, “CLAR: Contrastive learning of auditory representations,” in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2021, pp. 2530–2538.

- [22] Y.-A. Chung, Y. Zhang, W. Han *et al.*, “W2V-BERT: Combining contrastive learning and masked language modeling for self-supervised speech pre-training,” in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2021, pp. 244–250.
- [23] E. Fonseca, D. Ortego, K. McGuinness *et al.*, “Un-supervised contrastive learning of sound event representations,” in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 371–375.
- [24] D. Yao, Z. Zhao, S. Zhang *et al.*, “Contrastive Learning with Positive-Negative Frame Mask for Music Representation,” in *Proceedings of the ACM Web Conference 2022*, Apr. 2022, pp. 2906–2915.
- [25] C. Garoufis, A. Zlatintsi, and P. Maragos, “Multi-Source Contrastive Learning from Musical Audio,” no. arXiv:2302.07077. arXiv, May 2023.
- [26] H. Guo and L. Shi, “Ultimate Negative Sampling for Contrastive Learning,” in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Jun. 2023, pp. 1–5.
- [27] S. Ge, S. Mishra, C.-L. Li, H. Wang, and D. Jacobs, “Robust Contrastive Learning Using Negative Samples with Diminished Semantics,” in *Advances in Neural Information Processing Systems*, vol. 34. Curran Associates, Inc., 2021, pp. 27 356–27 368.
- [28] T. Huynh, S. Kornblith, M. R. Walter, M. Maire, and M. Khademi, “Boosting Contrastive Self-Supervised Learning with False Negative Cancellation,” in *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. Waikoloa, HI, USA: IEEE, Jan. 2022, pp. 986–996.
- [29] T. Akama, H. Kitano, K. Takematsu *et al.*, “Auxiliary self-supervision to metric learning for music similarity-based retrieval and auto-tagging,” *PLOS ONE*, vol. 18, no. 11, p. e0294643, Nov. 2023.
- [30] P. Manocha, Z. Jin, R. Zhang *et al.*, “CDPAM: Contrastive Learning for Perceptual Audio Similarity,” in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Jun. 2021, pp. 196–200.
- [31] P. Alonso-Jiménez, X. Favory, H. Foroughmand, G. Bourdalas, X. Serra, T. Lidy, and D. Bogdanov, “Pre-training strategies using contrastive learning and playlist information for music classification and similarity,” in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.
- [32] X. Favory, K. Drossos, T. Virtanen, and X. Serra, “Coala: Co-aligned autoencoders for learning semantically enriched audio representations,” in *Self-supervision in Audio and Speech Workshop, International Conference on Machine Learning (ICML)*, 2020.
- [33] A. Ferraro, X. Favory, K. Drossos *et al.*, “Enriched Music Representations with Multiple Cross-modal Contrastive Learning,” *IEEE Signal Processing Letters*, vol. 28, pp. 733–737, 2021.
- [34] Q. Huang, A. Jansen, J. Lee, R. Ganti, J. Y. Li, and D. P. Ellis, “Mulan: A joint embedding of music audio and natural language,” in *Proc. of International Society for Music Information Retrieval (ISMIR)*, 2022.
- [35] B. Kim, J. Choo, Y.-D. Kwon, S. Joe, S. Min, and Y. Gwon, “Selfmatch: Combining contrastive self-supervision and consistency for semi-supervised learning,” in *34th conference on Neural Information Processing Systems (NEURIPS) Workshop*, 2021.
- [36] Y. Zhang, X. Zhang, J. Li, R. Qiu, H. Xu, and Q. Tian, “Semi-supervised contrastive learning with similarity co-calibration,” *IEEE Transactions on Multimedia*, 2022.
- [37] F. Yang, K. Wu, S. Zhang, G. Jiang, Y. Liu, F. Zheng, W. Zhang, C. Wang, and L. Zeng, “Class-aware contrastive semi-supervised learning,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, 2022, pp. 14 421–14 430.
- [38] E. Law, K. West, M. I. Mandel, M. Bay, and J. S. Downie, “Evaluation of algorithms using games: The case of music tagging,” in *Proc. of International Society for Music Information Retrieval (ISMIR)*, 2009, pp. 387–392.
- [39] M. Defferrard, K. Benzi, P. Vandergheynst, and X. Bresson, “FMA: A dataset for music analysis,” in *Proc. of International Society for Music Information Retrieval (ISMIR)*, 2017, pp. 316–323.
- [40] D. Bogdanov, M. Won, P. Tovstogan *et al.*, “The mtg-jamendo dataset for automatic music tagging,” in *Machine Learning for Music Discovery Workshop, International Conference on Machine Learning (ICML)*, 2019.
- [41] J. Engel, C. Resnick, A. Roberts, S. Dieleman, M. Norouzi, D. Eck, and K. Simonyan, “Neural audio synthesis of musical notes with wavenet autoencoders,” in *International Conference on Machine Learning*. PMLR, 2017, pp. 1068–1077.
- [42] R. M. Bittner, J. Salamon, M. Tierney, M. Mauch, C. Cannam, and J. P. Bello, “Medleydb: A multi-track dataset for annotation-intensive mir research,” in *Proc. of International Society for Music Information Retrieval (ISMIR)*, 2014, pp. 155–160.
- [43] R. M. Bittner, J. Wilkins, H. Yip, and J. P. Bello, “Medleydb 2.0: New data and a system for sustainable data collection,” *ISMIR Late Breaking and Demo Papers*, p. 36, 2016.

- [44] P. Knees, Á. Faraldo Pérez, H. Boyer, R. Vogl, S. Böck, F. Hörschläger, M. Le Goff *et al.*, “Two data sets for tempo estimation and key detection in electronic dance music annotated from user corrections,” in *Proc. of International Society for Music Information Retrieval (ISMIR)*, 2015, pp. 364–70.
- [45] R. Yuan, Y. Ma, Y. Li, G. Zhang, X. Chen, H. Yin, Y. Liu, J. Huang, Z. Tian, B. Deng *et al.*, “Marble: Music audio representation benchmark for universal evaluation,” *Advances in Neural Information Processing Systems*, vol. 36, 2024.
- [46] G. Tzanetakis and P. Cook, “Musical genre classification of audio signals,” *IEEE Transactions on speech and audio processing*, vol. 10, no. 5, pp. 293–302, 2002.
- [47] B. L. Sturm, “The GTZAN dataset: Its contents, its faults, their effects on evaluation, and its future use,” *arXiv preprint arXiv:1306.1461*, 2013.
- [48] J. Wilkins, P. Seetharaman, A. Wahl, and B. Pardo, “Vocalset: A singing voice dataset.” in *Proc. of International Society for Music Information Retrieval (ISMIR)*, 2018, pp. 468–474.
- [49] M. Soleymani, M. N. Caro, E. M. Schmidt, C.-Y. Sha, and Y.-H. Yang, “1000 songs for emotional analysis of music,” in *Proceedings of the 2nd ACM international workshop on Crowdsourcing for multimedia*, 2013, pp. 1–6.
- [50] J. Lee, J. Park, K. L. Kim, and J. Nam, “Samplecnn: End-to-end deep convolutional neural networks using very small filters for music classification,” *Applied Sciences*, vol. 8, no. 1, p. 150, 2018.
- [51] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, D. P. Ellis, and C. C. Raffel, “Mir_eval: A transparent implementation of common mir metrics.” in *Proc. of International Society for Music Information Retrieval (ISMIR)*, 2014, p. 2014.
- [52] P. Alonso-Jiménez, X. Serra, and D. Bogdanov, “Music representation learning based on editorial metadata from discogs,” in *Proc. of International Society for Music Information Retrieval (ISMIR)*, 2022, pp. 825–33.
- [53] L. Wang, P. Luc, Y. Wu *et al.*, “Towards Learning Universal Audio Representations,” in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2022, pp. 4593–4597.
- [54] F. Korzeniowski and G. Widmer, “End-to-end musical key estimation using a convolutional neural network,” in *2017 25th European Signal Processing Conference (EUSIPCO)*. IEEE, 2017, pp. 966–970.
- [55] P. Alonso-Jiménez, L. Pepino, R. Batlle-Roca, P. Zinemanas, D. Bogdanov, X. Serra, and M. Rocamora, “Leveraging pre-trained autoencoders for interpretable prototype learning of music audio,” *arXiv preprint arXiv:2402.09318*, 2024.

IMPROVED SYMBOLIC DRUM STYLE CLASSIFICATION WITH GRAMMAR-BASED HIERARCHICAL REPRESENTATIONS

Léo Géré¹ Nicolas Audebert^{1,2} Philippe Rigaux¹

¹ Conservatoire national des arts et métiers, CEDRIC, F-75141 Paris, France

² Univ. Gustave Eiffel, ENSG, IGN, LASTIG, F-94160 Saint-Mandé, France

leo.gere@lecnam.net, nicolas.audebert@ign.fr, philippe.rigaux@cnam.fr

ABSTRACT

Deep learning models have become a critical tool for analysis and classification of musical data. These models operate either on the audio signal, *e.g.* waveform or spectrogram, or on a symbolic representation, such as MIDI. In the latter, musical information is often reduced to basic features, *i.e.* durations, pitches and velocities. Most existing works then rely on generic tokenization strategies from classical natural language processing, or matrix representations, *e.g.* piano roll. In this work, we evaluate how enriched representations of symbolic data can impact deep models, *i.e.* Transformers and RNN, for music style classification. In particular, we examine representations that explicitly incorporate musical information *implicitly* present in MIDI-like encodings, such as rhythmic organization, and show that they outperform generic tokenization strategies. We introduce a new tree-based representation of MIDI data built upon a context-free musical grammar. We show that this grammar representation accurately encodes high-level rhythmic information and outperforms existing encodings on the GrooveMIDI Dataset for drumming style classification, while being more compact and parameter-efficient.

1. INTRODUCTION

In the last few years, machine learning (ML) has significantly changed how the Music Information Retrieval (MIR) community deals with tasks such as style and composer classification, music generation, pitch and rhythm detection, etc. Yet, training deep learning models on music raises the question of the representation of this data. Depending on the input format (audio, MIDI, musical score. . .), different representations, *i.e.* different *encodings*, are possible. Each encoding has advantages and drawbacks: some representations, *e.g.* waveforms, focus on raw low-level acoustic features, while others, *e.g.* sheet music, encode high-level abstract semantics of the musical language.

While deep neural networks had great success on audio signal, *i.e.* waveforms and spectrograms, machine learning

for symbolic MIDI remains understudied. In this work, we seek to build effective representations of symbolic music, with a focus on recorded MIDI performances. Multiple possible representations of MIDI music coexist in the literature. Most of them contains only low-level information, such as the timing of the onset and the offset of each note and their velocity. This is due to practical constraints: typical MIDI recordings usually do not contain any information about tonality, tempo, time-signature or rhythm. Hence, a model trained on such MIDI samples typically needs to allocate a part of its weights to extract these relevant high-level features from the data. Building better representations of MIDI data to encode semantic musical information could therefore be beneficial to the training of deep models and their efficiency, as they could directly focus on using these features rather than extracting them from the data first.

Music classification has been a task of choice for MIDI performances. Preliminary works from [1] in 2007 encoded MIDI as strings and used Kolmogorov complexity to compare music pieces. [2] introduced jSymbolic, a library to extract high level features from MIDI files, such as pitch histograms, a line of work extended by `music21` [3] and `musif` [4]. As new MIDI datasets have been introduced for composer [5] and style classification [6], efforts have been made to evaluate how MIDI representations affect deep models. [7] introduced MidiTok, a tokenization framework to encode MIDI files as a sequence of tokens, suitable for Transformers and Recurrent Neural Networks (RNN). More recently, [8] compared different neural architectures for various MIDI encodings: Convolutional Neural Networks (CNN) trained on Piano rolls, Transformers trained on sequences of tokens, and Graph Neural Networks (GNN) trained on graphs extracted from MIDI files.

In this line of work, we aim to design a representation of MIDI files that is both efficient and discriminative for classification tasks, by incorporating high level musical information directly in the preprocessing. To do so, we explore a new representation based on the *rhythmic tree* structure, built from a context-free grammar tailored to symbolic music. We show that this representation outperforms existing encodings, such as tokenizations or piano rolls, on a drumming style classification built upon the GrooveMIDI Dataset [6]. In addition, our rhythmic tree-based encoding results in smaller deep models, with less parameters, able to be trained on less data compared to existing representations.



© L. Géré, N. Audebert, and P. Rigaux. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).
Attribution: L. Géré, N. Audebert, and P. Rigaux, “Improved symbolic drum style classification with grammar-based hierarchical representations”, in *Proc. of the 25th Int. Society for Music Information Retrieval Conf.*, San Francisco, United States, 2024.

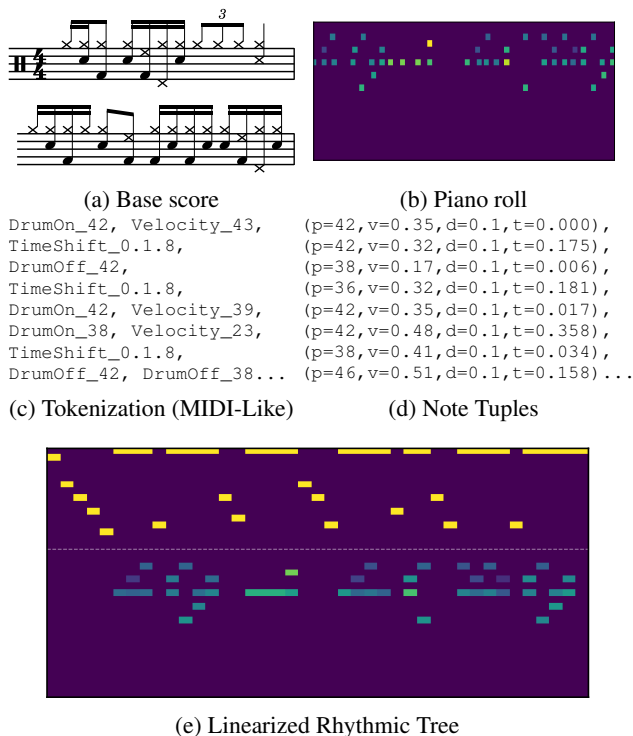


Figure 1: Different representations of the same two bars of drums. Score (a) is present for reference only.

2. BACKGROUND

2.1 MIDI Representations

MIDI is a lightweight musical information exchange format. It does not carry audio data, but only timestamped events, *e.g.* a note being played, featuring its pitch and velocity, a note being released, a pedal change, etc. It is suitable for recording as it captures the performer’s expressiveness, but does not require metadata that are found in a score, such as tempo, time-signature¹, tonality and voices [9]. We discuss below the most common MIDI representations for ML.

2.1.1 Piano Roll

The piano roll is a visual representation of MIDI files inspired by the analog rolls for piano players. It consists in a 2D matrix with one dimension for pitches, and one for time. A note at pitch p with a NOTE_ON event at x_{on} and NOTE_OFF event at x_{off} is given a positive value at positions $(x, p)_{x \in [x_{on}, x_{off}]}$, as shown in Figure 1b. Often, the value in a matrix cell is one of the properties of the MIDI event, *e.g.* the velocity. This representation is popular, as its 2D structure allows to easily adapt deep models inspired by image processing (*e.g.* CNN) to music tasks [10–12]. However, it can result in large sparse matrices with many zeros, since the time dimension must be discretized with a time step smaller than the shortest MIDI event. In addition, piano rolls tend to be very long and redundant, since many successive vectors will be identical.

¹ MIDI recordings can contain tempo and time-signature, but only through manual addition *a posteriori*.

2.1.2 Sequence of Tokens or Notes

Similar to Natural Language Processing (NLP) techniques, recent works have adopted sequence-like representations, especially suitable for RNN and Transformers architectures. They encode MIDI files as sequences of events. These events are in turn transformed into *tokens*, *i.e.* discrete values from a vocabulary V . Many tokenizations exist, some consisting in a simple token/event mapping with MIDI files (MIDI-Like [13, 14], see Figure 1c), while others include note durations (Structured [15], TSD [16]). More sophisticated tokenizers include higher level information about bar and position in the bar, such as REMI [17].

Finally, MIDI files can be represented as “note tuples”, *i.e.* sequences of notes with attributes. For example, [18] represents each note by a set of four values: pitch, velocity, duration and time-shift compared to the previous note (cf. Figure 1d). This representation is much more compact than piano rolls or sequence of tokens.

2.2 Formal Grammar

This work designs a symbolic music representation for deep networks based on a grammar-based rhythmic tree. As a starting point, a formal grammar defines the syntax of a language L . It consists in a set of symbols, associated with production rules used to rewrite non-terminal symbols into other (non-)terminal symbols. Applied successively, those rules can produce every possible sentence of L .

2.2.1 Context-Free Grammar

Succinctly, a context-free grammar [19] is a type of formal grammar for which the production rules do not depend on other context than the left-hand-side symbol. It is defined as a 4-tuple $G = (V, \Sigma, R, S)$. V is a finite set of non-terminal symbols, including the special *start symbol* S . Σ is a finite set of terminal symbols, called the alphabet. Finally, R is a finite set of production rules of the form $a \rightarrow b$, where $(a, b) \in V \times (V \cup \Sigma)^*$ in which $*$ denotes the Kleene star operator, *i.e.* a pattern repeated of 0, 1 or more times.

The application of a sequence of rules can be represented as a tree, in which the parent node is represented by the left-hand-side of each rule, and the child nodes are the symbols on the right-hand-side. Once every non-terminal symbol has been resolved into a terminal symbol, we obtain a *parse tree* representing the structure of a sentence of L according to G , with elements of Σ as leaves, and S as root.

2.2.2 Musical Grammar

In a homophonic musical score (monophonic voice that can include chords [20]), rhythm can be represented as a tree [21, 22]. For example, in a 4/4 music piece, a measure could be split into two half notes. Then, each half note can be further divided into two quarter notes, or into a triplet of quarter notes, etc. The `qparse` library [23] is a MIDI-to-score transcription framework that produces a sheet music by parsing a MIDI file with a weighted context-free grammar and dynamic programming, with applications *e.g.* to automatic drum transcription [24]. While designed for a handcrafted music transcription algorithm, the intermediate

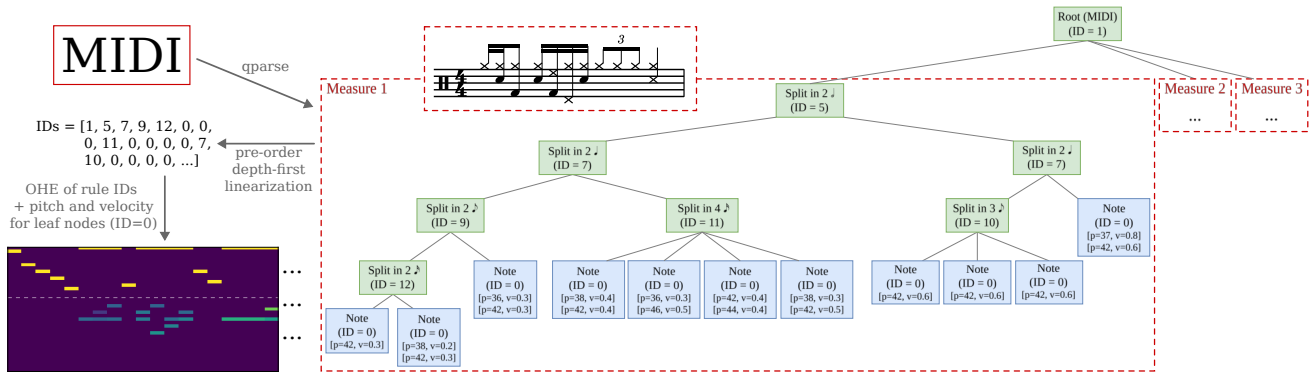


Figure 2: Example of tree built by `qparse` after rules simplification and re-rooting of measures (right), with its associated linearization and vector representation (left). In the matrix, the part above the dashed line contains the one-hot encoded rules (blue/yellow for 0/1), and the one below contains the playing instruments for terminal nodes (color representing velocity).

parsing tree computed by `qparse` contains rich rhythmic information that is also valuable as an input to deep models. Note that, while we our work uses `qparse` to obtain rhythmic trees, our contribution lies in evaluating this tree representation of music, regardless of its construction. We expect our representation to generalize to other parsers.

3. METHODOLOGY

3.1 Linearized Rhythmic Tree

To build our high-level MIDI representation, we linearize a rhythmic tree obtained using a context-free grammar, enriched by information about pitch and velocity in leaves. We call this representation Linearized Rhythmic Tree (LRT). To achieve this goal, we leverage the transcription framework `qparse` [23] to extract its internal intermediate rhythmic tree representation. Note that we only consider homophonic inputs since this is what `qparse` MIDI grammar supports. `qparse` needs the time-signature and the tempo of the track (because measures are parsed separately), as well as the specification of a weighted grammar. We use a rhythm-oriented grammar similar to [24], detailed in appendix.

As described in [25], the root of the intermediate rhythmic tree is the first measure. Its left child is a tree describing its beat decomposition, and its right child is a node pointing at the root of the next measure. We rewrite this tree so that all measures are children of the same global root. A n -measures-long track will therefore have a root with n children. This rewriting allows us to reduce the maximum depth of the rhythmic tree, which would otherwise grow linearly with n . The resulting tree is shown on the right of Figure 2. In this tree, each node is labeled by the identifier of the associated production rule in the grammar.² Each leaf is a terminal symbol, labeled by the note and properties from the associated MIDI events, *i.e.* pitch³ and velocity. Note that multiple instruments can be playing at the same time, so a leaf can be associated to several events.

As an example, in Figure 2, the first bar (red frame) is split in two sections, each of half note length (rule 5). Then,

² See the ruleset with IDs in Section 2 of the supplementary material.

³ In the case of drums, the “pitch” corresponds to the drum used, *e.g.* cymbal, snare, tom, etc.

the first half gets split into two quarter-length sections (rule 7). The second child of this node, a quarter note, is split into four sixteenth notes (rule 11). Finally, each of those sixteenth notes leads to a terminal symbol (rule 0), with MIDI events attached to it, *e.g.*, the second child has two `NOTE_ON` events, respectively with pitch 36 and velocity 0.3, and with pitch 46 and velocity 0.5.

As we cannot directly feed the tree structure to the models, we first linearize it using a pre-order depth-first traversal: we start from root, and traverse the nodes recursively following the left-most child, only going back up when the current branch has been fully traversed. This produces a sequence of nodes containing the identifier of their rule in the grammar, as well as, in the case of leaves, the list of playing instruments and their velocity. We encode every node into a $d = (m + n)$ -dimensional vector. m is the number of rules in the grammar, and the first part of the vector is the one-hot-encoded identifier of the production rule associated with the node. n is the number of possible instruments, and the second part of the vector contains the normalized velocity for each instrument. If an instrument is not playing for this note, its velocity is set to zero. For non-terminal rules, this second part is entirely zero. This linearization results in the matrix on the left of Figure 2, *i.e.* a sequence $S = \{s\}_{t \in [1, T]}$ where $s_t \in \mathbb{R}^d$ is the vector associated to a node, and T is the total number of nodes. Therefore, our linearized rhythmic tree results in a multi-dimensional sequence S , that can be fed in all usual deep models such as RNN and Transformers.

Note that this representation is significantly shorter than tokenizations or piano rolls. In average, the sequences are only around 18% longer than note tuples, while containing much more information about the rhythm structure.

3.2 Tree-based Positional Encoding for Transformers

While RNN can model the position in the sequence through their hidden state, Transformers process sequences as a bag of words, without any positional information. To overcome this issue, positional encoding [26] was introduced to incorporate information about the position of an element in the Transformer model.

Classical positional encoding [26] creates a vector PE of dimension d using sine and cosine functions of increasing frequencies:

$$\omega_{pos,i} = \frac{pos}{\tau \left(\frac{2^i}{d}\right)}, \quad \text{PE}(pos, 2i) = \sin(\omega_{pos,i}) \quad (1)$$

$$\text{PE}(pos, 2i+1) = \cos(\omega_{pos,i})$$

where pos is the position of the element in the sequence, d the size of the embedding, $i \in \llbracket 1, d/2 \rrbracket$ the dimension, and $\tau = 10000$ as in [26].

3.2.1 Continuous Positional Encoding

For musical data, this positional encoding is not related to the temporal organization of the notes. Depending on how the sequence S was built, the position pos of an element can be arbitrary, such as *e.g.* tokenizations where a note is split into several tokens for pitch, velocity and duration, or note tuples where two simultaneous notes can be interchanged. For encoding note tuples, we therefore introduce a continuous positional encoding that replaces the position in the sequence by the timestamp of the note in the track:

$$\omega_{t,i} = \frac{2\pi}{T_S} \cdot \frac{t}{(T_L/T_S)^{\frac{2^i}{d}}}, \quad \text{PE}(t, 2i) = \sin(\omega_{t,i}) \quad (2)$$

$$\text{PE}(t, 2i+1) = \cos(\omega_{t,i})$$

where t is the absolute starting time of the note in seconds and T_S and T_L are respectively the smallest and largest periods of the sine functions. This encoding allows two simultaneous notes to share the same positional encoding.

3.2.2 Tree-based Positional Encoding

A downside of linearizing the rhythmic tree is that we lose the explicit hierarchical structure between a parent node and its children. The structure is still implicitly encoded in the linearized sequence S in the rule identifiers, but the model would have to learn how the grammatical rules operate to rebuild the tree and leverage its structure.

To better represent the rhythmic tree, we use a hierarchical tree-based positional encoding (TBPE) that encodes the position of a node *in the tree*, rather than its position in the linearized sequence. Some TBPE have been proposed in the literature, *e.g.* for code translation to help Transformers process abstract syntax trees [27, 28]. Since our trees are bounded in depth at d_{\max} , we associate to each node \mathcal{N} a vector of size $2d_{\max}$ that represents the path to a node from the root of the tree. This process is illustrated in Figure 3. Element k represents the index of the child traversed at depth k , while element $k + d_{\max}$ is the total number of children of the parent node at depth k . For example, to reach node F , we go through node R (child #1 over 1), then node A (child #1 over 4), then node F (child #2 over 2). If the depth of the node \mathcal{N} is less than d_{\max} , then the remaining elements of the vector are padded with zeros. This makes explicit in the positional encoding the *parent* \rightarrow *child* relations, along with depth and breadth properties. It becomes easier for the model to understand that notes can belong to a larger structures (*e.g.* triplet or four semiquavers).

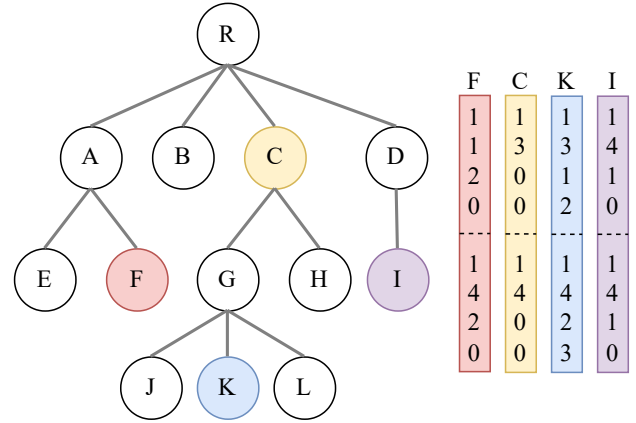


Figure 3: Example of tree-based positional encoding (TBPE) for a tree of maximum depth $d_{\max} = 4$.

4. EXPERIMENTS AND RESULTS

4.1 Dataset and Task

Our models are trained and evaluated for style classification on the Groove MIDI Dataset (GMD) [6]. It consists in 13.6 hours of drumming music, played by humans with a metronome. Each track is labelled with a style provided by the drummer, alongside tempo and time-signature. The dataset is composed of long sequences (few minutes) and short beats and fills. We only consider long sequences, as short sequences are less representative of a specific style. We also discard non-4/4 tracks (around 1% of the dataset), as we use a 4/4 musical grammar, and a few tracks that `qparse` failed to parse⁴. We focus on the 4 most represented styles: funk, jazz, latin and rock. The final subset contains 326 tracks, representing 7.5 hours of drumming, split into the train/validation/test sets (80%/10%/10%) as the original dataset [6]. Each track is then further divided into multiple chunks of n measures with a sliding window.

4.2 Representations

In addition to our LRT, we evaluate common representations of MIDI data for style classification.

Piano Roll We sample the MIDI data at frequency f . We compare $f = 30$ Hz ≈ 33.3 ms per time step, as 30 ms is considered as the simultaneity threshold for the human ear [29], and $f = 50$ Hz = 20 ms per time step, to see if models would improve with finer granularity, at the expense of sequence length. Every time step is represented by a vector in $v \in [0, 1]^{22}$. Each dimension represents one of the 22 instruments of the drum kit. v_i encodes the velocity of the i -th instrument, normalized between 0 and 1 using maximum normalization. Note that the duration of notes in drums MIDI files is arbitrary, as only onset and velocity matter. All durations are set to 100 ms in the Groove MIDI dataset. In our dataset, the average length of a piano roll is around 2455 for $f = 30$ Hz, and 4092 for $f = 50$ Hz.

Sequence of Tokens We experiment with various tokenizers from the literature, that quantify velocities and tim-

⁴ As these tracks are only in the train and validation sets, this does not affect the fairness of the final comparison.

Representation used			LSTM			Transformer		
Type	Variation	Avg. len.	Test F1 score	# params.	# bars	Test F1 score	# params.	# bars
Piano roll	50 steps/second	4092	<i>0.618 ± 0.033</i>	576 522	4	0.545 ± 0.011	253 130	2
	30 steps/second	2455	0.663 ± 0.023	552 458	4	0.486 ± 0.026	20 330	4
Note Tuple	-	733	0.568 ± 0.024	555 530	8	0.492 ± 0.014	216 506	2
Tokenization	MIDI-Like	2767	0.565 ± 0.026	42 442	4	0.576 ± 0.011	360 586	8
	REMI	2502	0.475 ± 0.051	38 282	8	0.517 ± 0.028	17 626	4
	Structured	2646	0.599 ± 0.014	282 826	2	<i>0.598 ± 0.011</i>	232 162	8
	TSD	2464	0.487 ± 0.029	23 274	8	0.486 ± 0.032	20 178	4
LRT	Simple linearization	863	0.603 ± 0.014	1 358 346	8	0.556 ± 0.037	230 378	8
	With TBPE	863	0.596 ± 0.014	252 170	4	0.660 ± 0.019	88 138	4

Table 1: Performance of the different representations and model combinations on the GrooveMIDI dataset. We report macro F1 scores on the test set for the best model of each couple model/representation, alongside the model’s number of parameters, the length (in bars) of input samples, and the average sequence length of each representation. Best results for each model type are in **bold**, second best in *italics*.

ings to limit the size of the vocabulary: MIDI-Like [13, 14], TSD [16], Structured [15] and REMI [17] tokenizers. We use the default parameters from [7], except for pitch range which is set to the min/max instrument ID from the GMD. Models trained on tokenizations use a 64-dimensional embedding, as recommended in [8]. Akin to piano rolls, tokenizers produce sequences with 2400 to 2800 elements.

Note Tuples We also consider the note tuples [18] representation that uses a single vector for each note. Each vector has 25 dimensions: the 22 one-hot-encoded instrument, followed by normalized velocity, note duration and time-shift to the previous note. This results in shorter sequences, with as many elements as there are notes. Average sequence has 733 elements, $3.5\times$ less than tokenization methods.

Linearized Rhythmic Tree We use a simplified rhythm grammar of 15 rules on the GMD. As this grammar does not allow notes shorter than a $1/32$ nd note, the maximum depth d_{\max} of a leaf in the rhythmic tree is 6. Although slightly longer than note tuples, the resulting sequences remain on the smaller side with an average of 863 elements.

4.3 Models

We chose to focus on sequential representations and therefore consider two popular architectures: LSTM [30] and Transformers [26]. The model inputs are fed as chunks of 2, 4 or 8 measures. We perform a hyperparameter search for the number of bars, number of layers and layer width on the validation set and retain the best architectures for each (model, representation) combination. As our grammar parser uses the track’s tempo, we inject this information in non-grammatical models for a fair comparison by concatenating the tempo to the features vector in the last layer.

LSTM architecture We consider bidirectional LSTM models [31] and we experiment with a depth of 1 to 4 layers and a fixed width of 8 to 256 neurons per layer. Even though LSTMs do not require positional encoding, we also evaluate our LRT representation with TBPE to assess whether the explicit rhythmic structure is beneficial to the model.

Transformer architecture We use standard Transformers with an embedding layer, *i.e.* a linear projection, between the input and the first Transformer block. The models have 1 to 4 encoder layers, each with 2 to 16 attention heads. We also experiment with a feature size of 2 to 32 dimensions per head and 8 to 64 neurons in the feedforward network. We use the classical positional encoding for token sequences, the continuous positional encoding for piano rolls and note tuples, and either the classical or the tree-based positional encoding for LRTs. Regarding continuous encoding, we use $T_S = 100$ ms so that even close notes have a different encoding, and $T_L = 300$ s, as temporal context is unlikely to matter beyond several minutes.

Final models are trained with a batch size of 128, using the AdamW optimizer [32] with a learning rate of 0.001, decayed by a factor 10 every 50 epochs with weight decay and dropout. Early stopping occurs when the validation F1 score plateaus with a patience of 200 epochs. Models are trained using the standard cross-entropy loss. To alleviate the class imbalance (185 rock tracks versus 50 for the other classes), we use class inverse median frequency weighing. We report the macro F1 scores averaged over all classes.

4.4 Main Results

We report in Table 1 the test scores of the best combinations from the hyperparameter search, averaged over five runs.

LSTM with 30 Hz piano rolls and Transformer with LRT/TBPE are the combinations that lead to the best F1 scores overall (≈ 0.66). The former is a 3-layer LSTM model, each composed of 64 neurons, performing on 2-bar-long samples. The latter is a 4-layer Transformer model, each using 2 heads with 32 features per head (so a 64-dimensional input vector), and a feedforward network of 32 neurons trained on 4-bar-long chunks. Although both models achieve comparable performance, note that the Transformer model needs $6\times$ fewer parameters than the LSTM.

We observe that the TBPE provides important information for style classification. Transformer models using a

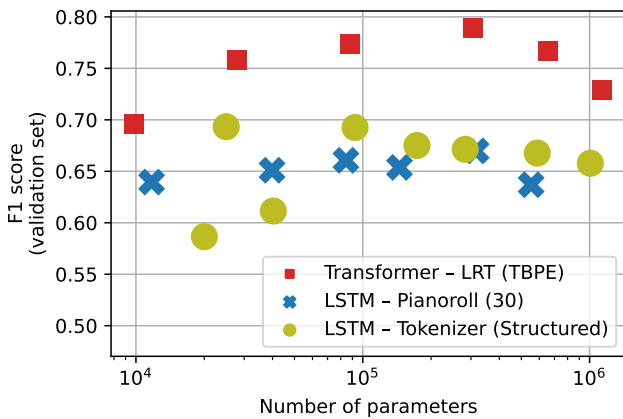


Figure 4: F1 scores on the validation set vs. number of parameters for a selected set of models. We observe that Transformers trained on LRT consistently outperform other models at similar capacity.

classical positional encoding achieve lower classification performance (≈ 0.56). Surprisingly, using TBPE is beneficial for LSTMs also: both our LSTM models trained on LRT achieve nearly identical F1 scores (≈ 0.6), however injecting the TBPE allows us to use a RNN with $5\times$ fewer parameters. This confirms that explicitly encoding the node position in the tree makes it easier for the models to understand the rhythmic structure of the track.

Finally, we observe that tokenization and note tuples tend to underperform overall. Structured MIDI tokenization achieves the best of tokenizer F1 score (≈ 0.6) both for LSTM and Transformer architecture, followed by MIDI-Like, however at the cost of a higher number of parameters. Token or note tuple sequences seem difficult to learn for the models. For RNN, we hypothesize that this is due to the regular sampling assumption made by these models. Each element is processed by the same recurrent loop, meaning that the model needs to learn the structure of the sequence, *e.g.* what each token represents. In comparison, piano rolls with a fixed time step where all elements represent the same object tend to have higher performances with LSTMs.

4.5 Model Parameter Efficiency

We evaluate some representative models by varying their capacity, *i.e.* number of parameters. More specifically, we experiment with 4, 8, 32 and 64 number of features per head for the Transformer, and 16, 32, 48, 64, 96 and 128 neurons in the hidden layers for LSTM. We report F1 scores on the validation set in Figure 4. We observe that, at comparable number of parameters, the Transformer trained on the LRT always lead to higher F1 scores than the compared models. This demonstrates that the rhythmic information embedded in our rhythmic tree not only results in shorter sequences, but also can be leveraged by smaller models for better or on par performance compared to existing works.

4.6 Training Samples Efficiency

Finally, we evaluate how representation affects the amount of data needed to train our models. We compare the same

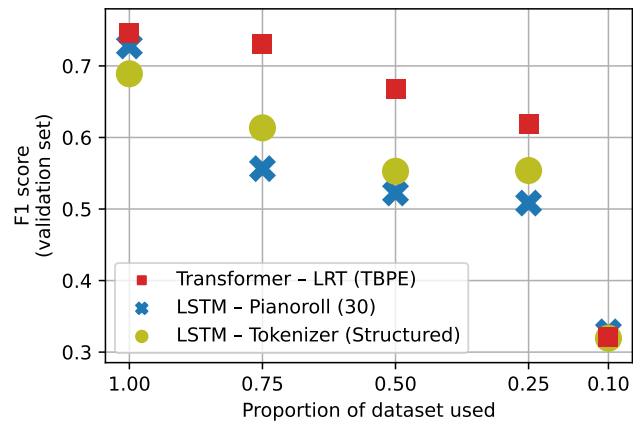


Figure 5: F1 scores on the validation set vs. percentage of training samples used. Transformers trained on LRT exhibit a less severe performance drop when the number of training samples decreases compared to existing models.

models as in Section 4.5 and train them with a random subset of 75%, 50%, 25% and 10% of the training set. F1 scores on the validation set are reported in Figure 5. We observe that the Transformer model trained with the linearized rhythmic tree and the tree-based positional encoding consistently outperforms the structured tokenizer and the piano roll. The performance drop between 100% and 75% is minimal, and overall the LRT-based Transformer degrades more gracefully when the number of training samples decreases compared to the other models. This underlines the relevance of the LRT, that encodes higher level musical information and better represents the invariance of musical style to spurious variations in the input MIDI file, such as slight changes in timings or velocity.

5. CONCLUSION AND FUTURE WORK

We evaluated different representations of MIDI data for drumming style classification. We introduced a new representation based on the linearization of a rhythmic tree obtained by parsing a MIDI file using a musical grammar. This representation provides richer features while being more compact than traditional piano rolls or tokenization strategies. Associated with a Transformer architecture using a tree-based positional encoding, we show that this representation achieves style classification performance on par with the best models from the literature with much fewer parameters. We also provide evidence that our representation is more resilient when trained on smaller datasets.

Future works involve extending this tree-based representation beyond homophonic input, *e.g.* for polyphonic piano pieces. Building the parsing tree could also be achieved on music scores, making it possible to directly classify scores at the mere symbolic level. In addition, we would like to evaluate this approach on more diverse tasks, as representation could be beneficial not only for discriminative models, but also for generative models, *e.g.* in music generation tasks, to produce syntactically correct performances with respect to the specified grammar [33].

6. ACKNOWLEDGEMENTS

We thank Florent Jacquemard for his work on `qpparse`, fruitful discussions on the design of the rhythmic grammar for drums and advice throughout this project. Additional thanks are dedicated to Lydia Rodriguez de la Nava for her help in adapting `qpparse` to drums rhythm parsing.

7. REFERENCES

- [1] Z. Cataltepe, Y. Yaslan, and A. Sonmez, "Music genre classification using MIDI and audio features," *EURASIP Journal on Advances in Signal Processing*, vol. 2007, pp. 1–8, 2007.
- [2] C. McKay and I. Fujinaga, "jSymbolic: A feature extractor for MIDI files." in *ICMC*, 2006.
- [3] M. S. Cuthbert, C. Ariza, and L. Friedland, "Feature extraction and machine learning on symbolic music using the music21 toolkit." ser. Proceedings of the 12th International Society for Music Information Retrieval Conference, 2011, pp. 387–392.
- [4] F. Simonetta, A. Llorens, M. Serrano, E. García-Portugués, and Á. Torrente, "Optimizing feature extraction for symbolic music," *Proceedings of the 24th International Society for Music Information Retrieval Conference*, 2023.
- [5] Q. Kong, K. Choi, and Y. Wang, "Large-scale midi-based composer classification," *arXiv preprint arXiv:2010.14805*, 2020.
- [6] J. Gillick, A. Roberts, J. Engel, D. Eck, and D. Baman, "Learning to groove with inverse sequence transformations," in *International Conference on Machine Learning (ICML)*, 2019.
- [7] N. Fradet, J.-P. Briot, F. Chhel, A. El Fallah Seghrouchni, and N. Gutowski, "MidiTok: A python package for MIDI file tokenization," in *Extended Abstracts for the Late-Breaking Demo Session of the 22nd International Society for Music Information Retrieval Conference*, 2021.
- [8] H. Zhang, E. Karystinaios, S. Dixon, G. Widmer, and C. E. Cancino-Chacón, "Symbolic Music Representations for Classification Tasks: A Systematic Evaluation," Milan, Italy, pp. 848–858, Nov. 2023.
- [9] G. Wiggins, E. Miranda, A. Smaill, and M. Harris, "A Framework for the Evaluation of Music Representation Systems," *Computer Music Journal*, vol. 17, Oct. 1993.
- [10] B. Wang and Y.-H. Yang, "PerformanceNet: Score-to-Audio Music Generation with Multi-Band Convolutional Residual Network," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 1174–1181, Jul. 2019.
- [11] F. Foscari, K. Hoedt, V. Praher, A. Flexer, and G. Widmer, "Concept-Based Techniques for "Musicologist-friendly" Explanations in a Deep Music Classifier." [object Object], 2022.
- [12] G. Velarde, T. Weyde, C. E. Cancino-Chacón, D. Meredith, and M. Grachten, "Composer Recognition Based on 2D-Filtered Piano-Rolls," in *International Society for Music Information Retrieval Conference*, Aug. 2016.
- [13] C.-Z. A. Huang, A. Vaswani, J. Uszkoreit, N. M. Shazeer, I. Simon, C. Hawthorne, A. M. Dai, M. Hoffman, M. Dinculescu, and D. Eck, "Music Transformer: Generating Music with Long-Term Structure," in *International Conference on Learning Representations*, Sep. 2018.
- [14] S. Oore, I. Simon, S. Dieleman, D. Eck, and K. Simonyan, "This time with feeling: Learning expressive musical performance," *Neural Computing and Applications*, vol. 32, no. 4, pp. 955–967, Feb. 2020.
- [15] G. Hadjeres and L. Crestel, "The Piano Inpainting Application," *ArXiv*, Jul. 2021.
- [16] N. Fradet, N. Gutowski, F. Chhel, and J.-P. Briot, "Byte pair encoding for symbolic music," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, H. Bouamor, J. Pino, and K. Bali, Eds. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 2001–2020.
- [17] Y.-S. Huang and Y.-H. Yang, "Pop Music Transformer: Beat-based Modeling and Generation of Expressive Pop Piano Compositions," *Proceedings of the 28th ACM International Conference on Multimedia*, pp. 1180–1188, Oct. 2020.
- [18] C. Hawthorne, A. Huang, D. Ippolito, and D. Eck, "Transformer-nade for piano performances," in *NIPS 2nd Workshop on Machine Learning for Creativity and Design*, 2018.
- [19] J. E. Hopcroft and J. D. Ullman, "Introduction to automata theory, languages and computation," 1979. [Online]. Available: <https://api.semanticscholar.org/CorpusID:31901407>
- [20] Y. Amagasu, F. Jacquemard, and M. Sakai, "Tokenization of MIDI Sequences for Transcription," in *9th International Conference on Technologies for Music Notation and Representation (TENOR 2024)*, Apr. 2024.
- [21] C. Agon, K. Haddad, and G. Assayag, "Representation and rendering of rhythm structures," in *Second International Conference on Web Delivering of Music, 2002. WEDELMUSIC 2002. Proceedings.*, Dec. 2002, pp. 109–113.
- [22] F. Jacquemard, P. Donat-Bouillud, and J. Bresson, "A Structural Theory of Rhythm Notation based on Tree Representations and Term Rewriting," in *Mathematics*

- and Computation in Music: 5th International Conference, MCM 2015*, vol. 9110. Springer, Jun. 2015, p. 12.
- [23] F. Foscarin, F. Jacquemard, P. Rigaux, and M. Sakai, “A Parse-based Framework for Coupled Rhythm Quantization and Score Structuring,” in *MCM 2019 - Mathematics and Computation in Music*, vol. Lecture Notes in Computer Science. Springer, Jun. 2019.
- [24] M. Digard, F. Jacquemard, and L. Rodriguez-de la Nava, “Automated Transcription of Electronic Drumkits,” in *4th International Workshop on Reading Music Systems (WoRMS)*, ser. Proceedings of the 4th International Workshop on Reading Music Systems, online, Spain, Nov. 2022.
- [25] F. Jacquemard and L. Rodriguez de La Nava, “Symbolic Weighted Language Models, Quantitative Parsing and Automated Music Transcription,” in *CIAA 2022 - International Conference on Implementation and Application of Automata*, ser. Lecture Notes in Computer Science, Vol 13266, P. Caron and L. Mignot, Eds. Rouen, France: Springer, Jun. 2022, pp. 67–79.
- [26] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. ukasz Kaiser, and I. Polosukhin, “Attention is All you Need,” in *Advances in Neural Information Processing Systems*, vol. 30. Curran Associates, Inc., 2017.
- [27] V. Shiv and C. Quirk, “Novel positional encodings to enable tree-based transformers,” in *Advances in Neural Information Processing Systems*, vol. 32. Curran Associates, Inc., 2019.
- [28] H. Peng, G. Li, Y. Zhao, and Z. Jin, “Rethinking Positional Encoding in Tree Transformer for Code Representation,” in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Y. Goldberg, Z. Kozareva, and Y. Zhang, Eds. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 3204–3214.
- [29] W. Goebel, “Melody lead in piano performance: Expressive device or artifact?” *The Journal of the Acoustical Society of America*, vol. 110, pp. 563–72, Aug. 2001.
- [30] S. Hochreiter and J. Schmidhuber, “Long Short-term Memory,” *Neural computation*, vol. 9, pp. 1735–80, Dec. 1997.
- [31] A. Graves and J. Schmidhuber, “Framewise phoneme classification with bidirectional LSTM networks,” in *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, vol. 4, Jul. 2005, pp. 2047–2052 vol. 4.
- [32] I. Loshchilov and F. Hutter, “Decoupled Weight Decay Regularization,” in *International Conference on Learning Representations*, Sep. 2018.
- [33] M. J. Kusner, B. Paige, and J. M. Hernández-Lobato, “Grammar Variational Autoencoder,” in *Proceedings of the 34th International Conference on Machine Learning*. PMLR, Jul. 2017, pp. 1945–1954.

NESTED MUSIC TRANSFORMER: SEQUENTIALLY DECODING COMPOUND TOKENS IN SYMBOLIC MUSIC AND AUDIO GENERATION

Jiwoo Ryu¹

Hao-Wen Dong²

Jongmin Jung¹

Dasaem Jeong³

¹Dept. of Artificial Intelligence, ³Dept. of Art & Technology, Sogang University, Seoul, South Korea

²University of California San Diego, US

{judejiwoo, jongmin, dasaemj}@sogang.ac.kr, hwdong@ucsd.edu

ABSTRACT

Representing symbolic music with compound tokens, where each token consists of several different sub-tokens representing a distinct musical feature or attribute, offers the advantage of reducing sequence length. While previous research has validated the efficacy of compound tokens in music sequence modeling, predicting all sub-tokens simultaneously can lead to suboptimal results as it may not fully capture the interdependencies between them. We introduce the Nested Music Transformer (NMT), an architecture tailored for decoding compound tokens autoregressively, similar to processing flattened tokens, but with low memory usage. The NMT consists of two transformers: the main decoder that models a sequence of compound tokens and the sub-decoder for modeling sub-tokens of each compound token. The experiment results showed that applying the NMT to compound tokens can enhance the performance in terms of better perplexity in processing various symbolic music datasets and discrete audio tokens from the MAESTRO dataset.

1. INTRODUCTION

The effectiveness of the autoregressive language model becomes dominant in generative tasks in various domains, including music. The language model has been the most widely used generative model in symbolic music generation [1–4]. After the success of vector quantization or residual vector quantization [5], the language model is also widely applied to audio-domain music generation [6–8].

The power of the language model comes from its autoregressive modeling of sequential information. Once the data is *flattened* to a sequence of discrete tokens, the language model can be applied in a straightforward manner. There have been many successive works on representing symbolic music data in a sequence of flattened tokens, such as MIDI-like encoding [9] or REMI [3].

However, a limitation of this approach is that the sequence length is quite lengthy, with the average number

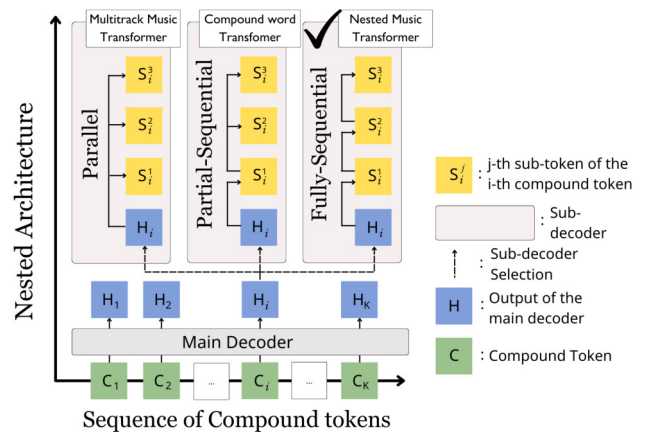


Figure 1: Diagram of the nested architecture with three different methods for predicting sub-tokens.

of tokens for pieces within the Lakh MIDI dataset [10] reaching 14,647. To overcome this limitation, the Compound Word Transformer [4] proposed an encoding scheme named Compound word that represents symbolic music as a sequence of compound tokens, in which several musical features or attributes are encoded into a single multi-dimensional token. By grouping musical features into two different compound token types, metric and note, Compound word shortens the sequence length to less than half of what is encoded with REMI as depicted in Figure 2. Similarly, Multitrack Music Transformer [11] employed a compound token scheme that encodes beat position, instrument, pitch, and duration into a single token, resulting in a sequence length approximately one-third of that encoded with REMI. Furthermore, note-level compound tokens demonstrated a clear advantage in performance for discriminative tasks such as identifying the genre or style of music and suggesting accompaniments [12].

Despite these attempts to reduce the sequence length by packing musical features into a single compound token for various purposes, encoding schemes which flatten tokens like REMI are still dominant in symbolic music generation. Both [4] and [11] in symbolic music generation showed that the generation with REMI was favored in their listening tests. One of the causes is that the previous models are designed to predict multiple features in a parallel [11] or partial-sequential [4] way without considering interdependencies between different musical features encoded within

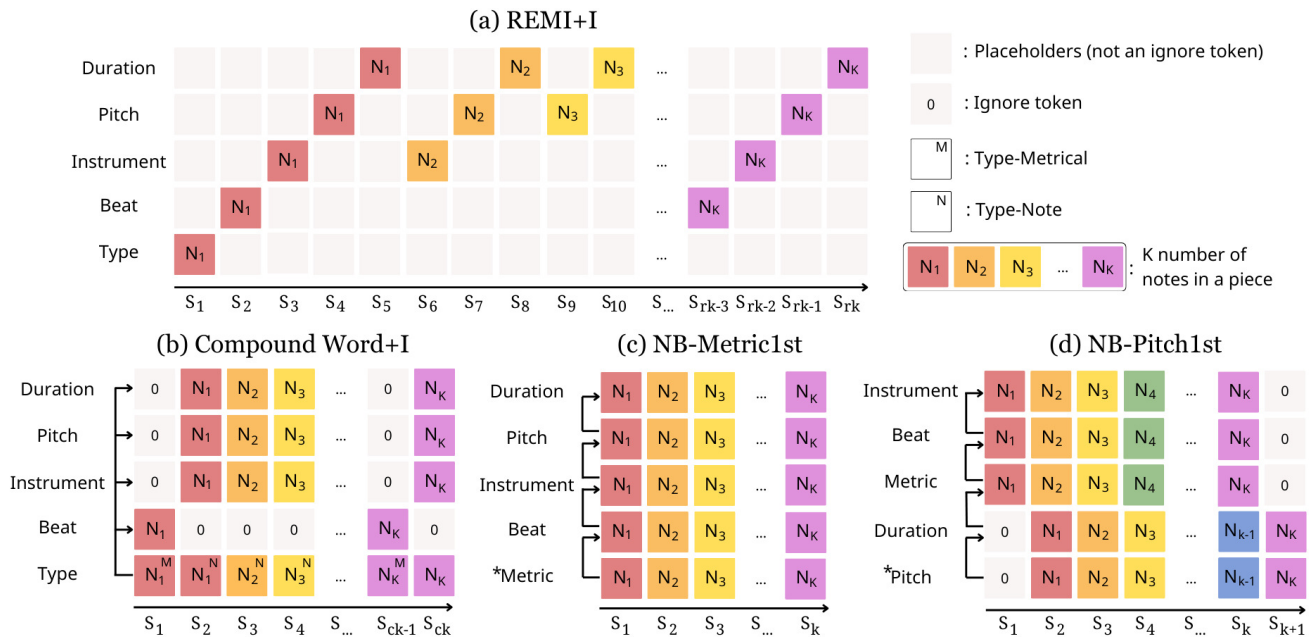


Figure 2: An example illustrating the proposed representations, note-based (NB) encoding (c) NB-Metric1st and (d) NB-Pitch1st, alongside REMI and Compound word. All encodings represent the same piece of music by using five musical features. Specifically, REMI and Compound word were not originally designed for multi-instrument pieces, which is why we renamed the encodings with “+I” to (a) and (b). Here, k denotes the number of notes and sequence length for NB, while r and c represent the ratios for REMI and Compound word, with values greater than 1.

the compound token, as depicted in Figure 1.

To address this challenge, we introduce a novel decoding framework called the Nested Music Transformer (NMT). The primary goal of this framework is to decode compound tokens in a fully sequential manner while maintaining efficient memory usage. The proposed NMT combines two distinct cross-attention architectures within its sub-decoder: the intra-token decoder and the Embedding Enricher. The intra-token decoder autoregressively decodes the sub-tokens of a single compound token, while the Embedding Enricher updates embedding of each sub-token by attending to the hidden states of previous compound tokens.

We demonstrated that our proposed architecture achieves performance comparable to that of flattening-based models, while requiring fewer computational resources in terms of GPU memory and training time. This was confirmed through both quantitative evaluations and subjective listening tests for symbolic music generation. Furthermore, our experiments showed that the NMT and other nested architectures perform similarly to strong baseline models when generating audio samples using discrete audio tokens. All source code, pretrained models and generated samples are available at <https://github.com/JudeJiwoo/nmt>.

2. NOTE-BASED ENCODING

Before we introduce the Nested Music Transformer, we explain Note-based encoding (NB), a compound token encoding scheme that we utilized as the primary encoding method. NB stands out for its ability to encapsulate the most comprehensive set of musical features within a single compound token, as illustrated in Figure 2.

2.1 Musical Features in Symbolic Encoding

As depicted in Figure 2, REMI, Compound word, and NB utilize several musical features to represent music pieces. We used a total of eight features: beat (position), pitch, and duration were essential, while instrument, chord, tempo, and velocity were selectively included based on the dataset characteristics. To encode other information, such as measure boundary and change in time signature, we also employed one additional feature *Type* or *Metric* following [4].

In Compound word (CP), musical features are categorized into two groups: “metrical” and “note.” Consequently, the encoding employs two *Type* tokens to specify the group of each compound token. Unlike CP, NB does not require group indicator tokens however, since each note token in NB is assigned a beat, unlike REMI and CP, we designed the *Metric* feature to encode changes in the metrical structure. This allows the model to efficiently represent metrical changes within a single sub-token. Specifically, the *Metric* feature indicates whether the current note introduces a new time signature, measure, or beat, or continues the previous metrical context. For this purpose, we define four distinct values for the *Metric* feature vocabulary, each representing a different combination of metrical changes or continuations.

The *Beat* indicates the relative position of each note within a measure. The *Chord* was derived using a rule-based algorithm from [4]. The *Tempo* was set to follow an exponential scale for value changes, with this application varying across datasets. The *Instrument* feature specifies the instrument playing the note. In order to keep the variety of instruments manageable, we adopted the approach suggested in [11], trimming to 61 types of instruments. The

Pitch feature utilized 128 categories of pitch values represented in MIDI. The *Duration* refers to the length of time each note is played. The *Velocity* represents MIDI velocity (dynamics) of each note.

For the NB encoding method, a music piece P with K number of notes, $P = \{n_1, n_2, n_3, \dots, n_K\}$ can be conceptualized as a sequence of compound tokens, denoted by $P_{nb} = \{x_1, x_2, x_3, \dots, x_K\}$, wherein each event x_i is a compound token comprising up to eight sub-tokens in the orders like followings:

$$(x_i^{\text{metric}}, x_i^{\text{beat}}, x_i^{\text{chord}}, x_i^{\text{tempo}}, x_i^{\text{inst}}, x_i^{\text{pitch}}, x_i^{\text{dur}}, x_i^{\text{vel}})$$

2.2 Compound Shift

By reordering the sub-tokens within a compound token, we can position the target sub-token to be predicted first. This adjustment enhances the objective metric of the target sub-token, as it benefits from being processed primarily by the more powerful main decoder rather than the sub-decoder. Each event x_i which is shifted to pitch-first option comprises features like following:

$$(x_{i-1}^{\text{pitch}}, x_{i-1}^{\text{dur}}, x_{i-1}^{\text{vel}}, x_i^{\text{metric}}, x_i^{\text{beat}}, x_i^{\text{chord}}, x_i^{\text{tempo}}, x_i^{\text{inst}})$$

Note that the order of prediction of each sub-token in the entire *flattened* sequence does not change, and only the grouping boundary for a single compound token is shifted as depicted in Figure 2 (d). We will refer to the non-shifted representation as *NB-MF* and the shifted version as *NB-PF*.

3. NESTED MUSIC TRANSFORMER

In this section, we introduce the architecture of Nested Music Transformer (NMT), which is designed to handle compound tokens. The structure is composed of three primary components: token embedding, main decoder, and sub-decoder. The token embedding component summarizes the embeddings of each sub-token into a single vector which represents each compound token. Subsequently, the main decoder processes the sequence of these vectors using a decoder-only transformer architecture. Lastly, the sub-decoder decodes sub-tokens from the output of the main decoder. The proposed NMT integrates two distinct cross-attention architectures within its sub-decoder: the intra-token decoder and the Embedding Enricher. As the NMT generates sub-tokens, their embeddings are updated with contextual information by the Embedding Enricher, as illustrated in Figure 3.

3.1 Token Embedding & Main Decoder

To summarize multiple embeddings from each sub-token, we simply sum them along the sub-token axis following [6, 11]. Additionally, we integrate learnable absolute positional embedding [13] to denote the position of compound tokens within the sequence. Specifically, the i -th compound token x_i in the sequence is converted into a vector through the token embedding process and aggregated with its positional embedding. This combined vector is then fed into the main decoder, producing the output of the main decoder, also known as the hidden vector h_i .

3.2 Sub-decoder with Cross Attention

The main goal of the sub-decoder is to obtain proper hidden state to predict output sub-token s_i^j which is j -th sub-token of i -th compound token, based on output of the main decoder h_i and the preceding output sub-tokens s_i^0, \dots, s_i^{j-1} that are predicted before.

Many previous works have suggested using a similar sub-decoder to sequentially predict the sub-token sequence, such as updating hidden state by concatenating with the embedding of sub-tokens [4], using RNN [14] or causal self-attention [8]. However, through comparative experiments presented in Section 4, we found that applying cross-attention is one of the most effective way to model the compound token sequence in symbolic music.

The cross-attention-based sub-decoder operates by iteratively concatenating a key/value pair sequence K/V_i with embeddings of sub-tokens $\text{Emb}(s_i)$, starting with an initial key/value sequence that contains only the beginning-of-sequence *BOS* token. For each sub-token to be sampled, the architecture computes multi-head scaled dot-product cross-attention between the query sequence, consisting of positionally encoded output of the main decoder h_i , and the current key/value sequence. The positional encoding of h_i ensures that the hidden vector has a distinct bias for predicting target sub-token. From the attention output a_i^j , the matrix W_{logits}^j is applied to create logits. This iterative process continues until all sub-tokens are sampled. The process can be expressed as follows:

$$\text{Query}_i^j = \text{PositionalEncoding}(h_i), \quad (1)$$

$$K/V_i^j = \begin{cases} \text{BOS} & \text{if } j = 0, \\ \text{Concat}(\text{BOS}, \dots, \text{Emb}(s_i^{j-1})) & \text{if } j > 0, \end{cases} \quad (2)$$

$$a_i^j = \text{Cross-Attention}(\text{Query}_i^j, K/V_i^j), \quad (3)$$

$$s_i^j = \text{Sampling}(\text{Softmax}(a_i^j W_{\text{logits}}^j)) \quad (4)$$

3.2.1 Embedding Enricher

Since the embedding of a sub-token is a shallower vector compared to the output of the main decoder, we designed a cross-attention architecture called the Embedding Enricher. This architecture updates embedding of sub-token $\text{Emb}(s_i)$ with a context sequence derived from the prior outputs of the main decoder $h_{i-(w-1)}, \dots, h_i$, where w represents the window size.

$$\text{Context}_i = \text{Concat}(\text{BOS}, h_{i-(w-1)}, \dots, h_i), \quad (5)$$

$$\text{Enriched}_i = \text{Cross-Attention}(\text{Emb}(s_i), \text{Context}_i) \quad (6)$$

In the Nested Music Transformer, the output vector Enriched_i replaces the original embedding of sub-tokens before being concatenated into the key/value pair sequence in Equation (2) as depicted in Figure 3. These context-enriched embeddings allow the architecture to process attention with deeper vectors than the original embeddings, resulting in better performance on the objective metric compared to the standalone cross-attention-based sub-decoder, as demonstrated in Table 1.

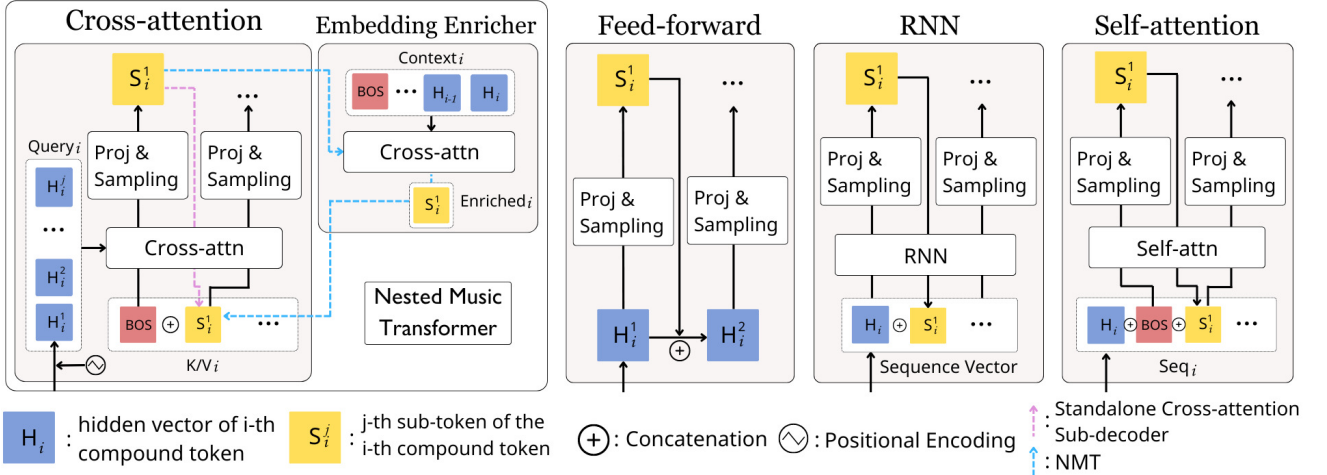


Figure 3: Illustrations of the proposed Nested Music Transformer (NMT) and other sub-decoder structures

3.3 Other Comparative Structures

3.3.1 Feed-forward-based Architecture

The Feed-forward-based sub-decoder, inspired by [4], iteratively updates the output of the main decoder to predict sub-tokens. It concatenates the previously used hidden state with the embedding of the last sampled output to predict the next sub-token.

3.3.2 RNN-based Architecture

RNN-based sub-decoder capitalizes on the sequential nature of recurrent neural network to update hidden state. The initial input sequence and hidden state utilize the output of the main decoder h_i , and through the iteration the embedding of the sampled output is appended to the input sequence until all the sub-tokens are generated.

3.3.3 Self-attention-based Architecture¹

The self-attention-based sub-decoder aims to get the sequence vector Seq_i by iteratively concatenating it with the embeddings of the sampled output $\text{Emb}(s_i)$. The initial sequence vector consists of the output of the main decoder h_i and BOS token to ensure that the initial attention values can be properly processed. This sequence vector Seq_i is then used as the query, key, and value in the self-attention mechanism. The process can be summarized as follows:

$$\text{Seq}_i^j = \begin{cases} \text{Concat}(h_i, \text{BOS}) & \text{if } j = 0, \\ \text{Concat}(h_i, \text{BOS}, \dots, \text{Emb}(s_i^{j-1})) & \text{if } j > 0, \end{cases} \quad (7)$$

$$a_i^j = \text{Self-Attention}(\text{Seq}_i^j), \quad (8)$$

$$s_i^j = \text{Sampling}(\text{Softmax}(a_i^j W_{\text{logits}}^j)) \quad (9)$$

¹ The proposed self-attention-based sub-decoder operates differently from the method described in [8]. Unlike ours, [8] used h_i as a base of every vector in the sequence, which is updated by the embedding of generated sub-tokens, similar to the operation of our proposed cross-attention-based sub-decoder. Experimental results indicate that the architecture in [8] outperforms our self-attention-based architecture and delivers comparable results to our cross-attention-based architecture.

3.4 Self-attention versus Cross-attention

The preference for cross-attention over self-attention arises from the observation that the output of the main decoder, h_i , already contains sufficient information to predict sub-tokens, as demonstrated in the parallel prediction method used in the Multitrack Music Transformer [11]. On the other hand, the embedding of the sampled output is comparatively shallow, lacking the previous context despite having the same dimension as h_i . Additionally, since both attention layers use a residual connection for the vectors used as keys, utilizing h_i as the key facilitates a direct gradient flow. Therefore, updating h_i as the key with cross-attention can be more advantageous than updating the embedding of the sampled sub-token with self-attention.

3.5 Applying to Audio Tokens

MusicGen [6] has employed a four-level residual vector quantization technique for a single token, which bears similarity to using four musical features or sub-tokens for compound tokens in symbolic music. Given that the optimal architecture, particularly for decoding compound features in a fully-sequential manner, is still being explored for audio tokens [8], we employed the Nested Music Transformer on discrete audio tokens to assess the potential of our proposed architecture.

4. EXPERIMENTS

4.1 Dataset Preparation

We selected four datasets to conduct our experiments on symbolic music generation: Pop1k7 [4], Pop909 [15], the Symbolic Orchestral Database (SOD) [16], and the clean version of the Lakh MIDI Dataset (LMD clean) [10], which is free of data leakage problems. During preprocessing, MIDI files without a time signature or with excessive or insufficient length were filtered out, and we specifically selected pieces featuring a minimum of four instruments for LMD clean. Note quantization varied across datasets: twelve resolutions per beat for SOD and four resolutions

	SOD						Lakh			Pop1k7			Pop909		
	GPU mem.(GB)	Time(s) / iter.	Token Len.	Mean↓	Beat	Pitch	Mean	Beat	Pitch	Mean	Beat	Pitch	Mean	Beat	Pitch
REMI [3]	19.90	0.461	6,638(±7,518)	0.474	0.229	0.753	0.294	0.293	0.408	1.087	0.470	1.138	0.716	0.368	<u>0.984</u>
CP [4]	7.93	0.119	3,230(±3,480)	0.604	0.257	0.971	0.361	0.288	0.527	1.172	0.495	1.219	0.911	0.410	1.220
CP* + NMT	16.13	0.224	–	<u>0.545</u>	<u>0.237</u>	0.864	0.327	0.288	0.466	<u>1.103</u>	<u>0.483</u>	<u>1.154</u>	<u>0.724</u>	<u>0.334</u>	0.969
NB-MF + Par. [11]	8.40	0.123	2,398(±2,764)	0.712	0.466	1.084	0.431	0.431	0.604	1.480	0.871	1.802	1.003	0.674	1.393
NB-MF + NMT	16.14	0.215	–	0.567	0.246	0.906	0.324	0.276	0.466	1.168	0.503	1.304	0.803	0.264	1.114
NB-PF + Par.	8.30	0.120	–	0.632	0.565	0.913	0.376	0.502	0.481	1.396	0.998	1.604	0.986	0.824	1.359
NB-PF + CA	14.74	0.174	–	0.564	0.276	0.867	<u>0.305</u>	0.287	<u>0.424</u>	1.161	0.538	1.244	0.767	0.357	1.052
NB-PF + NMT	16.13	0.217	–	0.549	0.263	<u>0.855</u>	0.306	<u>0.285</u>	0.427	1.149	0.515	1.243	0.771	0.345	1.090
NB-PF + FF	8.12	0.122	–	0.607	0.361	0.881	0.338	0.372	0.449	1.280	0.635	1.396	0.850	0.431	1.121
NB-PF + RNN	9.77	0.144	–	0.591	0.300	0.915	0.315	0.297	0.437	1.166	0.531	1.257	0.792	0.366	1.077
NB-PF + SA	15.67	0.181	–	0.574	0.287	0.902	0.311	0.287	0.431	1.204	0.553	1.320	0.849	0.417	1.150

CP*: Compound word representation NB-MF: metric-first NB NB-PF: pitch-first NB NMT: cross-attention-based sub-decoder + Embedding Enricher CA: cross-attention-based sub-decoder FF: Feed-forward-based sub-decoder SA: self-attention-based sub-decoder

Table 1: Model comparison on their average NLL loss for symbolic music. The GPU memory usage and iteration times for each model in SOD is included. Additionally, we included the average token length and standard deviation across all pieces in SOD.

per beat for the others. We also filtered out MIDI files with expressive tempo and timing. We split the prepared data, reserving 10% for validation and 10% for testing. Additionally, augmentation techniques for pitch and chord involved random semitone shifts $s \in \mathbb{Z}$ within a range of $s \sim U(-5, 6)$.

4.2 EnCodec for MAESTRO

For discrete audio tokens, we prepared MAESTRO dataset [9], which has 200 hours piano performance audio files. We fine-tuned the audio tokenizer proposed by [6] with MAESTRO audio files to create sequences of discrete audio tokens, each with 30 seconds of length. The sampling rate of the token is 50 Hz, which means 30 seconds of audio is represented with 1500 audio tokens, each with 4 different codebooks.

4.3 Model and Hyperparameter Configuration

The baseline models for symbolic music generation are defined as follows: flattening for REMI [3], partial-sequential Feed-forward-based sub-decoder for Compound word [4], and parallel prediction with NB-MF [11]. Additionally, the *delay* method proposed by [6] is explored as a baseline for generating audio tokens, which utilizes rearranged residual vectors or sub-tokens in a parallel manner. In exploring both symbolic music and audio token generation, we conducted experiments using the Nested Music Transformer (NMT) and various sub-decoder architectures to assess the effectiveness of our proposed model. To ensure a fair comparison among these models, we aimed for a comparable number of model parameters, approximately 40 million for symbolic music and 62 million for discrete audio tokens². To enhance efficiency in processing long sequences within the transformer architecture, we integrated Flash attention [17].

Training the model entailed 100K steps for symbolic music and 200k for discrete audio tokens, utilizing the

² Both models have 8 attention heads and a dimension size of 512, with a single layer for all sub-decoder architectures and an additional single layer for the Embedding Enricher when using the NMT. However, they have a total of 12 and 15 decoder layers, respectively.

AdamW optimizer [18], with a segment batch size of 8 and 16 for each task, where β_1 were set to 0.9, β_2 to 0.95, and a gradient clipping threshold was set to 1.0. We implemented a cosine learning rate schedule with a warm-up phase of 2000 or 4000 steps for each task. During this warm-up phase, the learning rate gradually increased before reaching its maximum value $1 \times e^{-4}$. To address overfitting concerns, we applied dataset-specific dropout rates instead of using early stopping. These dropout rates were chosen to ensure that the optimal validation loss remained stable until the end of training. We utilized mixed precision techniques.

4.4 Quantitative Evaluation on Symbolic Music

We evaluated the symbolic music generation task using the average negative log-likelihood (NLL). However, directly comparing the loss values across models using different encoding schemes posed challenges. To address this, we first adjusted the input sequence length for each encoding scheme to ensure that the NLL is derived from a similar amount of context regardless of the encoding scheme. Furthermore, instead of calculating the average NLL as done during the training steps, we calculated it based on the set of probabilities of tokens processed with full context. To achieve this, we used a moving-window method with a window size equal to the input sequence length to create a set of overlapping input sequences.

Secondly, we adjusted the probabilities for each sub-token in a compound token to account for discrepancies between REMI and other encoding schemes like CP and NB. REMI omits redundant tokens such as repetitive positions (beat), as depicted in Figure 2. Thus, when predicting a new note, a model based on REMI must decide whether to add the note at a new position by predicting a new beat token, or to add the note at the same position by predicting a pitch token. In contrast, CP and NB, due to the nature of their encoding schemes, split this prediction into two steps: first, they determine the beat position, and then they predict the pitch. This means they have more prior information when predicting the pitch token since changes in beat are already fixed and provided as a condition. To ad-

	FAD-uncon↓	FAD-cond↓	KLD↓	mean NLL↓
Parallel	0.166	0.206	0.075	4.669
Flatten	0.140	0.176	0.068	4.482
Delay [6]	0.168	0.188	0.066	4.564
Self-attention	0.131	0.186	0.074	4.353
Cross-attention	0.145	0.190	0.065	4.314
NMT	0.165	0.198	0.067	4.318

Table 2: Model comparison for discrete audio tokens

just the probability of sub-tokens in NB and CP, which differ due to the discrepancy, we accumulated the probability of each sub-token to the next token in NB or CP if that sub-token was omitted in its corresponding REMI encoding. For example, when predicting a pitch token at the same beat, $P(\text{pitch} \mid \text{context})$ in REMI can be compared to $P(\text{same_beat} \mid \text{context}) \times P(\text{pitch} \mid \text{context, same_beat})$ in NB or CP.

From the results, we observe several key tendencies. First, applying the Nested Music Transformer (NMT) enhances the overall performance across all types of compound token encodings, including previously suggested schemes like CP and NB-MF (similar to [11]). Second, the NMT demonstrates a clear advantage in using the cross-attention-based sub-decoder and the Embedding Enricher compared to other baseline architectures. Finally, our pitch-first NB (NB-PF) encoding outperforms the metric-first NB (NB-MF) encoding in predicting pitch. This is because the model can predict the next pitch feature through the main decoder by leveraging the previously inferred note position information. Conversely, NB-MF showed lower loss in beat prediction. This difference arises from which sub-token relationships are calculated through the main decoder instead of the sub-decoder. Overall, the results indicate that pitch-first token grouping is an efficient strategy.

4.5 Quantitative Evaluation on Discrete Audio Tokens

We evaluated models with discrete audio tokens using following metrics: Fréchet Audio Distance (FAD), Kullback-Leibler Divergence (KL), and the mean NLL loss over sequences. A lower FAD score suggests that the generated audio is more plausible. To mitigate sample number bias for the test set, we employed adaptive FAD as proposed by [19], along with CLAP [20] embeddings for each sample. FAD scores were computed based on 500 unconditionally generated samples and 345 samples generated given prompts. Following [6], we computed the KL-divergence over the probabilities of the labels between the original and the generated audio samples. Table 2 shows the evaluation results.

We observe that using a cross-attention-based sub-decoder or the NMT achieves better NLL compared to a self-attention-based sub-decoder. However, the tendency differs from that seen in symbolic music. Adding the Embedding Enricher did not significantly improve performance in the audio domain. We hypothesize that this disparity arises from the distinct characteristics of tokens in both domains. In the symbolic domain, each musical feature requires context to form sufficient semantic information, whereas each token in the audio domain, with a 2048 vocab-

	Coherence↑	Richness↑	Consistency↑	Overall↑
	Mean(±margin of error)			
REMI [3]	3.18 ± 0.20	3.33 ± 0.18	3.33 ± 0.18	3.17 ± 0.18
CP [4]	2.94 ± 0.22	3.24 ± 0.18	2.97 ± 0.20	3.06 ± 0.20
CP + NMT	3.22 ± 0.19	3.35 ± 0.17	3.39 ± 0.17	3.32 ± 0.17
NB-PF + NMT	3.37 ± 0.19	3.44 ± 0.18	3.37 ± 0.19	3.36 ± 0.20

Table 3: Results of subjective listening test, presenting mean values with 95% confidence intervals.

ulary size codebook, contains more standalone information. This observation suggests potential avenues for future research, such as exploring effective methods to integrate the semantic information of symbolic music with discrete audio tokens.

4.6 Subjective Listening Test

For the subjective listening test, we used the Symbolic Orchestral Database (SOD) [16] to generate MIDI samples given four-measure prompts. We carefully selected eight prompts from the test split and generated continuation results using four different models: two baseline models (REMI and CP) and two proposed models (CP + NMT and NB-PF + NMT). We applied different sampling methods to each model.³ We conducted the test with 29 participants, asking them to evaluate the generated outputs based on three criteria: *Coherence* (the naturalness of transitions), *Richness* (the variety of harmony and rhythm), and *Consistency* (the lack of errors in composition), as well as an *Overall* rating for the perceptual quality of the samples as a whole.

As summarized in Table 3, our proposed models generated samples of comparable quality to REMI, outperforming the baseline CP. The smaller gap between REMI and NB + NMT in the subjective listening test compared to the teacher-forcing NLL evaluation suggests that NB + NMT may be more robust to exposure bias during sequence generation. Another possible explanation is that compound tokens are more effective at capturing the given context, as also demonstrated in the experiments of [4].

5. CONCLUSION

In summary, this work presents the Nested Music Transformer, an advanced architecture that decodes compound tokens in music generation, applicable to both in the symbolic and audio domain. Our architecture distinguishes itself by addressing the twin challenges of sequence length and feature interdependencies through a nested transformer setup that efficiently manages GPU resources and training processes. The experiments validate the competitiveness of our model over previous methods, achieving on par results in both objective metrics and subjective listening tests while lowering training costs.

³ During the generation process, we used nucleus sampling (top-p sampling) with $p = 0.99$. Our proposed models were sensitive to the choice of the temperature parameter, where an improperly selected temperature would result in excessive repetition regardless of encoding schemes. Therefore, we searched for the optimal temperature value for each model within the range of [1.0, 1.3] on the validation set.

6. ACKNOWLEDGEMENTS

This research was supported by the National R&D Program through the National Research Foundation of Korea (NRF) funded by the Korean Government (MSIT) (RS-2023-00252944, Korean Traditional Gagok Generation Using Deep Learning).

7. REFERENCES

- [1] C.-Z. A. Huang, A. Vaswani, J. Uszkoreit, I. Simon, C. Hawthorne, N. Shazeer, A. M. Dai, M. D. Hoffman, M. Dinculescu, and D. Eck, “Music transformer: Generating music with long-term structure,” in *International Conference on Learning Representations*, 2018.
- [2] Y. Ren, J. He, X. Tan, T. Qin, Z. Zhao, and T.-Y. Liu, “Popmag: Pop music accompaniment generation,” in *Proceedings of the 28th ACM international conference on multimedia*, 2020, pp. 1198–1206.
- [3] Y.-S. Huang and Y.-H. Yang, “Pop music transformer: Beat-based modeling and generation of expressive pop piano compositions,” in *Proceedings of the 28th ACM international conference on multimedia*, 2020, pp. 1180–1188.
- [4] W.-Y. Hsiao, J.-Y. Liu, Y.-C. Yeh, and Y.-H. Yang, “Compound word transformer: Learning to compose full-song music over dynamic directed hypergraphs,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 1, 2021, pp. 178–186.
- [5] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, “High fidelity neural audio compression,” *Transactions on Machine Learning Research*, 2023.
- [6] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Defossez, “Simple and controllable music generation,” in *Advances in Neural Information Processing Systems*, A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., vol. 36. Curran Associates, Inc., 2023, pp. 47 704–47 720.
- [7] G. Le Lan, V. Nagaraja, E. Chang, D. Kant, Z. Ni, Y. Shi, F. Iandola, and V. Chandra, “Stack-and-delay: a new codebook pattern for music generation,” in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 796–800.
- [8] D. Yang, J. Tian, X. Tan, R. Huang, S. Liu, H. Guo, X. Chang, J. Shi, J. Bian, Z. Zhao *et al.*, “Uniaudio: Towards universal audio generation with large language models,” in *Forty-first International Conference on Machine Learning*, 2024.
- [9] C. Hawthorne, A. Stasyuk, A. Roberts, I. Simon, C.-Z. A. Huang, S. Dieleman, E. Elsen, J. Engel, and D. Eck, “Enabling factorized piano music modeling and generation with the maestro dataset,” *arXiv preprint arXiv:1810.12247*, 2018.
- [10] C. Raffel, *Learning-Based Methods for Comparing Sequences, with Applications to Audio-to-MIDI Alignment and Matching*. Columbia University, 2016.
- [11] H.-W. Dong, K. Chen, S. Dubnov, J. McAuley, and T. Berg-Kirkpatrick, “Multitrack music transformer,” in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.
- [12] M. Zeng, X. Tan, R. Wang, Z. Ju, T. Qin, and T. Liu, “Musicbert: Symbolic music understanding with large-scale pre-training,” in *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, ser. Findings of ACL, vol. ACL/IJCNLP 2021. Association for Computational Linguistics, 2021, pp. 791–800.
- [13] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, “Convolutional sequence to sequence learning,” in *International conference on machine learning*. PMLR, 2017, pp. 1243–1252.
- [14] Z. Wang, Y. Zhang, Y. Zhang, J. Jiang, R. Yang, G. Xia, and J. Zhao, “PIANOTREE VAE: structured representation learning for polyphonic music,” in *Proceedings of the 21th International Society for Music Information Retrieval Conference, ISMIR 2020, Montreal, Canada, October 11-16, 2020*, J. Cumming, J. H. Lee, B. McFee, M. Schedl, J. Devaney, C. McKay, E. Zangerle, and T. de Reuse, Eds., 2020, pp. 368–375. [Online]. Available: <http://archives.ismir.net/ismir2020/paper/000096.pdf>
- [15] Z. Wang, K. Chen, J. Jiang, Y. Zhang, M. Xu, S. Dai, and G. Xia, “POP909: A pop-song dataset for music arrangement generation,” in *Proceedings of the 21th International Society for Music Information Retrieval Conference, ISMIR 2020, Montreal, Canada, October 11-16, 2020*, 2020, pp. 38–45.
- [16] L. Crestel, P. Esling, L. Heng, and S. McAdams, “A database linking piano and orchestral MIDI scores with application to automatic projective orchestration,” in *Proceedings of the 18th International Society for Music Information Retrieval Conference, ISMIR 2017, Suzhou, China, October 23-27, 2017*, 2017, pp. 592–598.
- [17] T. Dao, D. Fu, S. Ermon, A. Rudra, and C. Ré, “Flashattention: Fast and memory-efficient exact attention with io-awareness,” *Advances in Neural Information Processing Systems*, vol. 35, pp. 16 344–16 359, 2022.
- [18] I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” in *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.
- [19] A. Gui, H. Gamper, S. Braun, and D. Emmanouilidou, “Adapting frechet audio distance for generative music evaluation,” in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 1331–1335.

- [20] Y. Wu, K. Chen, T. Zhang, Y. Hui, T. Berg-Kirkpatrick, and S. Dubnov, "Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

CONTINUAL LEARNING FOR MUSIC CLASSIFICATION

Pedro González-Barrachina^{1,2}

María Alfaro-Contreras¹

Jorge Calvo-Zaragoza¹

¹ Pattern Recognition and Artificial Intelligence Group, University of Alicante, Spain

² Alice Biometrics

pcg71@alu.ua.es, {malfaro, jcalvo}@dlsi.ua.es

ABSTRACT

Music classification is a prominent research area within Music Information Retrieval. While Deep Learning methods can adequately perform this task, their classification space remains fixed once trained, which conflicts with the dynamic nature of the ever-evolving music landscape. This work explores, for the first time, the application of Continual Learning (CL) in the context of music classification. Specifically, we thoroughly evaluate five state-of-the-art CL approaches across four different music classification tasks. Additionally, we showcase that a foundation model might be the key to CL in music classification. To that end, we study a new approach called *Pre-trained Class Centers*, which leverages pre-trained features to create fixed class-center spaces. Our results reveal that existing CL methods struggle when applied to music classification tasks, whereas this simple method consistently outperforms them. This highlights the need for CL methods tailored specifically for music classification.

1. INTRODUCTION

Music Information Retrieval (MIR) is a multidisciplinary field dedicated to retrieving information from music sources [1]. Within the MIR domain, music classification stands as one of the most widespread research topics [2]. It involves the categorization of music into various predefined classes, with these categories defining the ultimate task at hand. There is a diverse range of classification tasks, including genre classification [3], vocal technique identification [4], instrument classification [5], and singer identification [4], among others. These tasks are essential for organizing and retrieving music efficiently, enabling applications such as recommender systems and music search engines to better serve the needs of users in the ever-evolving music landscape [6].

Traditional music classification approaches predominantly relied on signal processing methods, accompanied

by heuristics and handcrafted features, to categorize music data [7, 8]. However, these schemes often struggled to capture the complex and nuanced aspects of musical content, thus limiting their practical application. With the rise of Deep Learning (DL) strategies, alternative solutions emerged to ease this task [9, 10]. DL models address these issues by automatically learning hierarchical representations from the data itself, thereby improving the accuracy and flexibility of music classification systems.

However, DL models become static once they are trained; their feature space is fixed. Consequently, they may struggle or fail to accommodate new classes. This does not align well with the dynamic nature of music itself—characterized by evolving genres, emerging artists, and shifting musical trends. We could approach this challenge in two ways: either (i) retrain the model from scratch when new music data is introduced, which is computationally expensive, inefficient, and not always possible due to privacy or storage issues [11], or (ii) fine-tune the model only on the newly acquired music data. The latter alternative is known to lead to the so-called “catastrophic forgetting”, where the knowledge acquired from previous data diminishes as new information is incorporated [12]. This situation highlights the need for robust and adaptable music classification systems that can be updated with just new data.

Continual Learning (CL) promises a solution to catastrophic forgetting by enabling models to gradually incorporate new knowledge without forgetting previously acquired information [11, 13]. Fig. 1 graphically depicts this scenario. This adaptability is vital for music classifiers to stay up-to-date, ensuring they can accurately categorize a continuously evolving musical landscape. While some previous works in zero-shot [14] and few-shot learning [15] propose methods for recognizing new, unseen classes, they do not maintain nor update the knowledge acquired in one session in subsequent sessions. In contrast, our work introduces the use of CL in music classification, with the goal of not only recognizing unseen classes but also retaining this knowledge over time.

CL approaches are generally classified according to the following taxonomy [16]: (i) *data-centric* methods, which focus on preserving important data from previous tasks using *data replay* or *data regularization* techniques; (ii) *model-centric* methods, which focus on model development through *parameter regularization* or model structure



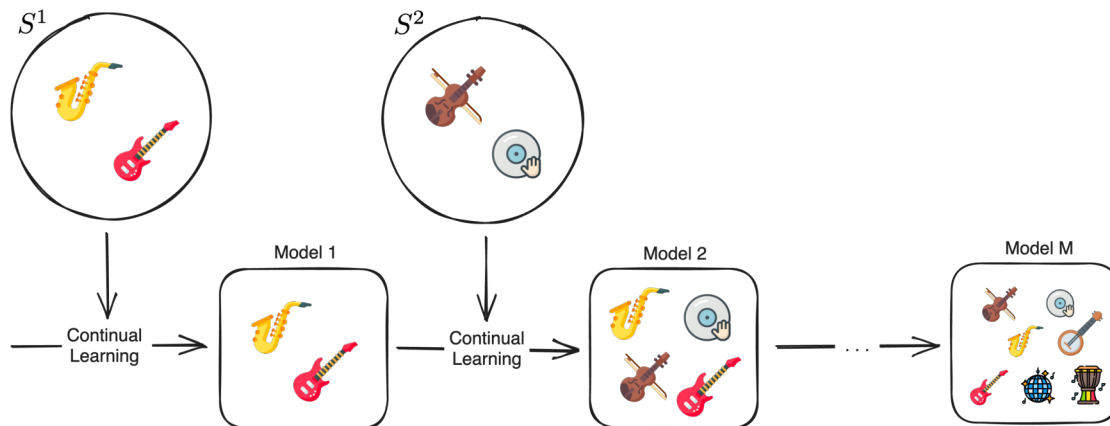


Figure 1. Graphical representation of class-incremental learning for music classification tasks. The process begins with Model 1, trained to differentiate some initial classes during session S^1 . Through a continual learning algorithm, Model 1 preserves its acquired knowledge while incorporating new classes in a subsequent learning session, S^2 , thus evolving into Model 2. This iterative learning process continues, enabling the model to progressively expand its repertoire of recognizable classes.

expansion, i.e., *dynamic networks*; and, (iii) *algorithm-centric* methods, which focus on the learning process itself, employing *knowledge distillation* techniques or *rectifying model biases*.

In this work, we investigate the applicability of state-of-the-art CL techniques, originally designed for computer vision, in the context of music classification tasks. Additionally, we explore the use of foundation models to introduce a new CL method based on pre-trained representations to create fixed class-centers, showcasing the utility and robustness of foundational models. Our results reveal that existing CL methods, traditionally evaluated on image classification, struggle when applied to music, whereas our proposed method consistently outperforms them. This raises questions regarding the effectiveness and transferability of existing CL techniques to music classification. Moreover, it prompts us to consider whether leveraging foundational models might represent a better approach for addressing the CL paradigm in certain scenarios.

To summarize, the contributions of this work are as follows: (i) a first-time analysis of the applicability of CL techniques to music classification tasks, (ii) the introduction of a simple yet effective CL method that relies on the generalizability of large pre-trained models, and (iii) extensive experimentation to quantitatively evaluate five different CL approaches across four music classification benchmarks with two different pre-trained feature extractors.

2. METHODOLOGY

In this work, we address the music classification task from a CL perspective, with a specific focus on Class-Incremental Learning (CIL). In each learning *session*, the model is trained with new audio tracks from a new set of classes. Ideally, the model should learn to classify the new classes introduced in each session while retaining its capacity to classify classes from previous sessions (see Fig. 1).

Formally, let us assume a sequence of M training sessions $\{S^1, S^2, \dots, S^M\}$, where each session has a different set of non-overlapping classes. $S^m = (X_m, Y_m)$ represents the m -th incremental step, with X_m containing audio tracks whose labels belong to Y_m , and Y_m denoting the label space of session m , where $Y_m \cap Y_{m'} = \emptyset$ for $m \neq m'$. Note that the audios in X_m are in the format $[0, 1]^{l_j \times c}$, being l_j the length of the j -th audio.¹ In this work, we consider mono audio signals as input ($c = 1$), although other considerations may be applicable. After each session, the model is evaluated on all seen classes $Y_m = Y_1 \cup \dots \cup Y_m$. The main objective of CIL is to sequentially build a classification model capable of classifying all seen classes.

2.1 Classification model

For this learning framework, our classification model consists of a fixed pre-trained model, serving as the feature extractor, and an out-of-the-box fully connected network, acting as the downstream task classifier. We use this *same* learning framework with different CL strategies to compare their performance. In order to make our experimentation agnostic to a certain degree to the pre-trained model selected as the feature extractor, we consider two state-of-the-art pre-trained models:

1. **MERT** [10] is a recently released foundational model specifically designed for extracting rich representations from music data. It follows a self-supervised pre-training paradigm that relies on two teacher models, one for the acoustic aspect and one for the musical aspect, to generate pseudo-labels for sequential audio clips. This multi-task paradigm allows for a balanced acoustic and musical representation learning, guiding a BERT-style transformer encoder to better model music audio. Its state-of-the-art performance across various MIR tasks, including

¹ Audio chunks of l_j are considered to accommodate different lengths, as typically done in the literature.

those relevant to this work, makes it a compelling choice for our purposes.

2. **CLMR** [17] adapts the image self-supervised learning strategy SimCLR [18] to the domain of music. This method employs contrastive learning to train a convolutional feature extractor [19] to extract meaningful and transferable representations from music data. It achieves this by learning to predict similar representations for slightly altered versions of the same audio sample. We consider CLMR to be a robust classification model for our work, given its demonstrated effectiveness across various music classification tasks, along with its lightweight architecture.

A summary of the characteristics of these two models can be seen in Table 1. Both of these pre-trained models use raw audio samples as inputs. We chose them with the presumption that their differences in architecture and size would enable us to extract more nuanced insights and conclusions from our experiments. It is worth noting that, in order to have comparable results with recent research works, we adhere to the same evaluation protocol as outlined in [10].

Table 1. Overview of the feature extractor models used in this work, depicting their characteristics (architecture, number of trainable parameters, input audio length, and feature embedding size).

	Architecture	Audio length (s)	Embedding Size
MERT	Transformer (94.9M parameters)	5	764
CLMR	CNN (2.4M parameters)	2.7	512

2.2 Selected methods

We select a diverse range of state-of-the-art methods in CL, emphasizing the inclusion of methods from different subtypes across the entire taxonomy. Specifically, we consider five CL approaches:

1. **Replay** aims to prevent catastrophic forgetting by employing a data-centric approach, which involves revisiting past data during the learning process [20].
2. **GEM** adopts a data-centric strategy based on data regularization to stabilize continuous training. It constrains the model’s parameter updates to prevent significant forgetting of previously learned tasks, ensuring a balanced learning experience over time.
3. **EWC** employs a model-centric approach through parameter regularization [21]. It assigns importance to specific parameters based on their relevance in previously learned tasks, thereby preventing excessive adjustments during subsequent training on new tasks.
4. **L2P** utilizes a model-centric approach based on dynamic networks [11]. It aims to learn to prompt a pre-trained Transformer to adapt it to the new

tasks, managing both task-invariant and task-specific knowledge while maintaining model plasticity.²

5. **iCaRL** adopts an algorithm-centric strategy of knowledge distillation [22]. It leverages distillation from frozen models of past learning sessions, combined with data replay, to avoid forgetting.

As a baseline method, we fine-tune the model for each session without applying a CL strategy, referred to as **Fine-tune** in the experiments, following the fine-tuning protocol used in state-of-the-art research [11]. This serves as our lower bound, potentially leading to the strongest occurrence of catastrophic forgetting.

2.3 Pre-trained Class Centers

In addition to the CL methods considered, we explore a novel approach that relies on the generalizability of the representations of a foundation model. We use this method to showcase the potential efficacy of using pre-trained models with self-supervised learning for CL. The idea is to use the latent representations produced by pre-trained models to capture the underlying semantics of the data itself, causing these representations to be distributed in a way that enables classification. Our approach seems particularly well-suited for music classification tasks because there exist publicly available foundation models (e.g., MERT and CLMR) known for their strong generalization capabilities. The proposed method, termed as *Pre-trained Class Centers* (PCC), can be separated into three different stages:

1. **Feature Extraction.** We extract a pre-trained feature embedding for each training sample.
2. **Prototype Generation.** We compute *prototype* class-centers by averaging all the feature embeddings obtained for each class in a training session and store them in a prototype buffer.
3. **Similarity Calculation.** During the inference phase, the class of a given test audio track is determined by the class associated with the nearest class-center, calculated through the Euclidean distance between the pre-trained feature vector of the test audio and the class-center prototype.

Although PCC is conceptually simple, it has never been considered. The method belongs to the data-centric category of CL methods because it focuses on leveraging the generalizability of pre-trained representations. One notable advantage is its memory efficiency (only one prototype per class), making it suitable for scenarios with limited computational resources. Just as important, this method stores representations rather than the original data, thus avoiding privacy issues. Furthermore, in PCC, the training process for each new class is independent of the other classes, making it a robust method for CL.

² Given that this method assumes a Transformer architecture as the backbone, it cannot be evaluated with CLMR.

3. EXPERIMENTATION

This section encompasses the experimental setup, including the music classification tasks, evaluation protocol, and implementation details.

3.1 Tasks

To conduct an extensive and diverse analysis, we evaluate the selected CL methods on four distinct music classification tasks using three different datasets:

Genre classification estimates the most appropriate genre for a given song. We use the standard curated split of the GTZAN dataset [23, 24], which consists of 930 30-second audio tracks from 10 different genres.

Instrument classification determines the specific musical instrument present within a given sound. We consider the NSynth dataset [5], which contains 306 000 4-second audio samples of an instrument playing a single note. There are 11 instrument classes in this dataset. Due to the high computational cost associated with the large size of the training partition,³ we consider only 5 000 training samples for each class while keeping the validation and test sets intact.

Singer identification classifies the identity of a given vocal performer in an audio track. We employ the VocalSet dataset [4], which comprises 3 613 recordings of variable length from 20 professional singers performing using different vocal techniques.

Vocal technique detection recognizes the specific singing technique present within a given audio recording. We resort to the aforementioned VocalSet dataset, considering a subset of 10 different singing techniques, consisting of 1 736 audio samples, similar to referenced work [4].

For all tasks, we consider a sequence of $M = 5$ training sessions. Given a task comprised of C different classes,⁴ each session will have an equally distributed randomly selected subset of C/M non-overlapping new classes. To avoid any bias related to the order of the sessions or the order in which the classes are learned, we report the average performance over three scenarios, each with a different sequence of training sessions. In each scenario, we randomly arrange the classes and create random groups of C/M classes. Our goal is to obtain a better estimate of the expected performance of the CL methods under unknown learning situations. Table 2 provides a summary of the characteristics of CL paradigm posed for each task.

3.2 Implementation details

As mentioned in Section 2, our classification model comprises two fundamental components: a feature extractor, which can be either a MERT or CLMR model, and a downstream task classifier. The feature extractors are used out-

³ For each task, we launched 212 training processes following the experimental setup considered (2 feature extractors \times (3 scenarios \times 7 CL methods \times 5 sessions + 1 oracle baseline)).

⁴ Each dataset is balanced, i.e., the same number of samples, or a very similar number, is considered for each class.

Table 2. Overview of the continual learning scenario posed for each music classification task: the number of learning sessions, the total number of classes, and the number of classes per session.

Classification task	Number of learning sessions, M	Total number of classes, C	Classes per learning session, C/M
Genre		10	2
Instrument	5	11	2*
Singer		20	4
Vocal Technique		10	2

*The remaining class is randomly introduced in one of the learning sessions, i.e., there is one session with 3 classes.

of-the-box.^{5 6} These remain frozen during training not only for efficiency but also to improve stability and mitigate the effects of forgetting in CL. The classifier is an MLP with 512 hidden units. When using MERT, a one-dimensional convolutional layer is employed prior to the MLP to extract a weighted average embedding from the frame-level features obtained by MERT.

We follow the details provided in the work of Li et al. [10] to train the architecture described previously for the different considered tasks. To attain state-of-the-art results while keeping the feature extractor frozen, we train the downstream classifier for a maximum of 200 epochs using the ADAM optimizer with a fixed learning rate of 10^{-3} and a batch size of 64 audio chunks. We use early-stopping with the number of patience epochs adjusted accordingly to each task. Additionally, we employ a 25% dropout rate to mitigate overfitting and improve performance.

For the methods that require data storage from past sessions (Replay, GEM, iCaRL), we use a memory buffer of 100 memories equally distributed among the classes seen up to that session, following the implementation used in [25]. Moreover, we use PyTorch as the implementation framework. We rely on the PyCIL toolbox⁷ for all the considered CL methods, except for L2P, for which we adhere to the official implementation.⁸

The length of the audio chunks used for training and evaluating the models depends on the feature extractor used and can be seen in Table 1. For the task of singer identification and vocal technique, we use 3-second audio chunks as input, as in previous works [4, 10]. Finally, regarding the evaluation protocol, we segment each audio file into chunks (as aforementioned) and obtain a prediction for each chunk. The predictions for each chunk are then averaged to obtain a final prediction for each given audio file.

4. RESULTS

Fig. 2 reports the average performance of each method for each learning session in terms of classification accuracy.⁹

⁵ MERT’s weights available at <https://huggingface.co/m-a-p/MERT-v1-95M>

⁶ CLMR’s weights available at <https://github.com/Spijkervet/CLMR>

⁷ <https://github.com/G-U-N/PyCIL>

⁸ <https://github.com/google-research/l2p>

⁹ The code developed in the work is publicly available for reproducible research at: <https://github.com/pedrocg42/continual-music-classification>

Finetune Replay iCaRL GEM EWC L2P PCC

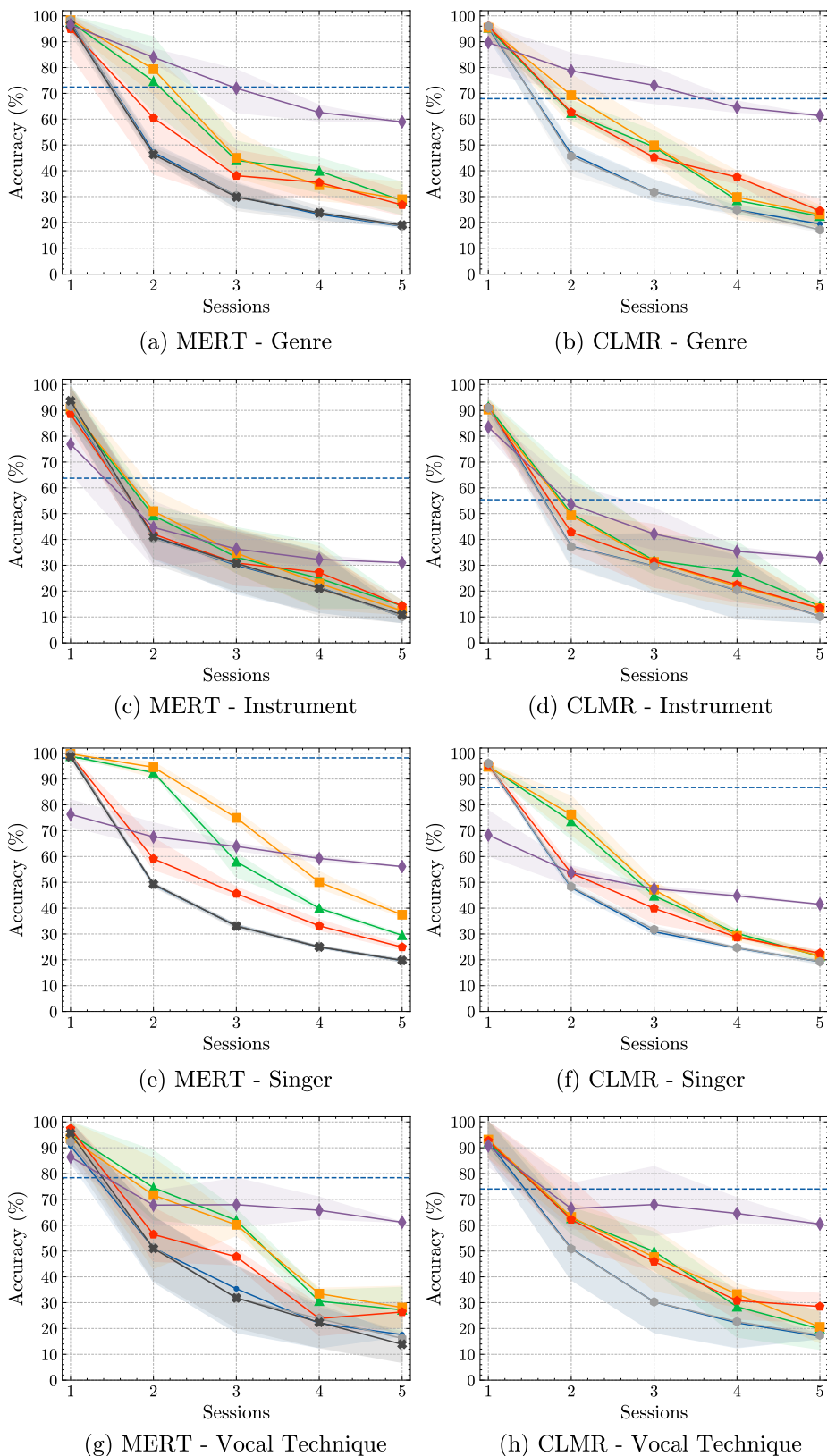


Figure 2. Accuracy (%) per session for each CL method. The solid lines represent the average accuracies, while the shaded areas indicate the minimum and maximum accuracies for each method and session. The dashed line represents the reference accuracy achieved when directly training with all classes in a single session.

We report as well the average accuracy along all the training sessions in Table 3 and the accuracy after the last session in Table 4.

Table 3. Comparison of the averaged accuracy after each session across the four tasks.

Task	Genre		Instrument		Singer		Vocal Tech	
	MERT	CLMR	MERT	CLMR	MERT	CLMR	MERT	CLMR
Finetune	43.4	43.6	38.6	37.9	45.3	43.7	43.4	42.6
Replay	56.9	51.6	42.4	43.0	63.8	53.0	57.9	50.6
iCaRL	57.2	53.5	42.4	41.3	71.4	53.8	57.4	51.6
GEM	51.1	53.2	40.6	40.2	52.4	48.0	50.4	52.0
EWC	43.5	43.1	38.7	37.6	45.4	44.0	43.1	42.4
L2P	43.1	-	39.5	-	45.1	-	42.9	-
PCC	74.8	73.5	44.2	49.5	64.6	51.2	69.8	70.0
Oracle	75.2	70.9	62.7	57.1	97.8	86.0	76.3	72.9

Table 4. Comparison of the final accuracy after the last session across the four tasks.

Task	Genre		Instrument		Singer		Vocal Tech	
	MERT	CLMR	MERT	CLMR	MERT	CLMR	MERT	CLMR
Finetune	18.5	19.4	10.1	10.3	19.7	19.3	17.6	17.0
Replay	28.4	22.4	14.4	14.3	29.5	21.3	27.2	19.8
iCaRL	29.0	23.1	12.4	13.4	37.5	21.6	28.1	20.6
GEM	26.8	24.5	14.3	13.4	24.9	22.5	26.4	28.5
EWC	18.6	17.1	10.1	10.1	19.8	19.3	16.2	17.4
L2P	19.0	-	10.9	-	19.8	-	13.9	-
PCC	59.0	61.4	31.0	32.9	56.1	41.5	61.1	60.5
Oracle	72.4	67.9	63.7	55.4	98.2	86.7	78.4	74.0

The first observation is the limitation of the existing CL literature, where methods are primarily evaluated over well-established computer vision tasks [11]. As evidenced by the reported results, such methods fall short in terms of generalization to other domains. Specifically, the considered state-of-the-art CL methods suffer from significant catastrophic forgetting, resulting in poor final performance for class-incremental music classification. This underscores the need to develop CL methods for this specific domain and encourages the assessment of CL methods across different fields to measure their overall performance more precisely.

Focusing our attention on the similarities illustrated in Fig. 2 for the two different feature extractors, MERT (left column) and CLMR (right column), we can observe that the accuracy curves for all tasks exhibit very similar trends across sessions. While the baseline performance—training directly with all data in a single session—is better when using MERT, this difference between the two feature extractors diminishes when comparing against the different CL methods, as similar performance is achieved. Consequently, we can conclude that the effectiveness of the CL methods is not solely attributable to the feature extractor.

If we examine each task separately, we observe a similar pattern in both music genre and vocal technique classification tasks. State-of-the-art methods exhibit signs of catastrophic forgetting, whereas PCC achieves a final performance that is relatively close to the reference bound. For singer identification, PCC starts with a lower accuracy but maintains good stability throughout the sessions.

However, despite achieving the highest final performance among the methods, it still falls considerably short of the task reference. Finally, instrument classification emerges as the most challenging task, with all methods displaying significant signs of catastrophic forgetting. As a result, their final performance remains far from reaching reasonable results, once again highlighting the existing room for improvement and the need to find new methods that can reduce catastrophic forgetting in music classification.

Among the considered state-of-the-art CL methods, both data-centric (Replay and GEM) and algorithm-centric (iCaRL) approaches outperform the results obtained by model-centric methods (EWC and L2P). However, it is worth noting that these first three methods rely on input data stored from previous sessions, which may not always be feasible due to privacy or storage issues. In contrast, our proposed method, PCC, remarkably surpasses all of them across all tasks without storing the original data (but their representations), thus avoiding such privacy issues.

5. CONCLUSIONS

This work studies the goodness of five state-of-the-art CL methods (Replay, EWC, iCaRL, GEM, and L2P) in the context of CIL for music classification. Additionally, we propose a simple yet effective CIL method (PCC) that relies on the generalizability of foundation models.

Our results reveal that current state-of-the-art CL methods suffer from catastrophic forgetting, whereas the proposed approach achieves the best results over four different music classification tasks. This highlights the need to investigate specific CL methods for music classification.

The results obtained with PCC showcase the robustness and utility of the features extracted with foundation models. We can only expect these models to improve over time, managing to extract more generalizable features for a wider range of tasks. This leads us to believe that it is worth further exploring these approaches for CL in music classification.

In future work, we also plan to study more sophisticated strategies for selecting the prototypes in PCC to improve both accuracy and robustness.

6. ACKNOWLEDGEMENTS

This paper is supported by grant CISEJI/2023/9 from “Programa para el apoyo a personas investigadoras con talento (Plan GenT) de la Generalitat Valenciana”.

7. REFERENCES

- [1] J. S. Downie, “Music Information Retrieval,” *Annual Review of Information Science and Technology*, vol. 37, no. 1, pp. 295–340, 2003.
- [2] Z. Fu, G. Lu, K. M. Ting, and D. Zhang, “A Survey of Audio-Based Music Classification and Annotation,” *IEEE Transactions on Multimedia*, vol. 13, no. 2, pp. 303–319, 2010.

- [3] N. Pelchat and C. M. Gelowitz, "Neural Network Music Genre Classification," *Canadian Journal of Electrical and Computer Engineering*, vol. 43, no. 3, pp. 170–173, 2020.
- [4] J. Wilkins, P. Seetharaman, A. Wahl, and B. Pardo, "VocalSet: A Singing Voice Dataset," in *Proceedings of the 19th International Society for Music Information Retrieval Conference*, Paris, France, 2018, pp. 468–474.
- [5] J. Engel, C. Resnick, A. Roberts, S. Dieleman, M. Norouzi, D. Eck, and K. Simonyan, "Neural audio synthesis of musical notes with WaveNet autoencoders," in *Proceedings of the 34th International Conference on Machine Learning - Volume 70*. PMLR, 2017, pp. 1068–1077.
- [6] M. Schedl, "Deep Learning in Music Recommendation Systems," *Frontiers in Applied Mathematics and Statistics*, p. 44, 2019.
- [7] M. F. McKinney, "Features for Audio and Music Classification," in *Proceedings of the 6th International Society for Music Information Retrieval Conference*, 2003.
- [8] M. I. Mandel and D. P. Ellis, "Song-Level Features and Support Vector Machines for Music Classification," in *Proceedings of the 6th International Society for Music Information Retrieval Conference*, 2005.
- [9] K. Choi, G. Fazekas, M. Sandler, and K. Cho, "Convolutional recurrent neural networks for music classification," in *Proceedings of the 42nd IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2017, pp. 2392–2396.
- [10] Y. Li, R. Yuan, G. Zhang, Y. Ma, X. Chen, H. Yin, C. Lin, A. Ragni, E. Benetos, N. Gyenge *et al.*, "MERT: Acoustic Music Understanding Model with Large-Scale Self-supervised Training," *arXiv preprint arXiv:2306.00107*, 2023.
- [11] D.-W. Zhou, Q.-W. Wang, Z.-H. Qi, H.-J. Ye, D.-C. Zhan, and Z. Liu, "Deep class-incremental learning: A survey," *arXiv preprint arXiv:2302.03648*, 2023.
- [12] M. McCloskey and N. J. Cohen, "Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem," in *Psychology of Learning and Motivation*. Elsevier, 1989, vol. 24, pp. 109–165.
- [13] M. De Lange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars, "A Continual Learning Survey: Defying Forgetting in Classification Tasks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 7, pp. 3366–3385, 2021.
- [14] J. P. Jeong Choi, Jongpil Lee and J. Nam, "Zero-shot learning for audio-based music classification and tagging," in *Proceedings of the 20th International Society for Music Information Retrieval Conference*. ISMIR, Nov. 2019, pp. 67–74. [Online]. Available: <https://doi.org/10.5281/zenodo.3527741>
- [15] Y. Wang, N. J. Bryan, M. Cartwright, J. Pablo Bello, and J. Salamon, "Few-shot continual learning for audio classification," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 321–325.
- [16] D.-W. Zhou, Q.-W. Wang, Z.-H. Qi, H.-J. Ye, D.-C. Zhan, and Z. Liu, "Deep Class-Incremental Learning: A Survey," *arXiv preprint arXiv:2302.03648*, 2023.
- [17] J. Spijkervet and J. A. Burgoyne, "Contrastive learning of musical representations," in *Proceedings of the 22nd International Society for Music Information Retrieval Conference*. ISMIR, Oct. 2021, pp. 673–681.
- [18] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PMLR, 2020, pp. 1597–1607.
- [19] J. Lee, J. Park, K. L. Kim, and J. Nam, "Samplecnn: End-to-end deep convolutional neural networks using very small filters for music classification," *Applied Sciences*, vol. 8, no. 1, 2018.
- [20] R. Ratcliff, "Connectionist models of recognition memory: constraints imposed by learning and forgetting functions," *Psychological Review*, vol. 97, no. 2, p. 285, 1990.
- [21] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Rammalho, A. Grabska-Barwinska *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the National Academy of Sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.
- [22] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "iCaRL: Incremental Classifier and Representation Learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5533–5542.
- [23] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, pp. 293–302, 2002.
- [24] C. Kereliuk, B. L. Sturm, and J. Larsen, "Deep Learning and Music Adversaries," *IEEE Transactions on Multimedia*, vol. 17, no. 11, pp. 2059–2071, 2015.
- [25] Z. Wang, Z. Zhang, C.-Y. Lee, H. Zhang, R. Sun, X. Ren, G. Su, V. Perot, J. Dy, and T. Pfister, "Learning to Prompt for Continual Learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 139–149.

THEGLUENOTE: LEARNED REPRESENTATIONS FOR ROBUST AND FLEXIBLE NOTE ALIGNMENT

Silvan David Peter¹

Gerhard Widmer^{1,2}

¹ Institute of Computational Perception, Johannes Kepler University Linz, Austria

² LIT AI Lab, Linz Institute of Technology, Austria

ABSTRACT

Note alignment refers to the task of matching individual notes of two versions of the same symbolically encoded piece. Methods addressing this task commonly rely on sequence alignment algorithms such as Hidden Markov Models or Dynamic Time Warping (DTW) applied directly to note or onset sequences. While successful in many cases, such methods struggle with large mismatches between the versions. In this work, we learn note-wise representations from data augmented with various complex mismatch cases, e.g. repeats, skips, block insertions, and long trills. At the heart of our approach lies a transformer encoder network — TheGlueNote¹ — which predicts pairwise note similarities for two 512 note subsequences. We postprocess the predicted similarities using flavors of weightedDTW and pitch-separated onsetDTW to retrieve note matches for two sequences of arbitrary length. Our approach performs on par with the state of the art in terms of note alignment accuracy, is considerably more robust to version mismatches, and works directly on any pair of MIDI files.

1. INTRODUCTION

Note alignment refers to the task of matching individual symbolically encoded notes in two versions of the same piece. Note matches can be derived for any two versions, however, this task is usually addressed for pairs of MIDI performances and scores encoded in various formats. The resulting performance-to-score alignments provide the data for several research directions in MIR and computational musicology, such as expressive performance generation, score quantization, and performance research.

To match notes can sometimes be a near-trivial task, especially with well corresponding versions, minimal expressive playing, and simple pieces. However, more often than not the unaligned data of interest and availability does not fit these criteria: performers make mistakes; play extra

repeats, variations, and ornamentations; rehearsal recordings discontinue or restart; automatic transcriptions contain various amounts of note mismatches; and the musical material tends towards the virtuosic, dense, and complex.

Due to its close similarity with sequence alignment, note alignment is usually approached with flavors of Dynamic Time Warping (DTW) or Hidden Markov Models (HMM) based on note or chord representations. Such representations are typically localized, and the alignment methods process them sequentially. The aforementioned common difficulties in MIDI performances do not harmonize well with these constraints: e.g., differently ordered chord onsets clash with DTW’s monotonicity condition, trills create (sometimes substantial) mismatches with similar pitches and thus misleading local distances, and repeats, skips, recording takes, etc. introduce large mismatches between the sequences which require a more zoomed-out perspective. To be clear, there is nothing that a priori prevents sequence alignment methods from working in these scenarios, however, in practice, the propensity of alignment methods for propagating errors render the matching quality hit-and-miss.

In this work, we address note alignment via learned representations which leverage non-local information, i.e., the entire sequence of notes influences the representation of each note. We train an attention-based encoder — TheGlueNote — to predict note representations for two 512 note subsequences. Before being passed to the network, the subsequences are augmented with a variety of challenging and large mismatch cases. At the network’s output, we compute a pairwise similarity matrix between the note representations and compare this matrix to target note matches via two classification loss terms. That is, the notes are guided towards similar representations if they match, and dissimilar representations for all others. In the process, TheGlueNote is trained to robustly predict note similarities even in the presence of substantial mismatches.

We took care to design TheGlueNote as annotation-agnostic as possible. Prior approaches mitigate edge cases by introducing additional submethods, e.g., by modeling left-right hand streams separately, excluding notated ornaments from certain steps, or requiring coinciding chord notes (see section 2). This introduces limitations on the types of files which can be processed, requiring staff or voice information, scores with ornament information, or even just quantized scores. In contrast, our model is trained

¹ <https://github.com/sildater/theglue-note>



directly on data from MIDI files with no quantization or score annotation requirement.

To extract final note matches from the similarity matrix, we present three possible additional methods. First, we simply match the notes with maximal similarity. Second, we add a decoder head to our note representation backbone – a network which predicts matching notes based on the similarity matrix. Third, we use DTW alignment techniques to extract a mapping from the similarity matrix and in turn use this mapping to match individual notes. Putting the pieces together, TheGlueNote leverages learned representations for note alignment, performs on par with the state of the art, excels at complex mismatch cases, and works with plain MIDI input data.

The rest of the paper is structured as follows: Section 2 discusses related work. Section 3 describes the model architecture and match extractor variants. Section 4 introduces training specifications, data processing, and metrics which are applied in Section 5 where we evaluate the trained model in an ablation study on the variations of architecture and match extraction, and in comparison with state-of-the-art methods both in regular and complex mismatch cases. Section 6, the discussion, concludes the paper.

2. RELATED WORK

Note alignment is a basic technology vital to many downstream tasks in symbolic music processing and computational musicology. We structure this review of related literature into a part on current state-of-the-art methods, relevant work in the neighboring domains of audio and real-time alignment, and pertinent literature on matching tasks for non-music data.

Note alignment methods almost universally make use of either Dynamic Time Warping or Bayesian Networks on pitch-based representations of either individual notes or chords [1–9]. As the principal formulation is straightforward, most recent efforts have focused on formalizations and heuristics that mitigate specific problems in edge cases. Skips and repeats present a major difficulty which can be directly modelled at the cost of computational complexity [5] or side-stepped if the use of annotated anchor points is possible [8]. Another difficulty are ornamentation notes which can be modelled as separate states [4] or excluded from a first coarse alignment and handled separately in a fine-grained note matching step [9]. Nakamura et al. [3] further model left-right hand asynchrony in piano performance. In their most recent work, note alignment is framed as a hierarchical refinement with explicit modelling of an alignment error identification step [6]. In recent work by Peter [9], sequence non-ordinality is mitigated by a score-based chord representation, the resulting model is thus limited to performance to score alignment. The current state of the art (SOTA) which we will use for reference in this work is given by two DTW-based methods [8, 9] and one HMM-based method [6].

Realtime alignment or score following methods have been developed since the 1980s [10, 11] and largely mir-

ror the previous methods in their core elements: On-Line Time Warping (OLTW), [12] and Bayesian Networks, in particular HMM [3, 7, 13].

Alignment of musical audio is an important idea generator for symbolic note alignment. For introductions of audio alignment, we refer the reader to Arzt [14] and Müller [15]. Audio alignment is prone to memory and compute bottlenecks. Several versions of DTW addressing these issues have been developed [16].

Deep Learning has largely been absent from music alignment, with notable exceptions in real-time audio-image matching [17, 18] and in symbolic score following [9]. On the other hand, we take inspiration from image processing, in particular from the task of local feature matching [19]: the matching of pixels encoding the same location on an object in two images of the same object. Local feature matching often uses neural network-based feature extractors [20, 21] and our proposed model in particular is informed by the MatchFormer [21].

3. MODEL

In this article, we present a model which is trained to create note representations for two sequences such that the representations’ pairwise similarity corresponds to the sequences’ alignment ground truth. Using these representations, we aim to uniquely match individual notes. The proposed model consists of a fixed-length tokenization, an encoder backbone, and a dual classification loss. Furthermore, we introduce three variants of match extraction from the model’s output similarity matrix: direct similarity matrix processing, using a decoder head for classification, and DTW-based match extraction. Figure 1 presents an overview of the components.

3.1 TheGlueNote

At the heart of our model is a non-causal transformer encoder (see Figure 1 middle left). Its purpose is to learn note representations for two note sequences s_1 and s_2 , and its target is the alignment between the sequences. A pairwise similarity matrix computed between the note representations of two sequences mediates between output and target. We treat the matrix as a match classifier for each note, i.e., for each row (a note in s_1) the column (a note in s_2) with maximal similarity should correspond to matching notes, and vice versa.

Technically, two at least partially matching subsequences s_1 and s_2 of 512 notes each are prepended with a default note and processed using the fixed-length structured tokenization [22, 23], which encodes relative onset, pitch, duration, and velocity. The now 513 note (2052 token) sequences s_1 and s_2 are concatenated and passed to the encoder. The encoder sums the four tokens per note and adds a learned positional embedding for a note-wise sequence of length 1026. Layer normalization is applied before the first encoder block and within the attention and feedforward blocks but not again on the residual stream. Self-attention is applied to the full concatenated

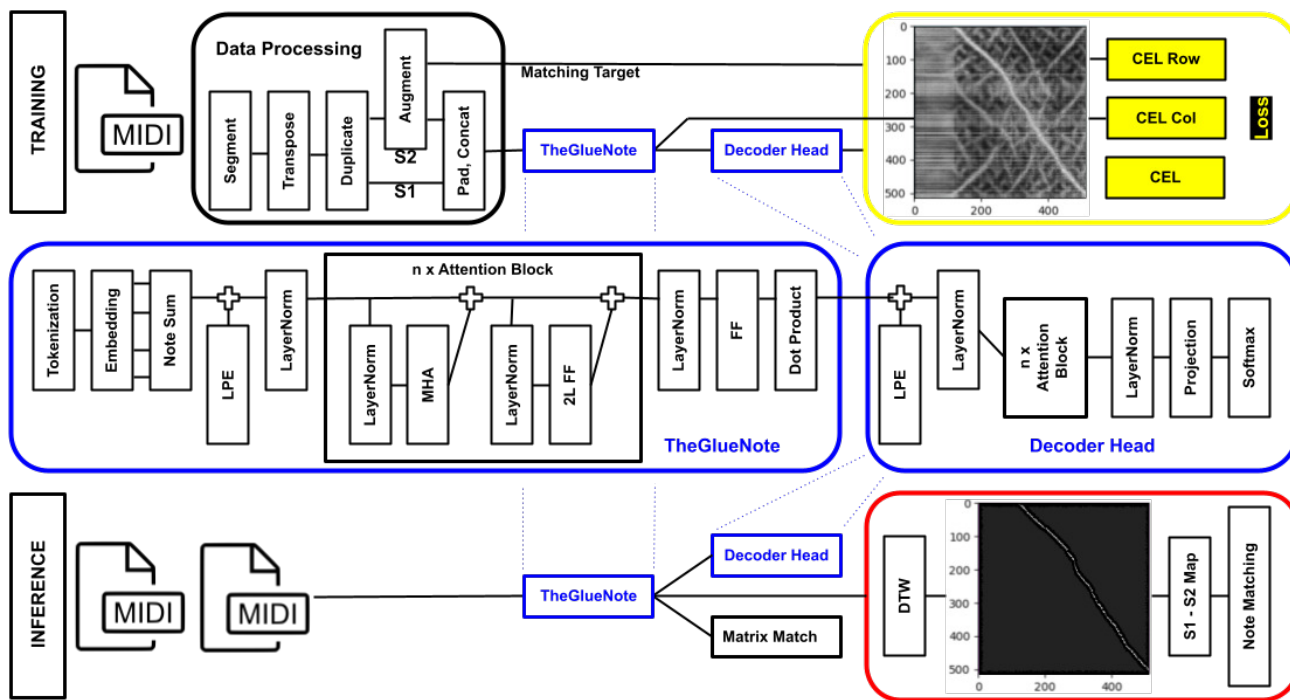


Figure 1. Overview of the proposed model. During **training** (top row), the data flows from Data Processing (black, top left) through TheGlueNote (blue, middle left) into the Decoder Head (blue, middle right) and the aggregated Loss (yellow, top right). Concretely, a MIDI file is loaded into the Data Processing module which outputs matching targets to the loss module and the concatenated sequences s_1 and s_2 to TheGlueNote. TheGlueNote (middle row) consist of a transformer encoder with learned positional embeddings (LPE) and repeated attention blocks (center module with multihead-attention MHA and a two-layer feedforward network 2L FF). The note-wise representations are split and multiplied for a pairwise similarity matrix with s_1 in the row and s_2 in the column dimension shown in the loss module. Two cross-entropy loss terms are computed from this matrix and it is also forwarded to the decoder head whose classifier output adds a third loss term. During **inference** (bottom row), two MIDI files to be matched are directly passed to TheGlueNote. The resulting similarity matrix can be processed in three ways: 1) direct maximal similarity match extraction (Matrix Match box) 2) using the decoder head’s output, or 3) using a DTW-based match extraction (red, bottom right).

sequence, which amounts to combined within-sequence attention (s_1-s_1, s_2-s_2) and between-sequence attention (s_1-s_2, s_2-s_1). The final residual is normalized after the last block and fed through a single dimension-conserving linear layer. For dimensions and hyperparameters of different versions see section 4.2 and in particular Table 3.

The 1026 final output vectors of size corresponding to the residual stream are treated as representations of individual notes. The sequences are split again and pairwise similarities between all 513 notes of s_1 and all 513 notes of s_2 are obtained via dot product. The resulting similarity matrix (s_1 in the rows and s_2 in the columns, see Figure 1 top right) is compared to two classification targets: Softmax across row dimension is the model’s prediction of the matching note in s_1 for each note in s_2 (except for the prepended default note), and softmax across the column dimension is the model’s prediction of the matching note in s_2 for each note in s_1 (again, except for its prepended default note). Both are compared against the ground truth via a cross-entropy loss (CEL). Unmatched notes in the ground truth receive a target corresponding to the default note in the other sequence, the default notes itself receive no loss.

3.2 Match Extractors

Note similarities are a useful intermediary, however, they do not yet define note matches. In this section, we detail three possible note match extractors. The simplest way of producing matches is to directly use **similarity matrix-based match extraction**. That is, for each note in s_2 , we match it to its most similar note in s_1 , including the default note (=unmatched) as possibility. A little bit of index housekeeping avoids conflicting matches and notes without prediction.

A second approach is to train TheGlueNote with an additional **decoder head for match extraction**. The decoder is also a non-causal neural network with the same high-level structure as the encoder² (see Figure 1 middle right). The decoder head processes the pairwise similarity matrix for each actual note of s_2 (hence a 512 by 513 matrix, excluding the default note in s_2) and directly predicts the matching note in s_1 via 513 output logits (including the default note in s_1 for unmatched notes). During training, its classification CEL is added to the other two losses in an

² The "decoder head" is technically also a transformer encoder without memory input, however, it decodes the representation towards classification logits, so we opt for this name.

unweighted fashion (see Figure 1 top right).

DTW match extraction offers a way of introducing meaningful constraints to the note matching. Concretely, naive note matching via maximum similarity in the prediction treats every note separately and ignores information on predictions for notes in its vicinity and previously matched notes. To introduce this information, we adapt the DTW-based mapping and matching procedure introduced by Peter [9] for similarity matrix post-processing.

DTW extraction is split in two processes (see Figure 1 bottom right). First, understanding similarity as the reciprocal of learned pairwise distance, we use the network’s output as input to a weighted DTW with possible directions $[[0, 1], [1, 1], [1, 0]]$ and associated weights $[1, 2, 1]$. This choice of weights normalizes the directions under the Manhattan distance, i.e., any direction is equally costly overall and the diagonal is not favored. A standard DTW path is computed starting at the top left and ending at the bottom right of the similarity matrix. Note that extracting a minimizing path through the learned distances discards information relevant to local non-ordinality in favor of added information about each note’s neighborhood. The extracted path should not be used as direct note match prediction, instead it defines a coarse sequence to sequence mapping $m : \mathbb{R} \rightarrow \mathbb{R}$ by linear interpolation between the onset times of notes in the path.

In a second process, we separate all notes in both sequences by pitch. For subsequences s_1^p and s_2^p of pitch p , we match onset sequences using a DTW pass to find onset pairs that minimize the distance between s_2^p and $m(s_1^p)$. Newly matched notes are then added to or overwritten in the original DTW path and the mapping m is updated. If the MIDI files to be aligned do not fit within the 512 note contexts of the model, which is often the case, we compute several similarity matrices for 512 note windows with a stride of 256 notes. We then aggregate the resulting output matrices to a global similarity matrix.

Feature	Noise and Mismatch Probabilities
Tempo T_t	$gT_t 2^{n_t}, g \sim \mathcal{N}(1, 0.5), n_t \sim \mathcal{N}(0, 0.5)$
Onset O_t	$O_t + n_t, n_t \sim \mathcal{U}(-50, 50)$
Velocity V_t	$V_t + n_t, n_t \sim \mathcal{U}(-10, 10)$
Duration D_t	$D_t + n_t, n_t \sim \mathcal{U}(-250, 250)$
Repeats	$\mathcal{P}_{repeat} = 1, \#note_{repeat} \sim \mathcal{U}(8, 200)$
Skips	$\mathcal{P}_{skip} = 1, \#note_{skip} \sim \mathcal{U}(8, 200)$
Insertions	$\mathcal{P}_{insertion} = 0.2, \text{random location}$
Deletions	$\mathcal{P}_{deletions} = 0.2, \text{random location}$
Trills	$\mathcal{P}_{trill} = 1, \#note_{trill} \sim \mathcal{U}(20, 100)$

Table 1. Augmentations to synthesize complex mismatch cases. Four noise terms are added to note features in the first row terms. Sampled noise is clipped to avoid degenerate cases like negative durations. Duration and onset noise are indicated in MIDI ticks. Skips, repeats, and trills are introduced with the indicated probability and uniformly sampled length. Insertions and deletions are added at random locations with overall probabilities given.

4. EXPERIMENTS

We report several experiments to assess the qualities of our proposed model. In this section, we describe the dataset, data preprocessing, and training as well as model configurations.

4.1 Data

A data sample for our model is a pair of 512 note subsequences. Note alignment ground truth data of real pieces and performances is available [8,24,25], however, this data is biased towards cases where prior note alignment methods could successfully be applied. To present the model with a wide variety of (mis)match cases we use synthetically augmented MIDI data for training. The original MIDI tracks are taken from the (n)ASAP dataset [8], albeit not its note alignments, only the score and performance MIDI files directly.

Ground truth match data is created entirely synthetically by copying each MIDI file and augmenting it with a combination of the processes which we describe in the following, and whose parameters are given in Table 1. The original inter-onset intervals (as a proxy for tempo) are stretched by a global factor g , and by note-wise factors n_t , these factors are multiplicative and normally distributed. Note onsets and durations are also changed note-wise, yet by additive uniform noise in MIDI ticks. All MIDI files are encoded using 480 ticks per beat and 120 beats per minute, one MIDI tick is thus slightly longer than one millisecond. Velocities are modified by additive uniform noise within the 128 standard MIDI velocity values. For repeats, skips, and trills, the probability of generating the mismatch per 512 note sequence is given, as well as note quantities sampled uniformly. The mismatches are inserted contiguously into the sequence and there is at most one augmentation of each mismatch type per 512 note sequence. Finally, insertions and deletions are generated from the existing notes, each note is deleted or copied and randomized (i.e., inserted) with the given probability. All augmentations are recomputed for every batch of training data. The values in Table 1 are given for reproducibility and transparency, although different variations were tested, we do not claim that these are optimal values.

We further add transposition to the maximal extent available on a piano keyboard as general augmentation affecting both subsequences. The augmentation is carried out in the data loader so each epoch will produce different samples from the 1032 valid MIDI files in our dataset. Data augmentation is only used during training, at inference two MIDI files are matched as is (see Figure 1).

For testing, we obtained the exact test files used in the reference literature [9]. These files stem from proprietary datasets [26, 27] and were chosen due to the alignment complexity they provide. To test for robustness under challenging mismatch scenarios, we augment these performances for an experiment including extended (100+ note) mismatches that cover approximately 20% of the notes. Each 512 note subsequence pair contains exactly two mismatching segments, one in s_1 and one in s_2 , each segment

Data Source	Vienna 4x22									Training Data		
	Sim Matrix			Decoder Head			DTW			n.a.		
Unit	Prec	Rec	F	Prec	Rec	F	Prec	Rec	F	TL	VL	VA
Pitch-Onset Similarity Matrix	7	7	7	-	-	-	85	89	82	-	-	-
TGN-large	97	97	97	96	97	96	99	100	99	0.183	0.126	0.958
TGN-mid	81	81	81	87	88	87	99	99	98	0.171	0.145	0.996
TGN-small	75	75	75	83	87	81	99	99	99	0.374	0.280	0.902

Table 2. Ablation of Model configuration and match extraction. All match results are computed on the Vienna4x22 Dataset. Values reported are average note match precision, recall, and F-score across all performances. The results are computed for each match extractor / model combination. The first line serves as a simple baseline for the similarity matrix-based and the DTW-based match extractors: we report results of their processing a pitch and onset-based similarity matrix. We further report the average training loss (TL), validation loss (VL), and validation accuracy (VA) in the final epoch for each model.

is contiguous and its notes are randomly sampled. Note that such randomized contiguous mismatches are different from the synthetic mismatch segments seen during training (i.e., trills, repeats, skips, see Table 1). For comparison with our reference models, we have to limit ourselves to score to performance alignment instead of general MIDI to MIDI alignment, as some of the compared models only work with this type of musical material. To compare different model configurations and match extractors, we further use the public Vienna 4x22 Dataset [24]. This dataset consists of 4 pieces with 22 performances each. The pieces are comparatively simple and mismatches minimal.

Model	#p	rd	#b	#h	bs	#ph
TGN-large	28M	512	8	8	8	27M
TGN-mid	5.7M	256	6	8	16	2M
TGN-small	1.1M	128	4	8	24	0.6M

Table 3. Hyperparameters for TheGlueNote (TGN) variations: #p = parameter count, rd = residual dimension, #b = number of blocks, #h = number of attention heads, bs = batch size, and #ph = parameter count of decoder head. Parameter counts of the TheGlueNote models (#p) and their decoder heads (#ph) are approximate.

4.2 Training Setup

We train model variations differentiated in three sizes. All our models are trained on a single GeForce GTX 1080 Ti with 12 GB of memory. We train for 200k steps, independent of batch size, which is set to the maximal capacity of the GPU for each model. The learning rate is initialized at $5 * 10^{-4}$ and is scheduled using cosine annealing with warm restarts at an interval of 2k steps. Table 3 details the hyperparameters for model variations. For all attention blocks, the inverted bottleneck of the feedforward network is four times the residual dimension.

5. EVALUATION

In this section, we evaluate our proposed model. The first part compares different model configurations, the second part compares against state-of-the-art reference methods. To evaluate our models, we use note matching precision,

recall, and F-score as our main metrics. We further report mean final classification losses of the predicted similarity matrix which corresponds to direct note matching on the training and validation data as well as the runtime of different model setups.

5.1 Ablation Study of Model Configurations

In a first experiment we train three model configurations. We evaluate their note matching quality on the Vienna4x22 dataset using three different match extractors: direct similarity matrix processing, decoder head prediction, and DTW-based match extraction. Table 2 details the results. For all model configurations the match extractors are clearly ranked with DTW-based processing the most promising. DTW-based match extraction in itself is, however, not enough for good matching. To illustrate this point, we compute a simple pitch and onset based similarity matrix (the closer in pitch and onset, the higher the similarity) akin to what would be used to assess local distances in standard approaches. We then apply the similarity matrix-based match extractor and the DTW-based match extractor directly on this matrix. The first row in Table 2 shows these reference methods, which perform subpar.

5.2 Comparison to Reference Models

We compare our proposed model against three SOTA reference models: Nakamura’s HMM matcher [6], Peter’s DualDTWMatcher [9] and AutomaticNoteMatcher [8]. The first one is implemented in C++ and compiled locally³, the other’s are part of a python package⁴. The test data consists of five challenging pieces for solo piano in two settings, one default and one with mismatches. Table 4 details the results. All values are note match F-scores given in percent, except for the runtime given in seconds. In the default setting, all three model configurations perform on par with the best model with the best reference model "DualDTWMatcher". In the mismatch setting, all reference models show (partial) failure. Note that not all alignments fail, however, no reference model stays consistently

³ downloaded from: https://midialignment.github.io/AlignmentTool_v190813.zip

⁴ downloaded from: <https://github.com/sildater/parangonar>

Data Source	Default Data							20 % Mismatch Data								
Piece	B. Op. 53 3rd. m.	C. Op. 9 No. 1	C. Op. 9 No. 2	C. Op. 10 No. 11	C. Op. 60	mean of 5 pieces	total runtime	B. Op. 53 3rd. m.	C. Op. 9 No. 1	C. Op. 9 No. 2	C. Op. 10 No. 11	C. Op. 60	mean of 5 pieces	total runtime		
Unit	Match F-Score in %							s	Match F-Score in %							s
Nakamura HMM	98	99	98	94	95	98	152	39	65	35	20	63	44	6458		
Peter AutomaticNoteMatcher	99	84	94	96	89	92	588	82	74	89	71	75	78	808		
Peter DualDTWMatcher	99	98	99	96	98	98	96	85	96	94	80	83	88	208		
TGN-large + DTW	99	99	98	96	97	98	33	94	96	95	93	94	94	42		
TGN-mid + DTW	96	98	98	96	98	97	27	92	95	96	92	95	94	38		
TGN-small + DTW	99	98	98	96	97	98	21	94	97	95	93	94	95	31		

Table 4. F-scores of three reference models and our proposed models across five challenging pieces. The matching results are given as f-scores in % and the runtime in seconds. The data is split in two groups: a default case with the original performances, and a mismatch case, where challenging skips and repeats which in total constitute 20% of the notes have been introduced. The models are split in two groups: three state-of-the-art reference models and our proposed model in three configurations.

above 90 %. Our proposed models take a performance loss as well, yet only in the range of 0-7 % and the F-score stays above 92 % throughout. In terms of F-score, no significant difference between TheGlueNote configurations is found. Despite several forward passes to retrieve local similarity matrices, TheGlueNote configurations also require the lowest runtime. Unlike for the reference models the runtime does not seem to vary with the complexity of the match to be performed, only with the number of notes and the network size. The advantageous runtime comparison with the reference models is surprising and to be taken with a grain of salt as implementation details possibly outweigh the merits of each algorithm.

6. DISCUSSION

In this article, we presented TheGlueNote, a note representation model which effectively predicts note matching similarities. Despite the fundamental role of (note) alignment in several MIR areas, machine learning approaches have seen limited adoption so far — in contrast with many other areas of MIR, where machine learning models virtually superseded more traditional approaches. We can only conjecture on the reasons for this absence, however, it seems to us that sequence alignment methods faithfully model a variety of alignment problems and the established methods’ correspondingly high performance leaves little room for improvement. However, this observation does not hold for the question which sequence representations are to be processed by alignment algorithms, a question that is not settled, neither in the symbolic nor in the audio domain. Feature representations and local metrics abound, and have myriad downstream implications for alignment success or failure which are often hard to predict.

Our approach excels at this point by producing learned representations which leverage non-local information. The representations play well with DTW-based post-

processing, however, end-to-end note matching remains challenging. Learning representations shifts the problem of edge cases from the modeling stage (or even post-hoc engineering) towards the training data. Augmenting data with complex mismatches in combination with a model that effectively predicts matches frees us from having to address all possible cases explicitly. Randomized training mismatches enable the model to learn robust representation for a variety of mismatching sequences. In practice, this also leads to greater flexibility, as no quantized music, score annotations, or any attributes beyond the basic MIDI notes are required.

Many extensions of our approach are possible. The hyperparameter and architectural space of plausible representation models open several possibilities for future research. Furthermore, the data used to train and test the model is specific: solo piano pieces and performances of common practice period music. This is due to the fact that reference models work on the piano data, and large symbolic piano datasets are available. Note that this data presents one of the most challenging note alignment scenarios and we expect our core ideas to translate to other symbolic data — after retraining. An open question is whether this type of token-based match representation learning can be used in audio or multimodal domains, e.g. by applying it to discrete audio encodings. Lastly, the representation learning backbone is trained without any information about the DTW post-processing. SoftDTW [28] approaches appear promising to bridge this gap while keeping sensible alignment constraints in an end-to-end model. However, we want to stress again that the monotonicity condition of (soft)DTW does not strictly hold in symbolic music even though it has proven an effective heuristic. Many questions remain open, yet we hope to have shown that representation learning can be integrated successfully and beneficially into note alignment methods.

7. REPRODUCIBILITY

Code and pre-trained checkpoints and public data available at: <https://github.com/sildater/theglueNote>

8. ACKNOWLEDGMENTS

This work receives funding from the European Research Council (ERC), under the European Union’s Horizon 2020 research and innovation programme, grant agreement No. 101019375 (*Whither Music?*). The LIT AI Lab is supported by the Federal State of Upper Austria.

9. REFERENCES

- [1] B. Gingras and S. McAdams, “Improved Score-Performance Matching using Both Structural and Temporal Information from MIDI Recordings,” *Journal of New Music Research*, vol. 40, no. 1, pp. 43–57, 2011.
- [2] C.-T. Chen, J.-S. R. Jang, and W. Liou, “Improved Score-Performance Alignment Algorithms on Polyphonic Music,” in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 1365–1369.
- [3] E. Nakamura, N. Ono, Y. Saito, and S. Sagayama, “Merged-Output Hidden Markov Model for Score Following of MIDI Performance with Ornaments, Desynchronized Voices, Repeats and Skips,” in *International Conference on Mathematics and Computing*, 2014.
- [4] E. Nakamura, N. Ono, S. Sagayama, and K. Watanabe, “A Stochastic Temporal Model of Polyphonic MIDI Performance with Ornaments,” *Journal of New Music Research*, vol. 44, no. 4, pp. 287–304, 2015.
- [5] E. Nakamura, T. Nakamura, Y. Saito, N. Ono, and S. Sagayama, “Outer-product hidden markov model and polyphonic midi score following,” *Journal of New Music Research*, vol. 43, no. 2, pp. 183–201, 2014.
- [6] E. Nakamura, K. Yoshii, and H. Katayose, “Performance Error Detection and Post-Processing for Fast and Accurate Symbolic Music Alignment,” in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2017, pp. 347–353.
- [7] E. Nakamura, P. Cuvillier, A. Cont, N. Ono, and S. Sagayama, “Autoregressive hidden semi-markov model of symbolic music performance for score following,” in *16th International Society for Music Information Retrieval Conference (ISMIR)*, 2015.
- [8] S. D. Peter, C. E. Cancino-Chacón, F. Foscarin, A. P. McLeod, F. Henkel, E. Karystinaios, and G. Widmer, “Automatic note-level score-to-performance alignments in the asap dataset,” *Transactions of the International Society for Music Information Retrieval*, Jun 2023.
- [9] S. D. Peter, “Online symbolic music alignment with offline reinforcement learning,” in *Proceedings of the 24th International Society for Music Information Retrieval Conference (ISMIR)*, 2023.
- [10] R. B. Dannenberg, “An On-Line Algorithm for Real-Time Accompaniment,” in *Proceedings of the International Computer Music Conference (ICMC)*, vol. 84, 1984, pp. 193–198.
- [11] B. Vercoe, “The Synthetic Performer in the Context of Live Performance,” in *Proceedings of the International Computer Music Conference (ICMC)*, 1984, pp. 199–200.
- [12] C. Cancino-Chacón, S. Peter, P. Hu, E. Karystinaios, F. Henkel, F. Foscarin, N. Varga, and G. Widmer, “The accompanion: Combining reactivity, robustness, and musical expressivity in an automatic piano accompanist,” in *International Joint Conference on Artificial Intelligence (IJCAI)*, 2023.
- [13] C. Raphael and Y. Gu, “Orchestral Accompaniment for a Reproducing Piano,” in *International Conference on Mathematics and Computing*, 2009.
- [14] A. Arzt, “Flexible and robust music tracking,” Ph.D. dissertation, Johannes Kepler University Linz, Linz, Austria, 2016.
- [15] M. Müller, *Fundamentals of Music Processing – Audio, Analysis, Algorithms, Applications*. Springer, 2015.
- [16] M. Müller, Y. Özer, M. Krause, T. Prätzlich, and J. Driedger, “Sync toolbox: A python package for efficient, robust, and accurate music synchronization,” *Journal of Open Source Software*, p. 3434, 2021.
- [17] M. Dorfer, F. Henkel, and G. Widmer, “Learning to Listen, Read, and Follow: Score Following as a Reinforcement Learning Game,” in *Proceedings of the 19th International Society for Music Information Retrieval Conference*. Paris, France: ISMIR, 2018, pp. 784–791.
- [18] F. Henkel, S. Balke, M. Dorfer, and G. Widmer, “Score following as a multi-modal reinforcement learning problem,” *Transactions of the International Society for Music Information Retrieval*, Nov 2019.
- [19] S. Xu, S. Chen, R. Xu, C. Wang, P. Lu, and L. Guo, “Local feature matching using deep learning: A survey,” *arXiv preprint arXiv:2401.17592*, 2024.
- [20] P. Lindenberger, P.-E. Sarlin, and M. Pollefeys, “Lightglue: Local feature matching at light speed,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 17 627–17 638.
- [21] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, “Superglue: Learning feature matching with graph neural networks,” in *Proceedings of the*

IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 4938–4947.

- [22] G. Hadjeres and L. Crestel, “The piano inpainting application,” *arXiv preprint arXiv:2107.05944*, 2021.
- [23] N. Fradet, J.-P. Briot, F. Chhel, A. El Fallah Seghrouchni, and N. Gutowski, “MidiTok: A python package for MIDI file tokenization,” in *Extended Abstracts for the Late-Breaking Demo Session of the 22nd International Society for Music Information Retrieval Conference*, 2021.
- [24] W. Goebel. (1999) The Vienna 4x22 Piano Corpus. [Online]. Available: http://repo.mdw.ac.at/projects/IWK/the_vienna_4x22_piano_corpus/index.html
- [25] P. Hu and G. Widmer, “The Batik-plays-Mozart Corpus: Linking Performance to Score to Musicological Annotations,” in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2023.
- [26] C. E. Cancino-Chacón, T. Gadermaier, G. Widmer, and M. Grachten, “An Evaluation of Linear and Non-linear Models of Expressive Dynamics in Classical Piano and Symphonic Music,” *Machine Learning*, vol. 106, no. 6, pp. 887–909, 2017.
- [27] S. Flossmann, W. Goebel, M. Grachten, B. Niedermayer, and G. Widmer, “The Magaloff Project: An Interim Report,” *Journal of New Music Research*, vol. 39, no. 4, pp. 363–377, 2010.
- [28] M. Cuturi and M. Blondel, “Soft-dtw: a differentiable loss function for time-series,” in *International conference on machine learning*. ICML, 2017, pp. 894–903.

GAPS: A LARGE AND DIVERSE CLASSICAL GUITAR DATASET AND BENCHMARK TRANSCRIPTION MODEL

Xavier Riley Zixun Guo Drew Edwards Simon Dixon

Centre for Digital Music, Queen Mary University of London

j.x.riley@qmul.ac.uk, s.e.dixon@qmul.ac.uk

ABSTRACT

We introduce GAPS (Guitar-Aligned Performance Scores), a new dataset of classical guitar performances, and a benchmark guitar transcription model that achieves state-of-the-art performance on GuitarSet in both supervised and zero-shot settings. GAPS is the largest dataset of real guitar audio, containing 14 hours of freely available audio-score aligned pairs, recorded in diverse conditions by over 200 performers, together with high-resolution note-level MIDI alignments and performance videos. These enable us to train a state-of-the-art model for automatic transcription of solo guitar recordings which can generalise well to real world audio that is unseen during training.

For each track in the dataset, we provide metadata of the composer and performer, giving dates, nationality, gender and links to IMSLP or Wikipedia. We also analyse guitar-specific features of the dataset, such as the distribution of fret-string combinations and alternate tunings. This dataset has applications to various MIR tasks, including automatic music transcription, score following, performance analysis, generative music modelling and the study of expressive performance timing.

1. INTRODUCTION

Automatic Music Transcription (AMT) for instruments other than piano has faced challenges due to a lack of high-quality datasets [1]. This gap has limited the development of accurate transcription systems compared to those available for the piano, which benefit from comprehensive datasets like MAESTRO [2] and MAPS [3]. However, recent developments in audio-score alignment methods have shown promising results in improving transcription accuracy [1, 4].

With 2.7 million guitars sold in the US alone in 2019¹, the guitar is a popular instrument and retains a widespread cultural significance. Around 6% of these guitars sold were

of the classical or flamenco types (roughly 162,000 units). For comparison, around 31,000 acoustic pianos were sold in the US that year. Despite this popularity, we believe that the study of the guitar in the field of Music Information Retrieval (MIR) is underrepresented. Reviewing the paper titles for ISMIR conferences from 2013-2023 we find that publications with the word “piano” in the title outnumber those with “guitar” by 3 to 1². This imbalance may be due to the availability of high quality datasets for piano; new datasets and methods for guitar will help to address this.

In this paper, we present GAPS, a large and diverse classical guitar dataset that contains 14 hours of matched nylon string guitar audio recordings, note-level MIDI annotations, and corresponding music scores, where the recordings feature over 200 performers in diverse recording conditions. This is several times larger than GuitarSet [5], the EGDB dataset [6], the FrançoisLeduc dataset [4] and the IDMT-SMT-Guitar dataset [7] (see Section 2 for a detailed comparison). We use this data to train a benchmark transcription model which achieves state-of-the-art results for solo guitar transcription across 4 dataset splits.


The contributions of this paper are as follows:

- the largest available dataset consisting of real guitar audio, performance video, corresponding music scores and aligned MIDI annotations;
- metadata and external links for composers and performers, plus statistics of guitar-specific features;
- an efficient pipeline for verifying alignments of scores to audio;
- a benchmark state-of-the-art guitar transcription model trained on our dataset; and
- analysis and discussion of the effects of dataset quality, quantity and variety on AMT performance.

2. RELATED WORK

GuitarSet [5] is the most widely used MIR dataset for guitar. It provides around 3 hours of annotated guitar performances, where the data collection process required the use of a specialised guitar fitted with a hexaphonic pickup which was able to capture the output of individual strings. The use of a single guitar severely limits the diversity of

¹ <https://www.musictrades.com/us-retail-sales-guitar-market.html>

 © X. Riley, Z. Guo, D. Edwards and S. Dixon. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** X. Riley, Z. Guo, D. Edwards and S. Dixon, “GAPS: A Large and Diverse Classical Guitar Dataset and Benchmark Transcription Model”, in *Proc. of the 25th Int. Society for Music Information Retrieval Conf.*, San Francisco, United States, 2024.

² 46 piano and 15 guitar

Name	Audio type	Track count	Duration (m)	Note count	Scores
GuitarSet [5]	Real	360	180	62,476	No
IDMT-SMT-Guitar [7]	Real	1173	340	*5,767	No
EGDB [6]	Real	240	118	35,700	No
FrançoisLeduc [4]	Real	79	240	75,312	Yes (commercial)
GAPS (ours)	Real	300	843	259,410	Yes
SynthTab [8]	Synthetic	20,715	786,774	-	Yes, via DadaGP

Table 1. Comparison of existing guitar datasets, split into real and synthetic sources. * For IDMT, the note count is shown only for notes with annotations available.

timbres and recording conditions, and in turn makes it harder for AMT models to generalise from this data [4].

The EGDB [6] dataset contains 2 hours of guitar audio recorded by a professional guitarist using a hexaphonic pickup and recorded via DI (direct input). The DI signal is then further rendered using 6 different amplifier emulation plugins. The onsets and offsets of each note are annotated.

The IDMT-SMT-GUITAR database [7] is recorded by 3 musicians using 6 different guitars (5 electric, 1 acoustic). The final audio is either obtained from DI or microphone output. It contains 4 subsets each targeting a different MIR task, ranging from single notes to chords to various short musical pieces. Its utility in transcription tasks is limited however, as only a subset of the audio has corresponding time-aligned note annotations.

Improvements in diversity of audio sources were achieved by Maman and Bermano [1] through the use of score alignment techniques. Digital scores (in MIDI format) were aligned to the activations of the Onsets and Frames transcription model [9] trained on synthetic data. Low quality alignments were discarded and the remaining data was used to fine tune the model further. This expectation maximisation approach yielded a new state-of-the-art result on GuitarSet in the zero-shot setting, which demonstrated a generalisable model. The authors collected 5 hours of classical guitar recordings and scores in this work but these were not released as part of the publication.

Building on this approach, Riley et al. [4] published a new state-of-the-art model for guitar transcription. Instead of the Onsets and Frames model, they use the high resolution piano transcription model by Kong et al. [10], which was shown to be more tolerant of misaligned labels. Furthermore, instead of bootstrapping the process with synthetic data, they employ a pre-training step where a model is trained on the MAESTRO dataset with data augmentation, which was shown to improve generalisation. A dataset of around 4 hours of audio-MIDI pairs was published with their work, however the scores are not freely available as they were purchased from a commercial source.

As an alternative to annotating real world audio, Zang et al. [8] recently proposed a large scale dataset of synthesised audio from a subset of the DadaGP dataset [11]. When used as a pre-training step, the authors note improvements in multi-pitch estimation over 3 guitar datasets. De-

spite the large volume of additional training data, their note level results on the GuitarSet test split (86.1% F1 no off-set) are lower than those of several other methods which use GuitarSet alone (see [4]). This suggests that synthetic data alone is not sufficient to improve AMT systems, but a full comparison with consistent use of model architectures would be needed to establish this with certainty.

3. OVERVIEW OF DATASET

3.1 Dataset Curation

In an effort to improve the amount of available labelled, non-synthetic data, we have curated a new dataset of classical guitar recordings based on freely available scores from the ClassClef website³, together with matching performances on YouTube⁴. We align these sources using the automatic process described in [4] and then manually verified each alignment using the synchronised score viewer at soundslice.com. Following another alignment stage, any remaining scores with inaccurate alignments are rejected (using the criteria described below). This resulted in 300 performances sampled from the entire classical guitar canon totalling over 14 hours of music and over 250,000 note events. We have also curated extensive metadata, including information about the pieces, composers and performers, in order to enrich the dataset with details of the cultural context.

Our curation process is shown in Figure 1. It begins with the ClassClef website which provides around 5,500 pieces for download in PDF and GuitarPro formats. These focus mainly on the classical guitar with some flamenco and fingerstyle pieces included. Additionally, 547 of the pieces include links to videos on YouTube of a performance of the same piece. We first collected all GuitarPro files and converted them to MusicXML and MIDI formats using the free MuseScore software package⁵. We also downloaded the audio and video for the 547 pieces where YouTube links were available.

Using the alignment method described in [4], we produce an initial alignment between the score and the recording for each piece. This proceeds in two stages: an initial

³ classclef.com

⁴ youtube.com

⁵ musescore.org/en

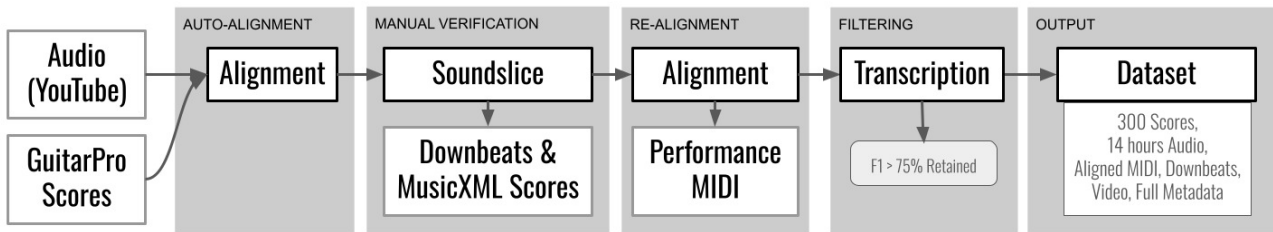


Figure 1. Flowchart of the dataset creation process.

alignment via Dynamic Time Warping (DTW), and a further fine alignment stage in which the notes of each chord are aligned to their closest activation from an existing transcription model. We emphasise this point as the resulting alignments are fully polyphonic in nature and as a result are more accurate than those produced by DTW alone, as described in [4].

In some cases the automatic alignment will not succeed, for example, when a linked video contains audio for an entire suite but the score only contains a single movement. For this reason a manual verification step was required. Using the `soundslice.com` website, we upload the automatically aligned downbeats to synchronize playback between the audio and the score. This allowed the authors of the paper (each with over 10 years of music experience) to review 474 of the scores (chosen at random) in an efficient workflow. More specifically, we manually verified the alignment between each downbeat location and the score for all 474 pieces. Particular attention was paid to the beginning and ending of each piece as these were a frequent source of issues in the DTW process. Moreover, any differences between the score and the performance that were identified were corrected, if feasible. In the end, 74 pieces were rejected for various reasons – for example those containing 7-string guitars, guitar duets and pieces where the edition did not match the performance. Out of the remaining 400 pieces examined, 280 were usable without corrections to the score and the remaining 120 required intervention to obtain correct downbeat alignments.

The 400 reviewed scores were then re-aligned using the same alignment method from step two of figure 1. The corrected downbeats were used as anchor points during this alignment stage to ensure that any alignment errors would be localised to one measure of music. To validate accuracy, we then compared our aligned versions of the score to outputs of the guitar transcription model from [4]. We retained the 300 scores with the highest agreement, measured using the “F-measure no offset” metric from the `mir_eval` library [12]. We retained scores which had an F-measure of more than 75%, yielding 300 audio-score pairs. We manually reviewed the lower scoring alignments and found a number of issues including errors with the processing of anacrusis bars, non-440Hz tunings and discrepancies between the performance and score editions. We hope to address these where possible as part of future work.

A summary of existing guitar datasets is shown in Table 1. When considering datasets with real (as opposed to syn-

thesised) audio, GAPS represents a significant advance in terms of the duration of audio and number of note events. In addition, ours is the first dataset of real audio to include freely available full music scores, tablatures in MusicXML format, and accompanying performance videos.

3.2 Composers

Works from 93 different composers are included, ranging from the Renaissance (Luys Milan, c.1500-1561) to the present day. The majority of works are from the classical guitar repertoire, with a small number of flamenco pieces and arrangements of popular music. We include the dates, nationality and presumed gender of each composer with links to canonical URLs (IMSLP and Wikipedia) where possible.

Examining the diversity of composers contained in the dataset, Figure 2 shows their nationalities, according to data from the canonical URL for each composer. This shows they are broadly divided between Europe and Latin America. In terms of chronology Figure 3 shows the distribution of pieces according to the year in which the composer was born. This shows that the included pieces are mainly weighted around the Romantic era (1850-1900). The peak around 1650 is almost entirely due to J.S. Bach, who is the second most common composer in our dataset with 23 pieces. We also include information about the presumed gender of composers in our metadata, however only two female composers (Maria Linnemann and Luise Walker) are included who together represent 2% of the total by piece count. We acknowledge that this is a shortcoming of the current dataset and we will seek to address this in future work.

3.3 Performances

The accompanying videos are drawn from 205 different performers with YouTube views totalling over 35 million across all videos. Some are professionally produced recordings whereas others are recorded on commodity equipment such as phones and laptops. We believe this is an advantage of this dataset in that recordings are drawn from a wide variety of real world recording conditions, which in turn helps to increase the robustness of trained AMT models.

In the metadata we include information about the name of the performer (where available), their social media links (if available), the YouTube channel, the view count and

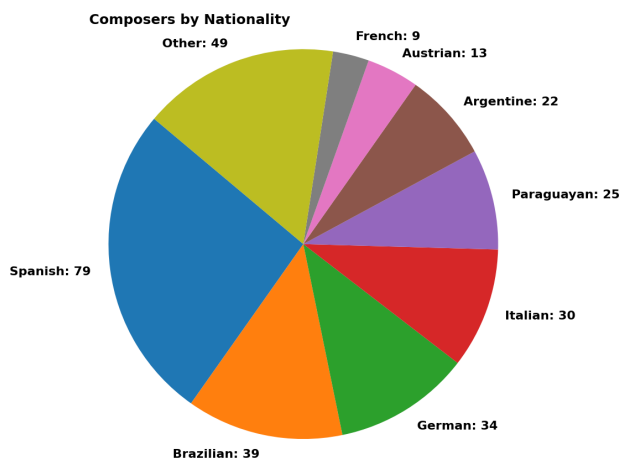


Figure 2. Nationalities of the composers

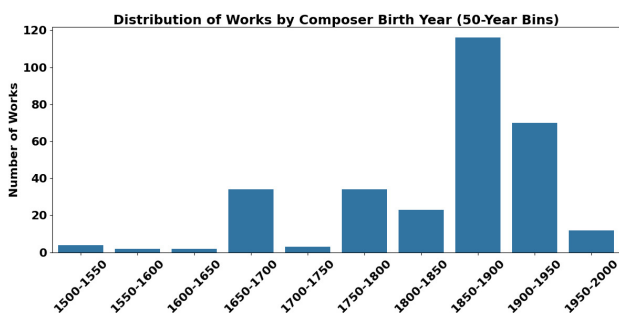


Figure 3. Histogram of works according to composer’s birth year at 50 year intervals

Tuning	Count	% of total
EADGBE	232	77.33
DADGBE	58	19.33
DGDGBE	5	1.67
EADF#BE	2	0.67
FADGBE	1	0.33
CGDGBE	1	0.33
EBDGBE	1	0.33

Table 2. Distribution of guitar tunings in GAPS. The tuning is expressed from low to high pitch.

the presumed gender of the performer. This was gathered to examine the extent to which classical guitar is a male dominated field. We find that female performers are better represented than composers in our dataset, but still only comprise 23% of the total.

3.4 Guitar-Specific Features

The large number of scores allows us to examine several guitar-specific features of the data. In Table 2 we see that two different tunings account for 97% of the data. While standard tuning is most common, almost 20% of pieces have the lowest string tuned down one tone to D. Other alternate tunings account for around 3.3% of the total.

To see the distribution of notes across the guitar neck in

this dataset, we have plotted a heat map as shown in Figure 4 using the fret information contained in the MusicXML tablature. Over the 259,000 note events we see that the pieces in the classical guitar repertoire favour the use of open strings and the first position. The strong peak at the 2nd fret A on the G string also suggests a preference towards “guitar friendly” keys such as E and A which allow the performer to use the open bass and top strings. While this distribution is uneven, we consider this to be representative of the classical guitar repertoire. We encourage other dataset authors to explore similar visualisations in future work to see if this varies with other genres.

Since most pitches can be played on more than one position on the guitar, there is an exponentially large number of tablatures that correspond to any one given score, including many physically unplayable versions. While each tablature in our dataset represents one valid way to play the score, we have not verified the extent to which the tablatures correspond to the choices of the performers in the specific performances in the GAPS dataset. This is left for future work. As we were not able to trace the provenance of the ClassClef data, we presume the data is crowdsourced and reflects the playing habits of a subset of computer-literate guitarists. It is also possible that some of the tabs were generated algorithmically from the score data.

4. TRANSCRIPTION BASELINE

4.1 Experimental Settings

To demonstrate the utility of the GAPS dataset of aligned score-audio pairs, we trained several guitar transcription models using the high resolution model of Kong et al. [10], which achieved state-of-the-art performance when trained for guitar transcription [4]. This model is a convolutional recurrent neural network (CRNN) that is trained in a supervised manner to map log mel-spectrograms of 10-second segments of audio to MIDI. The convolutional layers span only across the frequency dimension, maintaining the time-resolution of the original spectrogram (10ms). These features are then processed by a gated recurrent unit (GRU) to produce the final outputs of onset, offset, frame activity, and velocity activations per pitch per time window.

There are two reasons why we used the high resolution model [10]. Firstly to ensure fair comparisons with the state-of-the-art model in [4] as it shares the same architecture. This allows us to examine how our GAPS dataset influences the same transcription model. Secondly, fine-tuning becomes feasible due to the shared architecture among multiple piano transcription models [10, 13]. This allows us to investigate whether different pre-trained piano transcription models can improve guitar transcription through domain adaptation.

For our experiments, we trained 2 sets of models. The first set of models is trained only on the GAPS dataset and the second set of models is trained with a combination of GuitarSet and GAPS. We employ the first set of models for zero-shot inference on the complete GuitarSet, while the

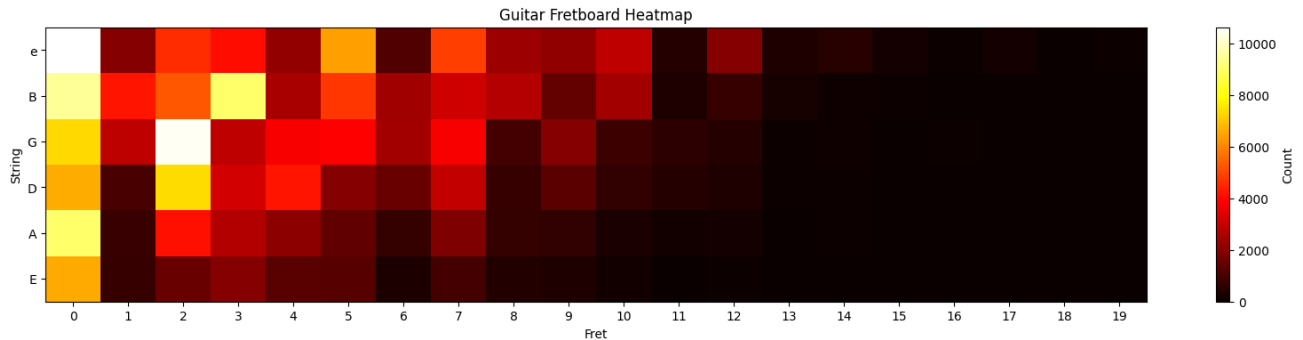


Figure 4. Heat map of the fret/string combinations in the GAPS MusicXML tablatures.

second set is utilised to evaluate guitar transcription performance across the test splits of GuitarSet, theFrançoisLeduc dataset and GAPS. To study the effects of pre-training and finetuning [8, 13], each set of models has 3 variants: one trained from scratch and two finetuned from one of two published checkpoints for piano transcription [10, 13]. This also allows for a more direct comparison with results reported in [4].

Regarding our training data and strategy, we randomly divide the GAPS dataset with a 90:10 split by piece, for training and testing respectively. Following [4], each audio file is split into 10-second chunks, using a hop size of 1 second. We adopt the same train-test split from [4, 14] for GuitarSet. During training, pitch shifting of up to ± 3 semitones was randomly applied as data augmentation [14].

4.2 Transcription Results

In Tables 3 to 6, we report the evaluation results for the models described in Section 4.1. Our proposed combination of model, pre-training checkpoint and dataset achieves state-of-the-art performances on all 4 test sets mentioned in Section 4.1. Considering the similarities to the approach used by Riley et al. [4], our larger dataset appears to drive the improvement in results.

4.2.1 Generalisation and Guitar Types

GuitarSet contains audio for one acoustic steel string guitar recorded via microphone and also via the guitar pickup (the “DI” outputs). Despite our GAPS data containing only performances on nylon-stringed classical guitars, our model is able to generalise well to GuitarSet in the zero-shot setting (F-measure 88.1% - see Table 4). This result is interesting as it appears that timbral differences between guitars are not a strong factor in the success of the model for this task. On the other hand, GAPS does include a large range of guitars and recording conditions (unlike GuitarSet’s one guitar), which we expect would contribute to the generalisation performance of models trained on it.

We also note that for the other solutions based on encoder-decoder architectures [14, 16], the strong results in the supervised setting on GuitarSet fail to perform as well on unseen data. Table 4 shows the transcription accuracy on GuitarSet in the zero-shot setting, i.e. where models are trained without any access to GuitarSet. F-measure scores

for MT3 fall from 90.0% to 32.0% on GuitarSet. The previous state-of-the-art model (Time-Frequency Perceiver) from Lu et al. [14] attains 91.1% in the GuitarSet supervised task but drops to 80.0% on the unseen FrançoisLeduc test set. It may be the case that these architectures require more data to generalise effectively and we hope to explore training them on GAPS in future work.

For the FrançoisLeduc test split in Table 5, our proposed model outperforms Riley et al. [4] by a small margin, however their model was trained in a supervised fashion whereas this dataset was unseen by our model.

Conversely, our proposed method outperforms Riley et al. [4] on the GAPS test split by a margin of 2.2% (see Table 6). This indicates that, despite our method’s strong generalisation (see Table 4), it is somewhat specialised to classical guitar timbres and that the strongest results in the future may rely on the use of specific training data.

	P_{50}	R_{50}	F_{50}
Basic Pitch [15]	-	-	79.0
MT3 [16]	-	-	90.0
Zang et al [8]	-	-	84.5
Lu et al. [14]	-	-	91.1
SpecTNT (in [14])	-	-	90.7
Riley et al. [4] ($_{FL}$)	87.6	86.8	86.9
Riley et al. ($_{GS+FL}$)	91.1	88.5	89.7
Ours			
($_{GAPS}$)	89.9	85.4	87.2
($_{GAPS}$ Finetuned from [10])	88.8	86.8	87.5
($_{GAPS}$ Finetuned from [13])	90.1	86.6	88.0
($_{GAPS+GS}$)	90.2	90.9	90.4
($_{GAPS+GS}$ Finetuned from [10])	89.4	92.1	90.7
($_{GAPS+GS}$ Finetuned from [13])	91.3	90.7	91.2

Table 3. Results for note-level transcription accuracy on the GuitarSet test split. P_{50} , R_{50} , and F_{50} are Precision, Recall and F1-measure, expressed as percentages, at 50ms resolution. All are evaluated on onsets only (no offsets or velocity), using the `mir_eval` library. Baseline results are described in [4].

	P_{50}	R_{50}	F_{50}
MT3 [16]	-	-	32.0
Kong et al. [10]	67.5	49.7	54.8
Kong et al. (w/ aug)	80.6	44.0	50.3
Zang et al. [8] (Synthtab)	-	-	70.2
Maman (MusicNet _{EM}) [1]	86.6	80.4	82.9
Maman (Guitar) [1]	86.7	79.7	82.2
Riley et al. [4]	88.0	87.1	87.3
Ours	92.4	81.8	86.1
Ours (Finetuned from [10])	91.6	83.7	87.0
Ours (Finetuned from [13])	91.1	85.9	88.1

Table 4. Results for note-level transcription accuracy on the entire GuitarSet in the zero-shot setting.

	P_{50}	R_{50}	F_{50}
Basic Pitch [15]	54.6	85.0	66.1
Omnizart [17]	63.0	72.1	67.1
MT3 [16]	48.8	57.0	52.4
Lu et al. [14]	83.6	77.3	80.0
Riley et al. [4]	83.9	85.5	84.7
Ours (Finetuned from [13])	85.5	84.2	84.8

Table 5. Results for note-level transcription accuracy on the test split of the FrançoisLeduc dataset [4].

	P_{50}	R_{50}	F_{50}
Riley et al. [4]	92.9	91.4	92.1
Ours	94.9	92.1	93.4
Ours (Finetuned from [10])	94.6	93.4	94.0
Ours (Finetuned from [13])	95.0	93.6	94.3

Table 6. Results for note-level transcription accuracy on a test split of the GAPS dataset.

4.2.2 Effects of Pre-training

In each of our evaluations, we see a consistent trend whereby the model with no pre-training is surpassed by the model pre-trained on piano (MAESTRO) and fine-tuned on GAPS, which in turn is surpassed by the model pre-trained on an augmented version of MAESTRO [13] before fine-tuning on GAPS. This illustrates the importance of pre-training and fine-tuning, as well as data augmentation as important drivers of success in the transcription task (see Edwards et al. [13] for a detailed analysis of the effect of data augmentation on transcription generalisation).

We also note that strong results for other methods on the GuitarSet test split are obtained from models trained with a mixture of datasets [4, 14, 16]. One exception is Zang et al. [8], who use a large corpus of synthetically rendered guitar samples for pre-training. This does not perform as well as other methods but their results were obtained from a model (TabCNN) designed for guitar tablature prediction, as opposed to a state-of-the-art transcription model. A full comparison of synthetic and real audio for pre-training is something we also hope to explore in future work.

5. CONCLUSION

We present GAPS, a large dataset of score and audio pairs for solo classical guitar which comprises a wide range of composers, performers and real-world recording conditions, totaling 14 hours of recordings. The MIDI annotations are made freely available and the audio is available at the YouTube links provided. This represents the largest dataset of freely available guitar audio-score pairs to date.

We included analysis of the overall statistics of the GAPS dataset, but further musicological work could be done to examine connections between the composers, performers and musical features. The published MIDI annotations could be useful for generative modelling of classical guitar and other instruments. For future work we will look to expand the dataset and enhance the diversity where possible, particularly for the range of composers we include.

One application of this dataset is AMT for guitar, which we demonstrate through a comprehensive evaluation of a transcription model trained on our data. This shows state-of-the-art results when compared with existing methods trained on other datasets. In future work we look to examine further issues around pre-training for guitar transcription.

6. ETHICS STATEMENT

In addition to our role as researchers, we are also members of the global community of musicians and we seek to respect their important role in our culture. Our work here raises several issues which may have wider impact on this community which we hope to address as follows.

Firstly, we believe that using sources which are publicly available (subject to licence conditions) is important to reduce barriers to future research. At the time of writing, neither the scores nor their audio recordings are behind any kind of paywall. We have processed this data and make the results available on the basis of fostering research. We also obtained permission from the website owner of `classclef.com` to make use of their materials.

By publishing work on YouTube, artists do grant some kind of implicit licence that the data can be viewed, however the specific terms of the licence may restrict further use cases. We believe that our work is justified in using this data under fair use or fair dealing exemptions defined for research, but we are mindful that further use of the data may require express permission from the performers, composers or copyright-holders. We have attempted to address this by including detailed information about all performers and composers in the accompanying metadata to allow interested parties to contact them directly.

Finally we recognise that AMT models which approach human-level accuracy might pose a threat to those who are employed in music transcription and related fields. On the other hand, such models could also assist such work and become tools for improving the efficiency and accuracy of their daily work. For this reason we are carefully considering whether to make our model weights freely available.

7. ACKNOWLEDGMENTS

Authors XR, ZG and DE are research students at the UKRI Centre for Doctoral Training in Artificial Intelligence and Music, supported by UK Research and Innovation [grant number EP/S022694/1] and Yamaha Corporation (DE).

8. REFERENCES

- [1] B. Maman and A. H. Bermann, “Unaligned supervision for automatic music transcription in the wild,” in *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, ser. Proceedings of Machine Learning Research, vol. 162. PMLR, 2022, pp. 14 918–14 934.
- [2] C. Hawthorne, A. Stasyuk, A. Roberts, I. Simon, C. A. Huang, S. Dieleman, E. Elsen, J. H. Engel, and D. Eck, “Enabling factorized piano music modeling and generation with the MAESTRO dataset,” in *7th International Conference on Learning Representations*, New Orleans, USA, 2019.
- [3] V. Emiya, N. Bertin, B. David, and R. Badeau, “MAPS - a piano database for multipitch estimation and automatic transcription of music,” INRIA, France, Research Report 00544155, 2010. [Online]. Available: <https://hal.inria.fr/inria-00544155>
- [4] X. Riley, D. Edwards, and S. Dixon, “High resolution guitar transcription via domain adaptation,” in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2024, Seoul, 2024*. IEEE, 2024. [Online]. Available: <https://arxiv.org/abs/2402.15258>
- [5] Q. Xi, R. M. Bittner, J. Pauwels, X. Ye, and J. P. Bello, “GuitarSet: A dataset for guitar transcription,” in *Proceedings of the 19th International Society for Music Information Retrieval Conference*, Paris, France, 2018, pp. 453–460.
- [6] Y. Chen, W. Hsiao, T. Hsieh, J. R. Jang, and Y. Yang, “Towards automatic transcription of polyphonic electric guitar music: A new dataset and a multi-loss transformer model,” in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2022, Virtual and Singapore, 23-27 May 2022*. IEEE, 2022, pp. 786–790.
- [7] C. Kehling, J. Abeßer, C. Dittmar, and G. Schuller, “Automatic tablature transcription of electric guitar recordings by estimation of score- and instrument-related parameters,” in *Proceedings of the 17th International Conference on Digital Audio Effects, DAFX-14, Erlangen, Germany, September 1-5, 2014*, 2014, pp. 219–226.
- [8] Y. Zang, Y. Zhong, F. Cwitkowitz, and Z. Duan, “Synthtab: Leveraging synthesized data for guitar tablature transcription,” in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2024, Seoul, 2024*. IEEE, 2024. [Online]. Available: <https://arxiv.org/abs/2402.15258>
- [9] C. Hawthorne, E. Elsen, J. Song, A. Roberts, I. Simon, C. Raffel, J. H. Engel, S. Oore, and D. Eck, “Onsets and frames: Dual-objective piano transcription,” in *Proceedings of the 19th International Society for Music Information Retrieval Conference*, Paris, France, 2018, pp. 50–57.
- [10] Q. Kong, B. Li, X. Song, Y. Wan, and Y. Wang, “High-resolution piano transcription with pedals by regressing onset and offset times,” *IEEE ACM Transactions on Audio, Speech and Language Processing*, vol. 29, pp. 3707–3717, 2021.
- [11] P. Sarmento, A. Kumar, C. J. Carr, Z. Zukowski, M. Barthelet, and Y. Yang, “Dadagp: A dataset of tokenized guitarpro songs for sequence models,” in *Proceedings of the 22nd International Society for Music Information Retrieval Conference, ISMIR 2021, Online, November 7-12, 2021*, 2021, pp. 610–617.
- [12] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, and D. P. W. Ellis, “Mir_eval: A transparent implementation of common MIR metrics,” in *Proceedings of the 15th International Society for Music Information Retrieval Conference*, Taipei, Taiwan, 2014, pp. 367–372.
- [13] D. Edwards, S. Dixon, E. Benetos, A. Maezawa, and Y. Kusaka, “A data-driven analysis of robust automatic piano transcription,” *IEEE Signal Process. Lett.*, vol. 31, pp. 681–685, 2024.
- [14] W. T. Lu, J. Wang, and Y. Hung, “Multitrack music transcription with a time-frequency perceiver,” in *IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP 2023, Rhodes Island, Greece, June 4-10, 2023*. IEEE, 2023, pp. 1–5.
- [15] R. M. Bittner, J. J. Bosch, D. Rubinstein, G. Meseguer-Brocal, and S. Ewert, “A lightweight instrument-agnostic model for polyphonic note transcription and multipitch estimation,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Singapore, 2022.
- [16] J. Gardner, I. Simon, E. Manilow, C. Hawthorne, and J. H. Engel, “MT3: Multi-task multitrack music transcription,” in *Tenth International Conference on Learning Representations*, 2022.
- [17] Y. Wu, Y. Luo, T. Chen, I. Wei, J. Hsu, Y. Chuang, and L. Su, “Omnizart: A general toolbox for automatic music transcription,” *J. Open Source Softw.*, vol. 6, no. 68, p. 3391, 2021. [Online]. Available: <https://doi.org/10.21105/joss.03391>

A KALMAN FILTER MODEL FOR SYNCHRONIZATION IN MUSICAL ENSEMBLES

Hugo T. Carvalho^{1*} Min S. Li² Massimiliano di Luca³ Alan M. Wing³

¹ Department of Statistical Methods, Federal University of Rio de Janeiro, Brazil

² Bristol Interaction Group, School of Computer Science, University of Bristol, United Kingdom

³ Virtual Reality Lab, School of Psychology, University of Birmingham, United Kingdom

* Corresponding author: hugo@dme.ufrj.br

ABSTRACT

The synchronization of motor responses to rhythmic auditory cues is a fundamental biological phenomenon observed across various species. While the importance of temporal alignment varies across different contexts, achieving precise temporal synchronization is a prominent goal in musical performances. Musicians often incorporate expressive timing variations, which require precise control over timing and synchronization, particularly in ensemble performance. This is crucial because both deliberate expressive nuances and accidental timing deviations can affect the overall timing of a performance. This discussion prompts the question of how musicians adjust their temporal dynamics to achieve synchronization within an ensemble. This paper introduces a novel feedback correction model based on the Kalman Filter, aimed at improving the understanding of interpersonal timing in ensemble music performances. The proposed model performs similarly to other linear correction models in the literature, with the advantage of low computational cost and good performance even in scenarios where the underlying tempo varies.

1. INTRODUCTION

Synchronization of motor responses to rhythmic auditory cues represents a biological phenomenon found across various species [1], and social collectives often engage in activities necessitating precise temporal coordination among members, a crucial factor for successful group endeavors. For example, in scenarios such as rowing eights, temporal alignment may not be the primary focus, but individual timing remains tied to collective timing dynamics [2]. In domains like musical performances, achieving precise temporal synchronization serves as a prominent goal [3].

Typically, musicians do not adhere strictly to the exact timing of note onsets as indicated in the musical score: due to expressiveness, they often introduce deviations from the

prescribed timing [4]. These fluctuations require a high level of control over relative timing, where the phase of notes produced by the musician deviating from the timing aligns differently with the phases of other musicians. Rehearsals often involve reaching a consensus on expressive variations, ensuring that timing deviations are synchronized among players while maintaining relative timing [5]. Nevertheless, even with a unified understanding of the musical interpretation, individual musicians may opt to vary the timing of note onsets in specific passages between different performances [5,6]. Musical performance timing is also susceptible to inadvertent variations due to factors such as rhythmic intricacies, technical demands beyond timing (e.g., pitch, volume), lapses in concentration, and the inherent variability of biological timing [7]. While extensive individual practice can mitigate some of these unintended variations, complete elimination is unlikely.

The previous discussion raises the inquiry: how do musicians within an ensemble modulate their temporal dynamics to achieve synchronization with one another? In this paper a novel feedback correction model is presented, based on the Kalman Filter and aimed at improving timing accuracy in ensemble music performances. The proposed model generalizes the linear autoregressive model in [8] with the improvement of allowing two important quantities, the *phase* and *period correction gains*, to vary along time, since it makes the model suitable to describe synchronization in scenarios where the underlying tempo greatly varies (a realistic case in ensemble performance).

The paper is organized as follows: Section 2 recalls some linear models for synchronization, and the dynamic generalization of the model in [8] is presented, which is formulated as a Kalman Filter in Section 4; the fundamentals of the Kalman Filter are briefly recalled in Section 3; the computational experiments are shown and discussed in Section 5; conclusions are presented in Section 6. Directions for future work are identified throughout the paper.

2. LINEAR MODELS FOR ENSEMBLE SYNCHRONIZATION

The starting point for contextualizing the proposed model is [9], where a phase-correction model is presented as a method for an individual performer to achieve synchrony with a periodic metronome click or with another performer



© H. T. Carvalho, M. S. Li, M. di Luca, and A. M. Wing. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** H. T. Carvalho, M. S. Li, M. di Luca, and A. M. Wing, "A Kalman Filter model for synchronization in musical ensembles", in *Proc. of the 25th Int. Society for Music Information Retrieval Conf.*, San Francisco, United States, 2024.

(see also [10]). The fundamental concept revolves around utilizing the asynchrony, termed as a *phase error*, between a tone onset and the metronome click (or between two tone onsets produced by different performers) in a feedback mechanism, that guides the performer to adjust the time interval preceding the next tone onset. Consequently, the performer either decreases or increases the interval leading up to the subsequent tone onset proportionally to the preceding asynchrony. This process aims to achieve greater synchrony (“in phase”) between the next tone onset and the metronome click (or pair of tone onsets). This synchronization scheme can be represented by Equation (1):

$$t_n = t_{n-1} + T_n - \alpha A_{n-1} + \varepsilon_n, \quad (1)$$

where t_n and t_{n-1} represent the current and previous observed tone onset event times respectively, T_n denotes the time interval generated by an internal timekeeping mechanism, α is the *phase correction strength* or *phase correction gain*, A_{n-1} refers to the asynchrony of the previous onset event, and ε_n represents a random error term, which includes the internal timekeeper noise. The complete reduction of asynchrony to zero hinges on the value assigned to the gain, α , since this parameter determines the proportion of the preceding phase error that the performer endeavors to eliminate.

Following [11], *period correction* can also be incorporated in the model in Equation (1), by imposing that

$$T_n = T_{n-1} - \beta A_{n-1}, \quad (2)$$

where β is the *period correction strength* or *period correction gain*. Phase correction involves a local, within-cycle adjustment to the timing, while period correction entails a more enduring alteration to the underlying tempo, influencing subsequent cycles as well. Phase correction typically occurs automatically, without the need for conscious awareness of synchronization discrepancies. However, period correction appears to be more cognitively demanding, relying on the conscious detection of tempo variations in the external rhythm [12, 13].

As previously mentioned, Equations (1) and (2) model the asynchrony correction of an individual tapping according to a periodic metronome, or between two individuals tapping together. In [8] it is argued that the same modeling framework is also suited to describe synchronization in music ensemble performance, where a specific musician now tries to reduce asynchrony between him/her and every other performer. Therefore, Equations (1) and (2) can be jointly generalized to an ensemble of K performers as:

$$t_{i,n} = t_{i,n-1} + T_{i,n} - \sum_{\substack{i=1 \\ j \neq i}}^K \alpha_{ij} A_{ij,n-1} + \varepsilon_{i,n} \quad (3)$$

$$T_{i,n} = T_{i,n-1} - \sum_{\substack{i=1 \\ j \neq i}}^K \beta_{ij} A_{ij,n-1}, \quad (4)$$

where $i = 1, \dots, K$ indicate a specific performer, $t_{i,n}$ and $t_{i,n-1}$ are respectively the current and previous ob-

served tone onset event times for player i , $T_{i,n}$ is the timekeeper interval for player i at time instant n , $A_{ij,n-1} = (t_{i,n-1} - t_{j,n-1})$ is the asynchrony at the time instant $n-1$ between players i and j , α_{ij} and β_{ij} are respectively the phase and period correction gain applied by player i to compensate for $A_{ij,n-1}$, and $\varepsilon_{i,n}$ is a noise term identified with the internal timekeeper. Estimation of the values of α_{ij} and β_{ij} can be performed using the *bounded Generalized Least Squares* method (bGLS) [14, 15].

In [8], the model in Equation (3) is implemented and largely investigated for the case of a string quartet ensemble playing a homophonic section from the string quartet Op. 74 no. 1 by Joseph Haydn (fourth movement, bars 13–24), as this part has a steady tempo and all player’s quarter notes are aligned. In [14] the coupling of Equations (3) and (4) is investigated, with a simulated string quartet data with mild tempo changes, and the bGLS algorithm is shown to be capable of recovering the values of α and β . However, due to the nature of the bGLS algorithm, the authors point out that many data points are necessary for robust estimation of these variables, which may not be available or is an unrealistic aim in the case of a real-time implementation of the correction model (eg. for a virtual reality musical ensemble). In [16] the ADAM model (ADaptation and Anticipation Model) is proposed, including not only correction terms but also anticipatory ones, and in [17] this model is tested with tempo-changing tapping data, but since there is no adaptation of the bGLS algorithm to this new set of equations, the parameter estimation is done by exhaustive search, which is infeasible for real-life applications. Moreover, due to the nature of its parameters, the ADAM model is non-identifiable, meaning that more than one configuration of the parameters leads to the same estimate.

In order to circumvent the aforementioned issues, an alternative is to consider not a single value of α and β for each pair of performers through time, but *time-dependent correction gains*. Developing this intuition, a dynamic α_{ij} allows that a performer changes the phase correction at each onset, and a dynamic β_{ij} would allow him/her to correct differently for tempo variations during the performance of an excerpt. To model a dynamic variable, a good balance between simplicity and accuracy is a random walk, and in this case phase and period correction occur according to Equations (3) and (4), respectively, but with additional equations to allow the evolution of both correction gains. This new model is summarized in Equations (5), (6), (7), and (8):

$$t_{i,n} = t_{i,n-1} + T_{i,n} - \sum_{\substack{i=1 \\ j \neq i}}^K \alpha_{ij,n} A_{ij,n-1} + \varepsilon_{i,n} \quad (5)$$

$$T_{i,n} = T_{i,n-1} - \sum_{\substack{i=1 \\ j \neq i}}^K \beta_{ij,n} A_{ij,n-1} \quad (6)$$

$$\alpha_{ij,n} = \alpha_{ij,n-1} + w_{ij,n}^{(\alpha)} \quad (7)$$

$$\beta_{ij,n} = \beta_{ij,n-1} + w_{ij,n}^{(\beta)}, \quad (8)$$

where $w_{ij,n}^{(\alpha)}$ and $w_{ij,n}^{(\beta)}$ are independent zero-mean Gaussian random variables, allowing the evolution of $\alpha_{ij,n}$ and $\beta_{ij,n}$ through time, respectively (notice the novel subscript “ n ” in both α_{ij} and β_{ij}).

However, in the model proposed in Equations (5), (6), (7), and (8), it is not clear how to employ the bGLS method to obtain estimate of the variables of interest, and two distinct paths can now be followed: generalize the bGLS algorithm to this new situation, or resort to estimation techniques within the theory of dynamic models [18]. This work follows the latter, adopting the Kalman Filter as a framework to analyze Equations (5), (6), (7), and (8), due to its balance between flexibility and simplicity, as well as its simple and highly interpretable update equations. Section 3 recalls the basics of the Kalman Filter and Section 4 formulates the proposed model in this scenario.

3. A BRIEF RECALL ON THE KALMAN FILTER

The Kalman Filter (KF) was developed in the 1960’s, and served originally as a way to produce accurate estimates of variables of interest (eg. position of an object) by reaching a consensus between physical models and noisy measurements [19]. More generally, the KF can be seen as a state-space dynamic model, employed to describe more general time-series as a dynamic linear regression model as function of an underlying Markov model [18].

The main contribution of this paper is to propose the model in Equations (5), (6), (7), and (8), and formulate it as a KF, employing its filtering and smoothing equations to estimate the phase and period correction gains through time. The choice of a KF to achieve this goal are: linear nature of the model in Equations (5), (6), (7), and (8), high interpretability of the KF and its update equations, and potential low computational cost of its implementation.

The notation and basic equations of the KF are now briefly recalled, following [18]. In what follows, the index n ranges from 1 to N . Let $\mathbf{y}_n \in \mathbb{R}^m$ be a sequence of *observed variables* (or *measurements*), and $\boldsymbol{\theta}_n \in \mathbb{R}^p$ be a sequence of unobserved vectors, which are called the *hidden* (or *state*) variables. The KF model assumes that these two entities are related by Equations (9) and (10):

$$\mathbf{y}_n = \mathbf{F}_n \boldsymbol{\theta}_n + \mathbf{v}_n \quad (9)$$

$$\boldsymbol{\theta}_n = \mathbf{G}_n \boldsymbol{\theta}_{n-1} + \mathbf{w}_n, \quad (10)$$

where $\mathbf{F}_n \in \mathbb{R}^{m \times p}$ and $\mathbf{G}_n \in \mathbb{R}^{p \times p}$ are sequences of known matrices (*observation model* and the *state-transition model*, respectively). Vectors $\mathbf{v}_n \in \mathbb{R}^m$ and $\mathbf{w}_n \in \mathbb{R}^p$ are independent *observation* and *process* noise terms, respectively, and it is assumed that they follow Gaussian probability distributions, that is, $\mathbf{v}_n \sim N(\mathbf{0}, \mathbf{V}_n)$ and $\mathbf{w}_n \sim N(\mathbf{0}, \mathbf{W}_n)$,¹ where $\mathbf{V}_n \in \mathbb{R}^{m \times m}$ and $\mathbf{W}_n \in \mathbb{R}^{p \times p}$ are sequences of known covariance matrices of the observation and process noise terms respectively.

¹ The symbol \sim means “follows the probability distribution”, and $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes a multivariate Gaussian distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. The dimension of the support of the random vector is omitted, and compatibility between dimensions of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ is always assumed.

The KF dynamically estimates variables $\boldsymbol{\theta}_n$ and \mathbf{y}_n based on observations up to time $n - 1$, and updates the estimate of $\boldsymbol{\theta}_n$ when the observation at time n is available. This process is done accordingly to Equations (11), (12) and (13), called the *filtering equations*:²

Prediction step for hidden variables:

$$\boldsymbol{\theta}_n | \mathbf{y}_{1:n-1} \sim N(\mathbf{a}_n, \mathbf{R}_n) \quad (11)$$

Prediction step for observed variables:

$$\mathbf{y}_n | \mathbf{y}_{1:n-1} \sim N(\mathbf{f}_n, \mathbf{Q}_n) \quad (12)$$

Update step (compare predictions to measurements):

$$\boldsymbol{\theta}_n | \mathbf{y}_{1:n} \sim N(\mathbf{k}_n, \mathbf{C}_n), \quad (13)$$

where³

$$\mathbf{a}_n = \mathbf{G}_n \mathbf{k}_{n-1} \quad (14)$$

$$\mathbf{R}_n = \mathbf{G}_n \mathbf{C}_{n-1} \mathbf{G}_n^T + \mathbf{W}_n \quad (15)$$

$$\mathbf{f}_n = \mathbf{F}_n \mathbf{a}_n \quad (16)$$

$$\mathbf{Q}_n = \mathbf{F}_n \mathbf{R}_n \mathbf{F}_n^T + \mathbf{V}_n \quad (17)$$

$$\mathbf{k}_n = \mathbf{a}_n + [\mathbf{R}_n \mathbf{F}_n^T \mathbf{Q}_n^{-1}] \mathbf{e}_n \quad (18)$$

$$\mathbf{e}_n = \mathbf{y}_n - \mathbf{f}_n \quad (19)$$

$$\mathbf{C}_n = \mathbf{R}_n - [\mathbf{R}_n \mathbf{F}_n^T \mathbf{Q}_n^{-1}] \mathbf{F}_n \mathbf{R}_n, \quad (20)$$

assuming that the initial state is chosen according to a normal distribution, that is, $\boldsymbol{\theta}_0 \sim N(\mathbf{k}_0, \mathbf{C}_0)$. For more details on the KF, see [18, 19].

One of the appealing aspects of the KF is its ability to perform estimation and forecasting sequentially, as new data emerge. However, if observations \mathbf{y}_n for $n = 1, \dots, N$ are available beforehand, one is also able to retrospectively reconstruct the system’s states, in order to analyze its behavior given all the observations. For this purpose, a backward-recursive algorithm can be employed to compute the conditional distributions of $\boldsymbol{\theta}_n$ given $\mathbf{y}_{1:N}$, for any $n < N$ [18, 19]. The main ingredient of this algorithm is the *smoothing equation* (21):

$$\boldsymbol{\theta}_n | \mathbf{y}_{1:N} \sim N(\mathbf{s}_n, \mathbf{S}_n), \quad (21)$$

where

$$\mathbf{s}_n = \mathbf{k}_n + \mathbf{C}_n \mathbf{G}_{n+1}^T \mathbf{R}_{n+1}^{-1} [\mathbf{s}_{n+1} - \mathbf{a}_{n+1}] \quad (22)$$

$$\mathbf{S}_n = \mathbf{C}_n - \mathbf{C}_n \mathbf{G}_{n+1}^T \mathbf{R}_{n+1}^{-1} \times [\mathbf{R}_{n+1} - \mathbf{S}_{n+1}] \mathbf{R}_{n+1}^{-1} \mathbf{G}_{n+1} \mathbf{C}_n, \quad (23)$$

assuming that $\boldsymbol{\theta}_{n+1} | \mathbf{y}_{1:N} \sim N(\mathbf{s}_{n+1}, \mathbf{S}_{n+1})$. Notice that since the smoothing is performed backwards, it is necessarily to previously filter the set of observations to gain access to vectors \mathbf{k}_n and \mathbf{a}_n , and matrices \mathbf{C}_n and \mathbf{R}_n .

4. KALMAN FILTER MODEL FOR ENSEMBLE SYNCHRONIZATION

Equations (5), (6), (7), and (8) can be written as a KF by considering proper choices for the observed and hidden

² The conditional distribution of \mathbf{u} given \mathbf{z} is denoted by $\mathbf{u} | \mathbf{z}$, and $i : j$ means “observations from time instants i to j ”, both extremes included.

³ The superscript T after a vector or matrix denotes its transpose; the superscript $^{-1}$ after a matrix denotes its inverse.

variables, as well as the observation and state-transition matrices. The main goal of this section is to construct a sequence of matrices \mathbf{F}_n and \mathbf{G}_n , as well as vectors \mathbf{y}_n and $\boldsymbol{\theta}_n$ of observed and hidden variables respectively, such that Equations (9) and (10) recover the model proposed in Equations (5), (6), (7), and (8). Firstly, to simplify the formulation of the model, the observed variables are not the tone onset times for each player, but rather the *inter-onset-intervals* (IOIs), denoted by $r_{i,k} = t_{i,n} - t_{i,n-1}$, for $i = 1, \dots, K$. These values are assembled as in Equation (24):

$$\mathbf{y}_n = [r_{1,n} \dots r_{K,n}] \in \mathbb{R}^K. \quad (24)$$

The hidden variable $\boldsymbol{\theta}_n$ can be written as in Equation (25):

$$\boldsymbol{\theta}_n = [\mathbf{T}_n^T \mid \mathbf{r}_n^T \mid \boldsymbol{\alpha}_n^T \mid \boldsymbol{\beta}_n^T]^T \in \mathbb{R}^{2K^2}, \quad (25)$$

where

$$\mathbf{T}_n = [T_{1,n} \dots t_{K,n}]^T \in \mathbb{R}^K \quad (26)$$

$$\mathbf{r}_n = [r_{1,n} \dots r_{K,n}]^T \in \mathbb{R}^K \quad (27)$$

$$\boldsymbol{\alpha}_n = [\alpha_{ij,n} \text{ in the lexicographical order on } ij, \text{ for } 1 \leq i, j \leq K, i \neq j] \in \mathbb{R}^{K(K-1)} \quad (28)$$

$$\boldsymbol{\beta}_n = [\beta_{ij,n} \text{ in the lexicographical order on } ij, \text{ for } 1 \leq i, j \leq K, i \neq j] \in \mathbb{R}^{K(K-1)}. \quad (29)$$

The relation between $\boldsymbol{\theta}_n$ and \mathbf{y}_n is described by the observation matrix in Equation (30):⁴

$$\mathbf{F}_n = [\mathbf{0}_K \mid \mathbf{I}_K \mid \mathbf{0}_{K \times K(K-1)} \mid \mathbf{0}_{K \times K(K-1)}]. \quad (30)$$

Notice that matrices $\mathbf{F}_n \in \mathbb{R}^{K \times 2K^2}$ are constant through time. The evolution of the hidden variables in $\boldsymbol{\theta}_n$ is modelled by a sequence of state-transition matrices $\mathbf{G}_n \in \mathbb{R}^{2K^2 \times 2K^2}$, described in Equation (31):

$$\begin{bmatrix} \mathbf{I}_K & \mathbf{0}_K & \mathbf{0}_{K \times K(K-1)} & \mathbf{G}_n^{T\beta} \\ \mathbf{I}_K & \mathbf{0}_K & \mathbf{G}_n^{r\alpha} & \mathbf{G}_n^{r\beta} \\ \mathbf{0}_{K(K-1) \times K} & \mathbf{0}_{K(K-1) \times K} & \mathbf{I}_{K(K-1)} & \mathbf{0}_{K(K-1)} \\ \mathbf{0}_{K(K-1) \times K} & \mathbf{0}_{K(K-1) \times K} & \mathbf{0}_{K(K-1)} & \mathbf{I}_{K(K-1)} \end{bmatrix}, \quad (31)$$

where matrices $\mathbf{G}_n^{T\beta}$, $\mathbf{G}_n^{r\alpha}$, and $\mathbf{G}_n^{r\beta}$ (of dimensions $K \times K(K-1)$ each) describe the interaction between variables in their respective superscripts. These three matrices are equal to the matrix in Equation (32):

$$\begin{bmatrix} -\mathbf{A}_{1:,n-1}^T & \mathbf{0}_{1 \times (K-1)} & \cdots & \mathbf{0}_{1 \times (K-1)} \\ \mathbf{0}_{1 \times (K-1)} & -\mathbf{A}_{2:,n-1}^T & \cdots & \mathbf{0}_{1 \times (K-1)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{1 \times (K-1)} & \mathbf{0}_{1 \times (K-1)} & \cdots & -\mathbf{A}_{K:,n-1}^T \end{bmatrix}, \quad (32)$$

where each $\mathbf{A}_{i:,n-1} \in \mathbb{R}^{K-1}$ contain the asynchronies $A_{ij,n-1}$ of player i to all players j , for $j \neq i$, at time $n-1$. Vector $\mathbf{A}_{i:,n-1}$ is made explicit in Equation (33):

$$[\mathbf{A}_{i1,n-1} \dots \mathbf{A}_{i(i-1),n-1} \mathbf{A}_{i(i+1),n-1} \dots \mathbf{A}_{iK,n-1}]^T. \quad (33)$$

⁴ The identity matrix of dimensions $L \times L$ is denoted by \mathbf{I}_L ; the matrix of dimensions $L \times M$ filled with zeros is denoted by $\mathbf{0}_{L \times M}$; a square null matrix of dimensions $L \times L$ is abbreviated by $\mathbf{0}_L$.

A simple (but tedious) verification using Equations (9) and (10) with these choices for \mathbf{F}_n , \mathbf{G}_n , \mathbf{y}_n , and $\boldsymbol{\theta}_n$ ensures that the model in Equations (5), (6), (7), and (8) is recovered. It is also established that when $K = 1$ the model in Equations (1) and (2) is recovered, with the improvement of dynamic α and β .

When compared to the bGLS algorithm [14, 15], the state-of-the-art to estimate parameters in the scenario of sensorimotor synchronization, the KF model presents a great advantage, that is the possibility of performing on-line estimation as more data become available: this feature can be important if one desires to implement real-time synchronization schemes. When the complete time-series of onset times/IOIs is available, one can apply the smoothing equation (21), in order to dynamically estimate the parameters of interest throughout the performance, as well as estimate them by applying the filtering equations (11), (12), and (13), for example, to simulate an online scenario.

Notice that the dimensions of \mathbf{F}_n , \mathbf{G}_n , and $\boldsymbol{\theta}_n$ scale quadratically with the number of performers, which may render the model overly complicated or cause computational issues when computing the KF update/filtering equations.⁵ However, due to the sparseness of matrices \mathbf{F}_n and \mathbf{G}_n , block-multiplication will highly reduce the number of operations when computing Equations (14) to (20), mitigating the latter issue. Regarding the complexity of the model, notice that in real large-scale scenarios (eg. a symphony orchestra) it is not realistic to assume that each musician synchronizes with every other, thus allowing for potential simplifications, like considering a group of instruments as a single unity and synchronizing with every other group. This procedure would diminish the value of K from approximately 100 to less than 20. A useful topic for future research would be to investigate the possibility of modeling the synchronization scheme between performers (or group of performers) in a graph, in order to decrease even more the number of relevant connections.

Another issue that is important to point out is the design of the covariance matrices for the observation and process noises, \mathbf{V}_n and \mathbf{W}_n respectively. On a first view, it makes sense to consider \mathbf{V}_n as diagonal matrices, for simplicity, since the interaction between the performers is already “captured” by the correction gains in the hidden variables; however, it is not clear if \mathbf{W}_n should be a sequence of diagonal matrices, since it makes sense to consider at least correlations between both correction gains of the same performer. This work employs a particular choice for these covariance matrices, as will be further discussed in Section 5. Further investigation on this question could involve coupling the Expectation-Maximization algorithm with the KF in order to estimate not only matrices \mathbf{V}_n and \mathbf{W}_n but also \mathbf{F}_n and \mathbf{G}_n [20]. However a disadvantage would be that these estimates would need to be static through time, requiring a large amount of data, and being highly dependent of the piece of music being analyzed.

⁵ Other computational issues on the Kalman Filter are largely discussed in [18].

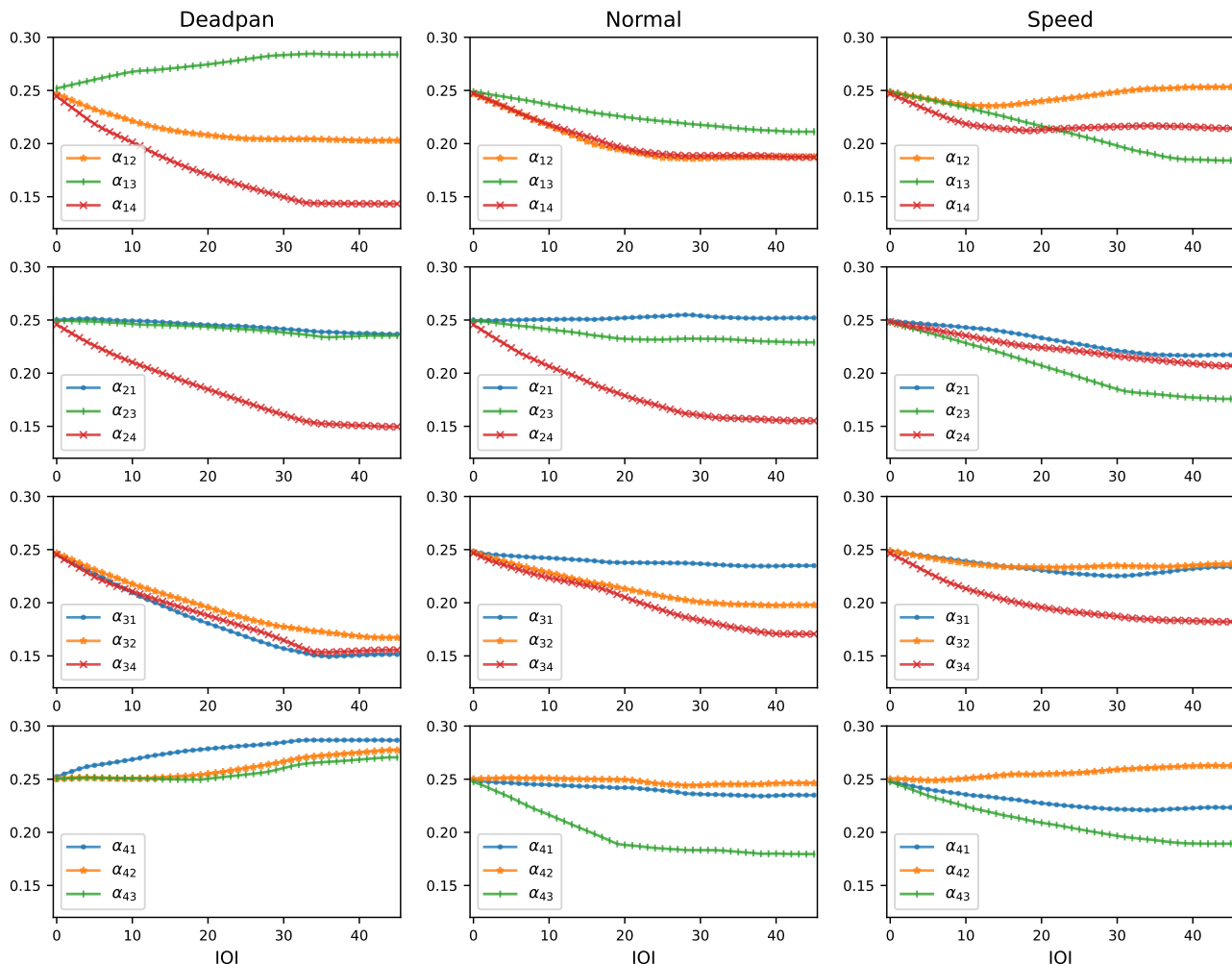


Figure 1. Smoothed time-series for the phase correction gain on three performance styles of an excerpt of the fourth movement of the string quartet Op. 74 no. 1, by Joseph Haydn. See Section 5 for discussion.

5. RESULTS

To illustrate the effectiveness of the proposed model, a set of simulations was performed, using an excerpt of the dataset presented in [21], similar to the one used in [8]: the homophonic section from the fourth movement of the string quartet Op. 74 no. 1 by Joseph Haydn, from bars 13 to 24. In this excerpt the instruments play a sequence of 47 quarter notes in rhythmic unison, with the first violin breaking the pattern near the end with an adornment of four sixteenth notes, which are disregarded in this study.

Three performance styles are considered: *Normal* condition (concert-style performance); *Speed* condition (including a spontaneous *accelerando* and *ritardando* initiated by a single musician – the designated leader, that can be the first or second violin); and *Deadpan* condition (performances with minimal expression in tempo and articulation). All the simulation were performed on a computer equipped with a 12th Generation Intel Core™ i7 processor and 16GB of RAM, running Windows 11 Pro™; the implementations were conducted in Python version 3.11.7.⁶

⁶ Codes available at <https://github.com/arme-project/ismir-2024>.

Regarding the parameters of the KF, the covariance matrix for the process noise, \mathbf{W}_n , plays an important role, since it indicates how the variables in θ interact. Based on the interpretation of the hidden variables, a reasonable choice for all the \mathbf{W}_n is the block-diagonal matrix in Equation (34):⁷

$$\begin{bmatrix} \mathbf{W}^{(T)} & & & \\ & \mathbf{W}^r & & \\ & & \mathbf{W}^\alpha & \\ & & & \mathbf{W}^\beta \end{bmatrix}, \quad (34)$$

where $\mathbf{W}^{(T)}$ and \mathbf{W}^r are given respectively by $\sigma_T^2 \mathbf{I}_K$ and $\sigma_r^2 \mathbf{I}_K$, being σ_T^2 the *timekeeper variance* and σ_r^2 the *motor variance*. Since it is known that the motor variance is way smaller than the timekeeper variance [8, 14, 15], the conservatively high values $\sigma_T^2 = 500$ and $\sigma_r^2 = 25$ were considered. Both \mathbf{W}^α and \mathbf{W}^β are also block-diagonal matrices, consisting of K blocks, each of dimen-

⁷ Off-diagonal blocks are null matrices, that were omitted exceptionally here, to avoid a line-break in the number of the equation. Moreover, notation $\mathbf{W}^{(T)}$ means to avoid confusion with the transpose matrix.

sions $(K - 1) \times (K - 1)$ and as in Equation (35):

$$\begin{bmatrix} v & c & \cdots & c \\ c & v & \cdots & c \\ \vdots & \vdots & \ddots & \vdots \\ c & c & \cdots & v \end{bmatrix}, \quad (35)$$

where v represents the variance of each α_{ij} (or β_{ij}) and c is the covariance between two distinct α_{ij} (or β_{ij}).

The rationale behind this construction for \mathbf{W}^α and \mathbf{W}^β is that it makes sense to assume that for performer i there is a correlation only between the α_{ij} (or β_{ij}) for $j \neq i$. This means that all the correction gains for performer i interact among themselves, but not directly with the correction gains of other performers. Also, it is expected that the correlation between two distinct α_{ij} (or β_{ij}) is negative, since increasing correction towards a specific performer may cause a decrease of the synchronization towards the others. With this in mind, for matrix \mathbf{W}^α the value of v was chosen as 10^{-4} ; the value of c was chosen such that the correlation between any two distinct α_{ij} is equal to -0.1 .⁸

In this preliminary set of experiments with the KF, the effect of the β_{ij} was disregarded, by considering \mathbf{W}^β a null matrix. It is known that the effect of the phase correction is way more relevant than the effect of the period correction [8, 14, 15], with the β_{ij} coefficients being usually much smaller than the α_{ij} . Also, preliminary experiments with artificial data also indicate that the dynamic values of the phase correction may render the period correction unnecessary. Since this is a point to be further investigated, it seemed safe to first experiment only with phase correction.

Matrices \mathbf{V}_n were chosen to be constant and equal to $10^{-5}\mathbf{I}_K$: since the block \mathbf{W}^r in matrix \mathbf{W}_n already captures the motor variance, \mathbf{V}_n should be a sequence of null matrices, but a negligible diagonal term was added to avoid numerical errors. Finally, the initialization of vector θ was done by choosing its first K components and the components from $K + 1$ to $2K$ to be equal to the first IOI of each of the K instruments, all the α_{ij} were initially set to 0.25, and all the β_{ij} to zero. This initialization of α_{ij} is supported by [8], where optimal correction values for ensembles of size K were derived.

Figure 1 summarizes one experiment performed in the aforementioned scenario. Three repetitions of the Haydn quartet excerpt were analyzed, being one for each of the three performance conditions, having the second violin as the leader in the ‘‘Speed’’ case. Each performance consists of a sequence of 46 four-dimensional vectors containing the IOIs for each instrument. Since it is not the goal of this set of experiments to evaluate online performance of the proposed model, these three sequences were smoothed by the KF,⁹ according to Equation 21. Each panel of Figure 1 displays the evolution of the α_{ij} , for $j \neq i$, organized as follows: each column contains a performance condition (made explicit at its top), and each row displays the evolution of α_{ij} for $j \neq i$ and a fixed value of i . The condi-

tioning of each α_{ij} on $\mathbf{y}_{1:N}$ is omitted, and the instruments are abbreviated by numbers, where 1, 2, 3, and 4 refers to the first violin, second violin, viola, and cello, respectively. On each panel of Figure 1 the values of α_{ij} promptly deviates from the optimal initialization of 0.25 (but still varies around it), and their respective behavior are now discussed.

In ‘‘Speed’’ condition (third column in Figure 1), on each panel the phase correction parameter toward the second violin (α_{i2} , for $i = 1, 3, 4$) shows a small increase by the end of the performance, when the change in speed occurs, since the second violin is assigned as the leader to initiate this change in speed. Notice also that his/her phase correction parameters towards the other performers (α_{2j} , for $j = 1, 3, 4$) decrease through time, specially near the last notes, reinforcing its leadership in this tempo change.

In the ‘‘Normal’’ condition (second column in Figure 1) it is noticeable that the second violin, viola, and cello are systematically synchronizing mainly to the first violin, which plays the melody in this excerpt: notice the almost constant value for α_{i1} , for $i = 2, 3, 4$. While the cello is synchronizing mainly with the first and second violin, it presents the weaker ‘‘synchronization attractor’’, as seen by the significant decrease in α_{i4} through time, for $i = 1, 2, 3$.

Finally, in the ‘‘Deadpan’’ condition (first column in Figure 1) the first and second violin and the cello are synchronizing mainly to the viola (steady increase of α_{i3} for $i = 1, 2, 4$, and decrease in α_{3j} for $j = 1, 2, 4$), which may be the cause of the cello synchronizing systematically with all the three other instruments.

This experiment indicates that the proposed model is capable of capturing local fluctuations in tempo, reinforces the role of the phase correction gain in interpreting synchronization mechanisms in musical ensembles, and assess qualitatively the validity of a time-varying model to the problem of ensemble synchronization. As a next step in this new direction for the field, the proposed model will be broadly tested and systematically compared with other models. Some issues to be addressed in future work are: perform experiments with other data contained on [21]; compare filtering and smoothing procedures, as well as investigate if the filtered estimates make sense from a music cognition perspective; implement tools from the theory of dynamic linear models to automatically estimate the covariance matrices \mathbf{V}_n and \mathbf{W}_n [18]; perform a systematic comparison with the bGLS and ADAM algorithms.

6. CONCLUSION

This paper presented a novel model, based on the Kalman Filter, for analysing asynchrony correction in music ensemble performances. The proposed model is founded on well-established models in the literature, and has the advantage of considering dynamic phase and period correction gains. A set of experiments (using only phase correction) on a homophonic section of a string quartet by J. Haydn was conducted, illustrating the capabilities of the model in explaining synchronization schemes within musical ensembles.

⁸ This procedure will not always lead to a positive-definite matrix, for sufficiently high value of K and depending on c – not the case here.

⁹ The computational time of each smoothing is less than 100ms.

7. ACKNOWLEDGMENTS

The ARME Project (Augmented Reality Music Ensemble – <https://arme-project.co.uk/>) is funded by the EPSRC grant with reference EP/V034987/1. We would like to thank the University of Birmingham for the scholarship awarded to the first author through the Brazil Visiting Fellows Scheme. We also would like to thank the reviewers and meta-reviewers for the useful insights and suggestions to improve this paper.

8. REFERENCES

- [1] A. D. Patel, J. R. Iversen, M. R. Bregman, and I. Schulz, “Experimental evidence for synchronization to a musical beat in a nonhuman animal,” *Current Biology*, vol. 19, no. 10, pp. 827–830, 2009.
- [2] A. M. Wing and C. Woodburn, “The coordination and consistency of rowers in a racing eight,” *Journal of Sports Sciences*, vol. 13, no. 3, pp. 187–197, 1995.
- [3] E. Goodman, “Ensemble performance,” in *Musical performance: a guide to understanding*, J. Rink, Ed. Cambridge, UK: Cambridge University Press, 2002, pp. 153–167.
- [4] C. Palmer, “Music performance,” *Annual Review of Psychology*, vol. 48, pp. 115–138, 1997.
- [5] J. W. Davidson and J. M. M. Good, “Social and musical co-ordination between members of a string quartet: An exploratory study,” *Psychology of Music*, vol. 30, no. 2, pp. 186–201, 2002.
- [6] J. K. Murnighan and D. E. Conlon, “The dynamics of intense work groups: A study of british string quartets,” *Administrative Science Quarterly*, vol. 36, no. 2, pp. 165–186, 1991.
- [7] J. Gibbon, C. Malapani, C. L. Dale, and C. R. Gallistel, “Toward a neurobiology of temporal cognition: advances and challenges,” *Current Opinion in Neurobiology*, vol. 7, no. 2, pp. 170–184, 1997.
- [8] A. M. Wing, S. Endo, A. Bradbury, and D. Vorberg, “Optimal feedback correction in string quartet synchronization,” *Journal of the Royal Society Interface*, vol. 11, no. 20131125, 2014.
- [9] D. Vorberg and H. H. Schulze, “Linear phase correction in synchronization: predictions, parameter estimation, and simulations,” *Journal of Mathematical Psychology*, vol. 46, no. 1, pp. 56–87, 2002.
- [10] D. Vorberg and A. M. Wing, “Modeling variability and dependence in timing,” in *Handbook of perception and action*, vol. 2, H. Heuer and S. Keele, Eds. New York, USA: Academic Press, 1996, pp. 181–262.
- [11] J. Mates, “A model of synchronization of motor acts to a stimulus sequence,” *Biological Cybernetics*, vol. 71, p. 186, 1994.
- [12] B. H. Repp, “Processes underlying adaptation to tempo changes in sensorimotor synchronization,” *Human Movement Science*, vol. 20, no. 3, pp. 277–312, 2001.
- [13] B. H. Repp and P. E. Keller, “Adaptation to tempo changes in sensorimotor synchronization: effects of intention, attention, and awareness,” *Quarterly Journal of Experimental Psychology*, vol. 57, no. 3, pp. 499–521, 2004.
- [14] N. Jacoby, N. Tishby, B. H. Repp, M. Ahissar, and P. E. Keller, “Parameter estimation of linear sensorimotor synchronization models: Phase correction, period correction, and ensemble synchronization,” *Timing & Time Perception*, vol. 3, no. 1–2, pp. 52–87, 2015.
- [15] N. Jacoby, P. E. Keller, B. H. Repp, M. Ahissar, and N. Tishby, “Lower bound on the accuracy of parameter estimation methods for linear sensorimotor synchronization models,” *Timing & Time Perception*, vol. 3, no. 1–2, pp. 32–51, 2015.
- [16] M. C. van der Steen and P. E. Keller, “The ADaptation and Anticipation Model (ADAM) of sensorimotor synchronization,” *Frontiers in Human Neuroscience*, vol. 7, 2013.
- [17] B. Harry and P. E. Keller, “Tutorial and simulations with ADAM: an adaptation and anticipation model of sensorimotor synchronization,” *Biological Cybernetics*, vol. 113, pp. 397–421, 2019.
- [18] G. Petris, S. Petrone, and P. Campagnoli, *Dynamic Linear Models with R*. Berlin/Heidelberg, Germany: Springer, 2009.
- [19] M. S. Grewal and A. P. Andrews, *Kalman Filtering: Theory and Practice with MATLAB*. New Jersey, USA: Wiley-IEEE Press, 2014.
- [20] W. Mader, Y. Linke, M. Mader, L. Sommerlade, J. Timmer, and B. Schelter, “A numerically efficient implementation of the expectation maximization algorithm for state space models,” *Applied Mathematics and Computation*, vol. 241, pp. 222–232, 2014.
- [21] M. Tomczak, M. S. Li, and M. D. Luca, “Virtuoso strings: A dataset of string ensemble recordings and onset annotations for timing analysis,” in *Extended Abstracts for the Late-Breaking Demo Session of the 24th Int. Society for Music Information Retrieval Conf.*, Milan, Italy, 2023.

STEM-JEPA: A JOINT-EMBEDDING PREDICTIVE ARCHITECTURE FOR MUSICAL STEM COMPATIBILITY ESTIMATION

Alain Riou^{1,2} Stefan Lattner² Gaëtan Hadjeres³ Michael Anslow² Geoffroy Peeters¹

¹ LTCI, Télécom-Paris, Institut Polytechnique de Paris, France

² Sony Computer Science Laboratories - Paris, France

³ Sony AI

alain.riou@sony.com

ABSTRACT

This paper explores the automated process of determining stem compatibility by identifying audio recordings of single instruments that blend well with a given musical context. To tackle this challenge, we present Stem-JEPA, a novel Joint-Embedding Predictive Architecture (JEPA) trained on a multi-track dataset using a self-supervised learning approach.

Our model comprises two networks: an encoder and a predictor, which are jointly trained to predict the embeddings of compatible stems from the embeddings of a given context, typically a mix of several instruments.

Training a model in this manner allows its use in estimating stem compatibility—retrieving, aligning, or generating a stem to match a given mix—or for downstream tasks such as genre or key estimation, as the training paradigm requires the model to learn information related to timbre, harmony, and rhythm.

We evaluate our model’s performance on a retrieval task on the MUSDB18 dataset, testing its ability to find the missing stem from a mix and through a subjective user study. We also show that the learned embeddings capture temporal alignment information and, finally, evaluate the representations learned by our model on several downstream tasks, highlighting that they effectively capture meaningful musical features.

1. INTRODUCTION

Musical stem compatibility indicates the degree to which a stem (i.e., an audio file of a single instrument) fits a given musical context (an audio file of another instrument or a mix of instruments) when played together. Its automatic estimation can be helpful for stem retrieval, automatic arrangement, or stem generation tasks. The compatibility between stems (or a stem and some musical context) depends on several global factors, such as tonality, tempo,

genre, timbre, and singing/playing style. In addition, local features like chords or pitches are crucial to performing temporal alignment between a stem and some musical context.

While initial works have studied musical compatibility between songs based on traditional Music Information Retrieval (MIR) tasks like beat tracking and chord estimation [1, 2], more modern approaches aim to learn compatibility directly from data using deep neural networks [3–5]. Using such learning-based approaches extends the notion of compatibility beyond music-theoretical aspects (like tonality and tempo) toward sound-related and expressive characteristics like timbre and playing style.

Moreover, there are potential applications for musical stem generation [6, 7], where generators usually require musical context conditioning to produce compatible accompaniments. With the proposed system, stem representations can be predicted from context information at inference time. This allows training a stem generation model based solely on stem representations, eliminating the need for context/target pairs.

Paper proposal and organization. In this paper, we introduce Stem-JEPA, a novel Joint-Embedding Predictive Architecture (JEPA) which acts directly on mixtures of stems. It consists of two neural networks, an encoder and a predictor, jointly trained to produce representations of a *context* mix and predict representations of a compatible *target* stem. Unlike previous JEPAs [8, 9], our approach does not rely on masking in the input space but rather on omitting stems within the process of mixing, and it uses the label of the missing stem for conditioning (see section 3).

We assess the performance of Stem-JEPA in a retrieval task and through a subjective evaluation in sections 4.1 and 4.2, respectively. Also, we investigate how well the learned representations encode the temporal alignment of stems and mixes (see section 4.3). We also perform an analysis showing that key and chord annotations of audio snippets close in the embedding space are musically compatible (see section 4.4). Finally, we evaluate the representations produced by our model on various downstream MIR tasks (see section 4.5).

To facilitate further research in this direction, we make our code available.¹

¹ <https://github.com/SonyCSLParis/Stem-JEPA>



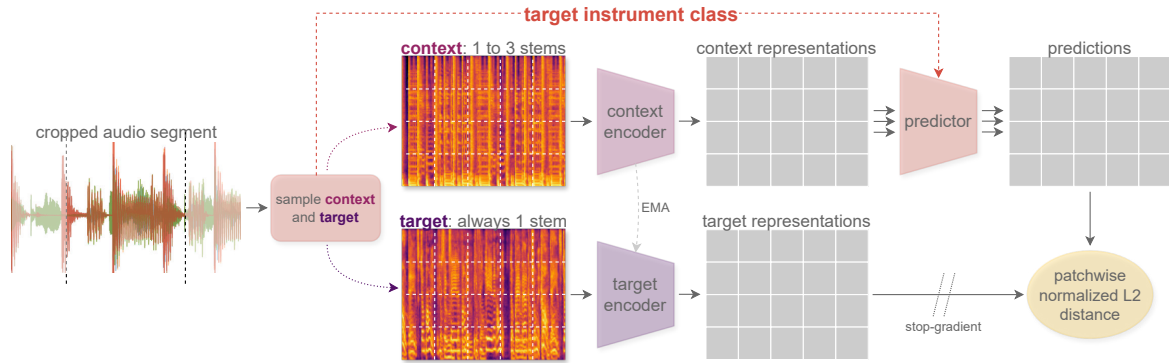


Figure 1. Overview of the Stem-JEPA framework. From an audio clip composed of 4 stems, we first crop a chunk of 8 seconds, then sample the target \bar{x} (one of the stems) and the context x (a mix of some of the remaining stems) as described in section 3.2. They are then converted into Log Mel Spectrograms and passed through the *context* and *target* encoders, respectively. Finally, the *predictor* (conditioned on the target instrument label) is trained so that each of its outputs individually predicts each target representation.

2. RELATED WORK

SSL for representation learning. Self-supervised learning (SSL) involves training networks on unlabeled corpora by solving pretext tasks using only the data itself. This paradigm has shown great potential for extracting meaningful representations in various domains [10–13].

A common approach to SSL is contrastive learning [10, 12, 14, 15] or its variants [16, 17]. In autoencoders, an encoder and a decoder are jointly trained to learn latent representations from which the original input can be reconstructed [13, 18]. JEPAs are trained to *predict* some target data from context data directly in the representation space [8, 19].

Joint-Embedding Predictive Architectures. A JEPA is an architecture composed of two trainable networks: an encoder and a predictor. The model receives a context/target pair as input, passes them through the encoder to create latent representations, and then the predictor is trained to predict the target representation from the context representation. Pairs can be generated through various data augmentations [11, 19, 20], or by masking part of the input, as in data2vec [21]. In particular, JEPAs do not require negative samples, unlike contrastive approaches [19], and enable the model to discard uninformative content given that reconstruction is not required.

To prevent model collapse, it is crucial to block gradients in the non-predictor branch [22], treating its output as the target. Moreover, adopting different but tied encoders for each branch as in Eq. (2) helps to stabilize training [19, 22, 23]. Finally, I-JEPA [8] creates pairs through masking and trains the model to predict the representations of small image patches by conditioning the predictor on their positions, allowing the model to grasp local nuances.

Learning from separated sources. Most SSL approaches, often stemming from the vision domain, have been explored and adapted to the audio domain [9, 12, 14, 15, 20, 24–26]. These works are not specific to musical audio, which is typically composed of several stems providing rich compositional potential for SSL. In practice, only

a few SSL approaches leverage separated stems for tasks such as audio classification [27], music tagging [28] and beat tracking [29]. Finally, a few works explore modeling the compatibility between stems with applications like automatic mashup creation [1, 3] and sample or loop retrieval for interactive composition [4, 5].

3. STEM-JEPA

3.1 Training pipeline

Our method, depicted in Figure 1, builds upon recent works in JEPAs for image and audio representation learning [8, 9]. Given a music track represented as a set of S stems (roughly corresponding to the separated audio sources) $\mathbf{x}_1, \dots, \mathbf{x}_S$, we crop a chunk of 8 seconds. We then randomly select one of the stems as **target** $\bar{x} = x_t$ with $t \in \{1, \dots, S\}$ and use the remaining ones to create a **context mix**: $\mathbf{x} = \sum_{c \in C} \mathbf{x}_c$ with $C \subset \{1, \dots, S\} \setminus \{t\}$.

Both \mathbf{x} and \bar{x} are then converted to Log Mel Spectrograms and divided into a regular grid (over the time and frequency dimensions), leading to K patches. The context patches are then fed to a **context encoder** f_θ to produce patch-wise embeddings $\mathbf{z} = (\mathbf{z}_1, \dots, \mathbf{z}_K)$, where θ are training parameters. Similarly, the target patches are fed to a **target encoder** $f_{\bar{\theta}}$ to produce the patch-wise embeddings $\bar{\mathbf{z}} = (\bar{\mathbf{z}}_1, \dots, \bar{\mathbf{z}}_K)$.

Finally, the context representations \mathbf{z} are independently fed to a **predictor** g_ϕ (with trainable parameters ϕ), which is conditioned on the instrument label l of the missing stem by concatenating a learnable embedding $\text{emb}(l)$ to \mathbf{z}_k . The output of the predictor is therefore the prediction $\tilde{\mathbf{z}} = (\tilde{\mathbf{z}}_1, \dots, \tilde{\mathbf{z}}_K)$, with $\tilde{\mathbf{z}}_k = g_\phi(\text{concat}(\mathbf{z}_k, \text{emb}(l)))$.

As in [8, 11, 19], the parameters (θ, ϕ) of the context encoder and predictor are updated through gradient descent by minimizing the mean squared error $\mathcal{L}(\tilde{\mathbf{z}}, \bar{\mathbf{z}})$ between the (normalized) predicted and target representations:

$$\mathcal{L}(\tilde{\mathbf{z}}, \bar{\mathbf{z}}) = \frac{1}{K} \sum_{k=1}^K \left\| \frac{\tilde{\mathbf{z}}_k}{\|\tilde{\mathbf{z}}_k\|} - \frac{\bar{\mathbf{z}}_k}{\|\bar{\mathbf{z}}_k\|} \right\|^2, \quad (1)$$

whereas the parameters of the target encoder $\bar{\theta}$ are updated using an Exponential Moving Average (EMA) of the ones of the context encoder, i.e.,

$$\bar{\theta}_i = \tau_i \bar{\theta}_{i-1} + (1 - \tau_i) \theta_i, \quad (2)$$

where the EMA rate τ_i is linearly interpolated between τ_0 and τ_T , T being the total number of training steps.

3.2 Sampling context and target

To avoid training the system on silent target stems or silent context mixes, we first analyze the amplitude content of each of the stems $\mathbf{x}_1, \dots, \mathbf{x}_S$ representing a chunk of a given music track.

Let $\mathcal{A} \subset \{1, \dots, S\}$ be the indices of active (i.e., non-silent) stems among $\mathbf{x}_1, \dots, \mathbf{x}_S$. We first pick a random index $t \in \mathcal{A}$ as target². Then, we randomly select a subset $C \subset \mathcal{A} \setminus \{t\}$ from the remaining non-silent tracks. The number of stems $|C|$ in this subset is uniformly sampled between 1 and the number of other non-silent stems $|\mathcal{A}| - 1$. Most of the time, the prediction task incorporates more stems in the context than in the target ($|C| > 1$), simplifying the predictor’s task. However, occasionally, the subset consists of only one stem ($|C| = 1$), allowing the model to process individual stems and learn their representations, which is crucial as these are also used as targets.

3.3 Architecture and training details

We employ a standard ViT-Base model as the encoder [30]. Our predictor is a 6-layer MLP with ReLU activations and 1024 dimensions in each hidden layer. In our ablation studies, the Transformer predictor we use is the same as in [9].

During training, we extract audio chunks of 8 seconds that are converted to log-scaled Mel Spectrograms with 80 mel bins and a window and hop size of 25 and 10 ms, respectively. We use patches of size 16×16 , leading to sequences of $\frac{80}{16} \times \frac{800}{16} = 250$ tokens during training.

We train our model during 300k steps using AdamW [31], with a batch size of 256, a base learning rate of $3e-4$, and a cosine annealing scheduling after 20k steps of linear warmup. All other hyperparameters are consistent with those used in [9], following their demonstrated effectiveness. Our model is trained for approximately four days on a single A100 GPU with 40 GB of memory.

3.4 Training data

We train the model on a proprietary dataset of 20k multi-track recordings of diverse music genres (e.g., pop/rock, R&B, rap, country) with a total duration of 1350 hours. We use existing instrument annotations to construct four standard categories: Bass, Drums, Vocals, and Other.

² If $|\mathcal{A}| < 2$ (a whole chunk is silent or only one active stem), we re-sample another audio chunk from the same track to prevent having silent context or target.

4. EVALUATION

We assess the efficacy of our model to retrieve compatible stems from a given mix through objective and subjective evaluations. We also demonstrate that the learned representations capture local harmonic and rhythmic information. Finally, we show that they also encode high-level features, making them suitable for various MIR tasks.

4.1 Stem retrieval task

Given an input audio, our model predictor has been trained to output a latent representation of a stem such that this stem would fit well with the input audio. To evaluate the performance of our model, we construct a retrieval task in which, given an existing music track, the model should be able to predict the representation of one stem given the mix of the others.

4.1.1 Experimental setup

For evaluation, we used the MUSDB18 dataset [32], which contains $N = 150$ tracks $\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N)}$, each track $\mathbf{x}^{(n)}$ being composed of $S = 4$ stems $\mathbf{x}_1^{(n)}, \dots, \mathbf{x}_S^{(n)}$ (vocals, bass, drums, other). This allows a total of $N \times S = 600$ runs. For any individual stem $\mathbf{x}_s^{(n)}$, define $\mathbf{x}_{-s}^{(n)}$ the mix containing all stems from $\mathbf{x}^{(n)}$ except $\mathbf{x}_s^{(n)}$. We aim to predict the embedding of the individual stem $\mathbf{x}_s^{(n)}$ from the one of the mix $\mathbf{x}_{-s}^{(n)}$. We compute and average (over time) the patch-wise representations of all stems $\mathbf{x}_s^{(n)}$. It gives us a *reference set* $\mathbf{Z} = \{\mathbf{z}_s^{(n)}\}$, with $\mathbf{z}_s^{(n)}$ being the embedding of $\mathbf{x}_s^{(n)}$. Then, we encode all mixes $\mathbf{x}_{-s}^{(n)}$, pass the resulting representations through the predictor conditioned on s , and average (over time and frequency) the result to get a *query embedding* $\mathbf{q}_s^{(n)}$. In other words, $\mathbf{q}_s^{(n)}$ is the prediction of (the embedding of) the missing instrument $\mathbf{x}_s^{(n)}$ from the remaining ones $\mathbf{x}_{-s}^{(n)}$. We therefore test if the actual embedding $\mathbf{z}_s^{(n)}$ is among the nearest neighbors of $\mathbf{q}_s^{(n)}$ in the reference set \mathbf{Z} .

Metrics. We measure the model performance using two metrics. The *Recall at K* ($R@K$) measures the proportion of relevant items successfully retrieved among the top K nearest neighbors. We consider here $K \in \{1, 5, 10\}$.

The *Normalized Rank* [5] of a query $\mathbf{q}_s^{(n)}$ is defined as the rank of the ground-truth $\mathbf{z}_s^{(n)}$ in the sorted list of distances $\{d(\mathbf{q}_s^{(n)}, \mathbf{z})\}_{\mathbf{z} \in \mathbf{Z}}$, normalized by the length of the list (here 600) to get a value in $[0, 1]$. For example, a mean Normalized Rank of 5% means that the actual embedding $\mathbf{z}_s^{(n)}$ is, on average, within the 5% nearest neighbors from the prediction $\mathbf{q}_s^{(n)}$. For each model, we report mean and median Normalized Ranks.

4.1.2 Results

The results are shown in Table 1 under the row "MLP w/ cond." Our model achieves a $R@1$ of 33%, and in half of the cases, the correct stem is within the top 0.5% of nearest neighbors (median Normalized Rank is 0.5%). Moreover, the median rank consistently outperforms the mean rank, indicating the presence of outliers with very high ranks.

Table 1. Influence of the design of the predictor on the retrieval performances. All metrics are in percentages.

Model	R@1	Recall \uparrow		Normalized Rank \downarrow	
		R@5	R@10	mean	median
MLP w/ cond.	33.0	63.2	76.2	2.0	0.5
MLP w/o cond.	28.2	58.0	69.2	3.3	0.7
Transformer	5.2	17.5	25.7	12.1	6.0
AutoMashupper	1.0	8.8	15.5	29.1	19.5

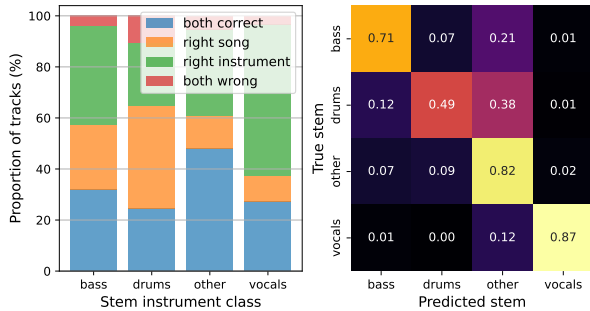


Figure 2. Analysis of the closest embedding z^* for all queries q from the MUSDB18 dataset [32]. Left: Categories of failures for each instrument (same song but wrong instrument, the opposite, or both wrong). Right: confusion matrix between conditioning instruments and retrieved instruments.

We also include in Table 1 results for scenarios without predictor conditioning during training (row "MLP w/o cond.") and when using a Transformer instead of an MLP for the predictor. In both cases, the performance drops substantially, emphasizing the importance of conditioning for the retrieval task. When using an MLP instead of a Transformer, the encoder must capture global information because the MLP cannot infer it, which leads to more informative embeddings.

Finally, we compare our model with AutoMashupper [1], which is, to the best of our knowledge, the only openly available work on compatibility estimation. We use their "mashability" measure as a similarity metric to compute the retrieval performances. Note that this metric involves beat tracking and chord detection, making it unsuited for vocals and drum stems, respectively. Therefore, the performance of this method on the retrieval task is relatively weak.

4.1.3 Influence of the instrument class

To get a better understanding of the failure modes of our model and the disparities between the different instruments, we study the nearest neighbor $z^* = \arg \min_{z \in Z} d(q, z)$ for all queries q from MUSDB18. This analysis, detailed in Figure 2 (left), categorizes z^* into four groups: "both correct" where the model predicts the correct instrument from the correct song, "right instrument" where the correct instrument is predicted but from a different song, "right song" where the model predicts the wrong instrument class but from the correct song, and "both wrong". Additionally, Figure 2 (right) displays

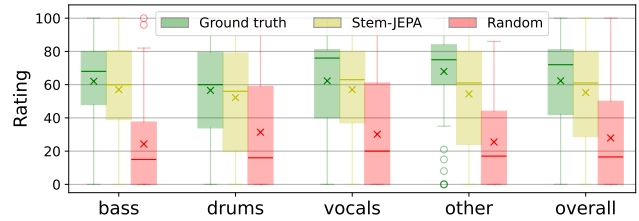


Figure 3. Box plot of the listening test for the different instrument classes. The \times represents the mean of the data.

a confusion matrix for the instruments.

A noticeable result is that the retrieval performances vary a lot between the different instruments, especially between "drums" ($R@1 \approx 25\%$) and "other" ($R@1 \approx 45\%$).³ A plausible explanation is that there are simply more possible candidate drum patterns that actually fit a given mix, resulting in closer neighbors within which it is harder to detect the ground truth.

Additionally, we can see that the "both wrong" scenario is quite uncommon. However, for bass and drums in particular, we predict another instrument (but for the correct song) in more than 25% of the cases. The confusion matrix shows that the category that mostly causes this failure is "other". A reason is probably that "other" is a broad and ill-defined set of instruments that could arguably overlap with bass or drums (e.g., choirs, synth bass, xylophone...).

4.2 User study

In section 4.1, we utilize the compatibility of a mix and a stem from the same song to assess the retrieval performance of our model. However, it is plausible that the dataset also includes compatible stems originally part of different songs. To evaluate our model's ability to retrieve these compatible yet non-original stems, we conduct an online listening test, focusing on retrieving instruments that are not present in the query mix (green segment in Figure 2).

For each trial, the user first listens to a query mix with one missing stem, followed (in random order) by the actual missing stem, the one retrieved by our system, and a random one, but with the same instrument class as $x_s^{(n)}$. They are then asked to rate (from 0 to 100) the three proposed stems' compatibility with the reference mix.⁴

The mixes and stems are 16-second chunks from the MUSDB18 dataset [32], randomly cropped to 10 seconds during the test to prevent listeners from relying on temporal alignment for rating. We conduct our study on the Go Listen platform [33]. Our test comprises 60 trials, and each user has to answer 12 of them (3 for each instrument class). We had 23 participants, 20 of whom had musical experience (11 for at least 10 years).

Results. The listening test results are depicted in Figure 3. While the ratings for the stems retrieved by our model

³ The proportion of "both correct" samples is exactly the Recall at 1.

⁴ Since we already test temporal alignment and tonality in sections 4.3 and 4.4 respectively, participants are explicitly instructed to rather concentrate on genre, timbre, and playing/singing style in this study.

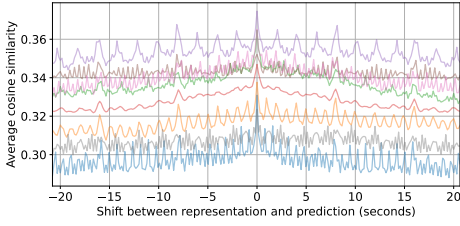


Figure 4. Average pairwise cosine similarity between embeddings and predictions across various temporal shifts. Each curve corresponds to a different track.

are slightly lower than those for the ground truth on average, they are substantially higher (approximately double) than the ratings of random samples. This highlights the ability of Stem-JEPA to retrieve stems compatible with the context mix.

However, we find some disparities between instrument classes. For example, the ratings for drums between the ground truth and our model’s suggestions are very close, whereas they are more different for the “other” category. Also, the variance of ratings is higher in “drums,” hinting at a generally higher compatibility of drums with any context. Finally, the length of the whiskers and the difference between the mean and median reveal significant disparities between users and samples, indicating the high subjectivity and difficulty of musical compatibility estimation.

4.3 Stem alignment analysis

In this section, we assess the model’s ability to evaluate the alignment between stems and mixes by temporally shifting them relative to each other. Our primary metric for this evaluation is the cosine similarity between learned embeddings and their predictions at various offsets, reflecting the local temporal features captured by the model.

Contrary to our previous approach that utilized embeddings averaged over time, here we retain the temporal sequence of the embeddings. We concatenate embeddings in the frequency dimension and stack them in the time dimension, maintaining a resolution of one embedding per 160 milliseconds of audio. We denote the representation of the i -th patch in stem $\mathbf{x}_s^{(n)}$ as $\mathbf{z}_s^{(n)}[i]$ and its corresponding predicted output conditioned on the mix $\mathbf{x}_{-s}^{(n)}$ as $\mathbf{q}_s^{(n)}[i]$.

We evaluate the fidelity of these embeddings by examining how the cosine similarity between $\mathbf{z}_s^{(n)}[i]$ and $\mathbf{q}_s^{(n)}[(i+j)\%M]$ evolves with varying j , the temporal offset. The formulation is given by:

$$s(\mathbf{z}, \mathbf{q}, j) = \frac{1}{MS} \sum_{s=1}^S \sum_{i=1}^M \langle \mathbf{z}_s[i], \mathbf{q}_s[(i+j)\%M] \rangle \quad (3)$$

where M is the total number of embeddings in sequence \mathbf{z} , S is the number of stems, and j represents the shift index.

The local nature of the information captured by the embeddings is reflected in how the cosine similarity $s(\mathbf{z}, \mathbf{q}, j)$ changes with different temporal offsets j . Specifically, if the embeddings predominantly contained global information, s would remain relatively constant across shifts. Con-

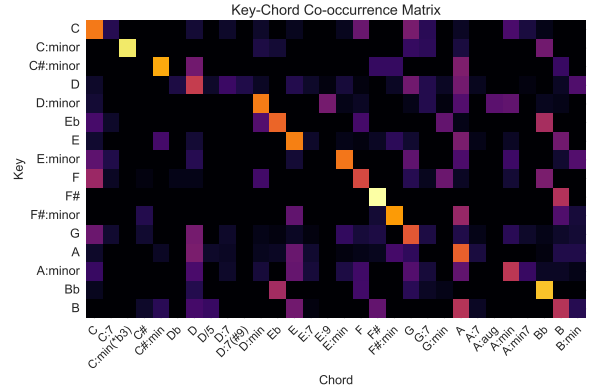


Figure 5. Key/Chord co-occurrence matrix between segments within the same clusters.

versely, a sharp peak in similarity at $j = 0$, followed by a rapid decrease, suggests that the embeddings are rich in local information and less information is shared between adjacent frames.

From our analysis of tracks from the MUSDB18 dataset (8 of them being displayed in Figure 4), we first observe that $s(\mathbf{z}, \mathbf{q}, j)$ always remains relatively high⁵, which indicates that the embeddings contain global information. We, however, observe a peak at $j = 0$, underlining the presence of local details that are temporally aligned.

We also observe periodic patterns in the curves, highlighting the model’s capacity to capture temporal structures (e.g., beats and bars). Finally, we observe smaller peaks every 8 seconds, the duration of the chunks used for computing the embeddings, which implies that embeddings also capture global position information. This behavior could potentially be avoided by replacing the absolute positional encodings in our encoder with other variants.

An interactive version of Figure 4 with audio examples is provided on the accompaniment website.⁶

4.4 Musical plausibility

We utilize key and chord annotations from Isophonics⁷ for 174 Beatles songs to label all patch embeddings. We then conduct k -means clustering on the latent space with $k = 32$ clusters. Within each cluster, we calculate the co-occurrence of all key and chord pairs and aggregate these counts across all clusters. To visualize these relationships, we display in Figure 5 a co-occurrence matrix for the keys and chords that appear in the top 80 most frequent combinations, considering all possible pairs for counting, not just the most common ones.

The matrix reveals that pairs close in the latent space often share significant musical relevance. The highest occurrences typically connect a key with its tonic (e.g., E/E), and prominently with its subdominant and dominant (e.g., C/F, G or D/G, A). Such patterns indicate that the embeddings capture meaningful tonal relationships.

⁵ As a reference, the average cosine similarity between random representations and predictions is approximately 0.17.

⁶ <https://sonycslparis.github.io/Stem-JEPA>

⁷ <http://isophonics.net/>

Table 2. Datasets used for downstream tasks.

Dataset	classes	Task
Giantsteps (GS) [36]	24	Key detection
GTZAN [37]	10	Genre classification
MagnaTagATune (MTT) [38]	50	Tagging
NSynth [39]	11	Instr. classification

4.5 Benchmark on downstream tasks

Lastly, we investigate the musical features encoded in the representations learned by our model. We hypothesize that the encoder captures shared musical information among different stems of the same track, such as rhythm or harmony, to aid the predictor. To verify this, we evaluate it on several downstream classification tasks, a standard protocol for representation learning methods [9, 12, 34, 35].

4.5.1 Experimental setup

Our experimental setup follows the constrained track of the MARBLE benchmark [35]. Each audio sample is processed by the *frozen* encoder, and its patch-wise outputs are concatenated and averaged along frequency and time dimensions to produce a 3840-dimensional global embedding, following [9]. These embeddings are passed through an MLP with 512 hidden units and a softmax layer, which is trained by minimizing the cross-entropy between the predicted distribution and the ground truth labels.

Downstream tasks. To validate our hypothesis, we focus on global musical features that are shared among the different stems of a track, namely tagging, key, and genre estimation. Additionally, we include an instrument classification task to observe whether the encoder preserves stem-specific information. For facilitating comparisons to existing work, we also pick our downstream tasks from the MARBLE benchmark [35]. The full list of datasets and associated tasks is depicted in Table 2.

Baselines. We compare our model to two variants: one trained with a Transformer as predictor, and one without conditioning, as in section 4.1. In addition, we include the two top-performing models from [35] in the considered tasks, namely MULE [12] and Jukebox-5B [40] as references. MULE [12] is an SSL model based on SF-NFNet-F0 [41] trained by contrastive learning on the MusicSet dataset (117k hours), while Jukebox [40] is a huge music generation model trained using codified audio language modeling on 1.2 million songs.

For a more in-depth description of the hyperparameters, datasets, tasks, and corresponding metrics, we refer the reader to [35].

4.5.2 Results

The performances of our model on downstream tasks are provided in Table 3. First, we observe that the choice of the predictor used for training, while extremely influencing for retrieval tasks, has little effect on the downstream performances of our encoder, apart from key detection on Giantsteps, for which the model trained with a Transformer predictor clearly outperforms the others. The

Table 3. Influence of the predictor architecture on the performances of Stem-JEPA on various downstream tasks, and comparison with existing baselines.

Model	GS	GTZAN	MTT		NSynth
	Acc ^{refined}	Acc	ROC	AP	Acc
MLP w/ cond.	40.2	68.6	89.9	42.8	73.5
MLP w/o cond.	36.8	72.5	90.1	42.9	75.0
Transformer	46.0	68.1	90.0	42.7	73.3
MULE [12]	64.9	75.5	91.2	40.1	74.6
Jukebox [42]	63.8	77.9	91.4	40.6	70.4

performances on NSynth also reveal that our model does not only capture information shared between stems but also stem-specific features. Surprisingly, this holds even without conditioning the predictor during training, and more generally, not conditioning the predictor improves performance on most downstream tasks.

We also compare our model to state-of-the-art works in music representation learning. Our performances are on par with baselines for two tasks (MTT and NSynth) but significantly lower on Giantsteps and GTZAN, despite being much better than random guessing. Considering the limited quantity of training data compared to the baselines (about 100 times less), these results suggest that our method is promising for music representation learning but that further efforts have to be made to make it competitive with current state-of-the-art approaches in this field.

5. CONCLUSION

In this study, we introduce a novel SSL paradigm based on stem prediction for musical stem compatibility estimation through the prism of representation learning. Our results show promising performances for retrieval applications and also indicate that the learned representations are localized, suggesting that they could also be valuable for music generation and possibly automatic arrangement. Additionally, these representations are musically meaningful and demonstrate linear separability for various Music Information Retrieval tasks.

Moreover, our model is, to the best of our knowledge, the first use of the predictor component of Joint-Embedding Predictive Architectures (JEPAs) during inference. Employing JEPAs to model compatibility instead of similarity, with appropriate conditioning, may open up possibilities in various fields beyond music.

Nevertheless, our study is not without its limitations. In particular, self-supervised learning usually benefits from very large corpora of training data; however, accessing large datasets of separated stems is challenging, though advancements in source separation technology may alleviate some of these issues. Finally, restricting the analysis to four instruments, while standard in source separation, currently limits the generalizability of our findings. Ideally, extending the predictor to accommodate any instrument would prevent the failure cases illustrated in section 4.1.3 and enhance the model’s utility, representing an exciting direction for future research.

6. ACKNOWLEDGMENTS

This work has been funded by the ANRT CIFRE convention n°2021/1537 and Sony France. This work was granted access to the HPC/AI resources of IDRIS under the allocation 2022-AD011013842 made by GENCI. We would like to thank Cyran Aouameur and Marco Comunità for their helpful suggestions. Finally, we would like to thank the reviewers and meta-reviewer for their valuable comments.

7. REFERENCES

- [1] M. E. P. Davies, P. Hamel, K. Yoshii, and M. Goto, "Automashupper: automatic creation of multi-song music mashups," *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, vol. 22, no. 12, p. 1726–1737, dec 2014.
- [2] C.-L. Lee, Y.-T. Lin, Z.-R. Yao, F.-Y. Lee, and J.-L. Wu, "Automatic mashup creation by considering both vertical and horizontal mashabilities," in *International Society for Music Information Retrieval Conference*, 2015.
- [3] J. Huang, J.-C. Wang, J. B. Smith, X. Song, and Y. Wang, "Modeling the compatibility of stem tracks to generate music mashups," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 1, 2021, pp. 187–195.
- [4] B. Y. Chen, J. B. Smith, and Y. H. Yang, "Neural Loop Combiner: Neural Network Models for Assessing the Compatibility of Loops," in *Proceedings of the 21st International Society for Music Information Retrieval Conference, ISMIR 2020*. International Society for Music Information Retrieval, aug 2020, pp. 424–431.
- [5] S. Lattner, "Samplematch: Drum sample retrieval by musical context," in *Proceedings of the 23rd International Society for Music Information Retrieval Conference, ISMIR 2022, Bengaluru, India, December 4-8, 2022, 2022*, pp. 781–788.
- [6] J. Nistal, M. Pasini, C. Aouameur, M. Grachten, and S. Lattner, "Diff-a-riff: Musical accompaniment co-creation via latent diffusion models," 2024.
- [7] J. D. Parker, J. Spijkervet, K. Kosta, F. Yesiler, B. Kuznetsov, J.-C. Wang, M. Avent, J. Chen, and D. Le, "Stemgen: A music generation model that listens," 2024.
- [8] M. Assran, Q. Duval, I. Misra, P. Bojanowski, P. Vincent, M. G. Rabbat, Y. LeCun, and N. Ballas, "Self-supervised learning from images with a joint-embedding predictive architecture," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*. IEEE, 2023, pp. 15 619–15 629.
- [9] D. Niizumi, D. Takeuchi, Y. Ohishi, N. Harada, and K. Kashino, "Masked modeling duo: Learning representations by encouraging both networks to model the input," in *IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP 2023, Rhodes Island, Greece, June 4-10, 2023*. IEEE, 2023, pp. 1–5.
- [10] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *37th International Conference on Machine Learning, ICML 2020*, vol. PartF16814. International Machine Learning Society (IMLS), feb 2020, pp. 1575–1585.
- [11] D. Niizumi, D. Takeuchi, Y. Ohishi, N. Harada, and K. Kashino, "BYOL for Audio: Exploring Pre-Trained General-Purpose Audio Representations," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, pp. 1–15, apr 2022.
- [12] M. C. McCallum, F. Korzeniowski, S. Oramas, F. Gouyon, and A. F. Ehmann, "Supervised and Un-supervised Learning of Audio Representations for Music Understanding," in *Proceedings of the 23rd International Society for Music Information Retrieval Conference, ISMIR 2022*, oct 2022.
- [13] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, vol. 1, pp. 4171–4186, oct 2018.
- [14] A. Saeed, D. Grangier, and N. Zeghidour, "Contrastive learning of general-purpose audio representations," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, vol. 2021-June. Institute of Electrical and Electronics Engineers Inc., oct 2021, pp. 3875–3879.
- [15] J. Spijkervet and J. A. Burgoyne, "Contrastive Learning of Musical Representations," *Proceedings of the 22nd International Society for Music Information Retrieval Conference, ISMIR 2021*, mar 2021.
- [16] A. Bardes, J. Ponce, and Y. LeCun, "VICReg: Variance-Invariance-Covariance Regularization for Self-Supervised Learning," in *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, may 2022.
- [17] T. Wang and P. Isola, "Understanding contrastive representation learning through alignment and uniformity on the hypersphere," in *37th International Conference on Machine Learning, ICML 2020*, vol. PartF16814. International Machine Learning Society (IMLS), may 2020, pp. 9871–9881.
- [18] K. He, X. Chen, S. Xie, Y. Li, P. Dollar, and R. Girshick, "Masked Autoencoders Are Scalable Vision

- Learners,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2022-June. IEEE Computer Society, nov 2022, pp. 15 979–15 988.
- [19] J. B. Grill, F. Strub, F. Althé, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar, B. Piot, K. Kavukcuoglu, R. Munos, and M. Valko, “Bootstrap your own latent a new approach to self-supervised learning,” in *Advances in Neural Information Processing Systems*, vol. 2020-Decem. Neural information processing systems foundation, jun 2020.
- [20] X. Li and X. Li, “ATST: Audio Representation Learning with Teacher-Student Transformer,” in *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, vol. 2022-Sept. International Speech Communication Association, apr 2022, pp. 4172–4176.
- [21] A. Baevski, W. Hsu, Q. Xu, A. Babu, J. Gu, and M. Auli, “data2vec: A general framework for self-supervised learning in speech, vision and language,” in *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, ser. Proceedings of Machine Learning Research, vol. 162. PMLR, 2022, pp. 1298–1312.
- [22] X. Chen and K. He, “Exploring simple Siamese representation learning,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, nov 2021, pp. 15 745–15 753.
- [23] Y. Tian, X. Chen, and S. Ganguli, “Understanding self-supervised learning dynamics without contrastive pairs,” in *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, ser. Proceedings of Machine Learning Research, vol. 139. PMLR, 2021, pp. 10 268–10 278.
- [24] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, “wav2vec 2.0: A framework for self-supervised learning of speech representations,” in *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.
- [25] Y.-A. Chung, Y. Zhang, W. Han, C.-C. Chiu, J. Qin, R. Pang, and Y. Wu, “W2v-bert: Combining contrastive learning and masked language modeling for self-supervised speech pre-training,” in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2021, pp. 244–250.
- [26] P. Huang, H. Xu, J. Li, A. Baevski, M. Auli, W. Galuba, F. Metze, and C. Feichtenhofer, “Masked autoencoders that listen,” in *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.
- [27] E. Fonseca, A. Jansen, D. P. Ellis, S. Wisdom, M. Tagliasacchi, J. R. Hershey, M. Plakal, S. Hershey, R. C. Moore, and X. Serra, “Self-supervised learning from automatically separated sound scenes,” in *2021 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2021, pp. 251–255.
- [28] C. Garoufis, A. Zlatintsi, and P. Maragos, “Multi-Source Contrastive Learning From Musical Audio,” in *Proceedings of the Sound and Music Computing Conferences*, vol. 2023-June. Sound and Music Computing Network, feb 2023, pp. 162–169.
- [29] D. Desblancs, V. Lostanlen, and R. Hennequin, “Zero-note samba: Self-supervised beat tracking,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.
- [30] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, “An image is worth 16x16 words: Transformers for image recognition at scale,” in *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.
- [31] I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” in *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.
- [32] Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimilakis, and R. Bittner, “The MUSDB18 corpus for music separation,” Dec. 2017.
- [33] D. Barry, Q. Zhang, P. W. Sun, and A. Hines, “Go listen: An end-to-end online listening test platform,” *Journal of Open Research Software*, 2021.
- [34] J. Turian, J. Shier, H. R. Khan, B. Raj, B. W. Schuller, C. J. Steinmetz, C. Malloy, G. Tzanetakis, G. Velarde, K. McNally, M. Henry, N. Pinto, C. Noufi, C. Clough, D. Herremans, E. Fonseca, J. H. Engel, J. Salamon, P. Esling, P. Manocha, S. Watanabe, Z. Jin, and Y. Bisk, “HEAR: holistic evaluation of audio representations,” in *NeurIPS 2021 Competitions and Demonstrations Track, 6-14 December 2021, Online*, ser. Proceedings of Machine Learning Research, vol. 176. PMLR, 2021, pp. 125–145.
- [35] R. Yuan, Y. Ma, Y. Li, G. Zhang, X. Chen, H. Yin, L. Zhuo, Y. Liu, J. Huang, Z. Tian, B. Deng, N. Wang, C. Lin, E. Benetos, A. Ragni, N. Gyenge, R. B. Dannenberg, W. Chen, G. Xia, W. Xue, S. Liu, S. Wang, R. Liu, Y. Guo, and J. Fu, “MARBLE: music audio representation benchmark for universal evaluation,” in *Advances in Neural Information Processing Systems 36:*

Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023, 2023.

- [36] P. Knees, Á. Faraldo, P. Herrera, R. Vogl, S. Böck, F. Hörschläger, and M. L. Goff, “Two data sets for tempo estimation and key detection in electronic dance music annotated from user corrections,” in *Proceedings of the 16th International Society for Music Information Retrieval Conference, ISMIR 2015, Málaga, Spain, October 26-30, 2015*, 2015, pp. 364–370.
- [37] G. Tzanetakis and P. R. Cook, “Musical genre classification of audio signals,” *IEEE Trans. Speech Audio Process.*, vol. 10, no. 5, pp. 293–302, 2002.
- [38] E. Law, K. West, M. I. Mandel, M. Bay, and J. S. Downie, “Evaluation of algorithms using games: The case of music tagging,” in *Proceedings of the 10th International Society for Music Information Retrieval Conference, ISMIR 2009, Kobe International Conference Center, Kobe, Japan, October 26-30, 2009*. International Society for Music Information Retrieval, 2009, pp. 387–392.
- [39] J. H. Engel, C. Resnick, A. Roberts, S. Dieleman, M. Norouzi, D. Eck, and K. Simonyan, “Neural audio synthesis of musical notes with wavenet autoencoders,” in *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, ser. Proceedings of Machine Learning Research, vol. 70. PMLR, 2017, pp. 1068–1077.
- [40] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, and I. Sutskever, “Jukebox: A generative model for music,” *CoRR*, vol. abs/2005.00341, 2020.
- [41] L. Wang, P. Luc, Y. Wu, A. Recasens, L. Smaira, A. Brock, A. Jaegle, J. Alayrac, S. Dieleman, J. Carreira, and A. van den Oord, “Towards learning universal audio representations,” in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2022, Virtual and Singapore, 23-27 May 2022*. IEEE, 2022, pp. 4593–4597.
- [42] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, and I. Sutskever, “Jukebox: A generative model for music,” apr 2020.

AUDIO PROMPT ADAPTER: UNLEASHING MUSIC EDITING ABILITIES FOR TEXT-TO-MUSIC WITH LIGHTWEIGHT FINETUNING

Fang-Duo Tsai¹ Shih-Lun Wu² Haven Kim³
Bo-Yu Chen¹ Hao-Chung Cheng¹ Yi-Hsuan Yang¹

¹ National Taiwan University ² Carnegie Mellon University ³ University of California San Diego

r12942150@ntu.edu.tw, shihlunw@andrew.cmu.edu, khaven@ucsd.edu
bernie40916@gmail.com, haochung@ntu.edu.tw, yhyangtw@ntu.edu.tw

ABSTRACT

Text-to-music models allow users to generate nearly realistic musical audio with textual commands. However, *editing* music audios remains challenging due to the conflicting desiderata of performing fine-grained alterations on the audio while maintaining a simple user interface. To address this challenge, we propose *Audio Prompt Adapter* (or AP-Adapter), a lightweight addition to pretrained text-to-music models. We utilize AudioMAE to extract features from the input audio, and construct attention-based adapters to feed these features into the internal layers of AudioLDM2, a diffusion-based text-to-music model. With 22M trainable parameters, AP-Adapter empowers users to harness both global (e.g., genre and timbre) and local (e.g., melody) aspects of music, using the original audio and a short text as inputs. Through objective and subjective studies, we evaluate AP-Adapter on three tasks: timbre transfer, genre transfer, and accompaniment generation. Additionally, we demonstrate its effectiveness on out-of-domain audios containing unseen instruments during training.

1. INTRODUCTION

Advancements in *text-to-music generation* have made it possible for users to create music audio signals from simple textual descriptions [1–4]. To improve the control over the generated music beyond textual input, several newer models have been proposed, using additional conditioning signals indicating the intended global or time-varying musical attributes such as melody, chord progression, rhythm, or loudness for generation [5–9] (see Section 2 for a brief review). Such controllability is important for musicians, practitioners, as well as common users in the human-AI co-creation process [10, 11].

However, one area that remains challenging, which we refer to as *text-to-music editing* below, is the precise editing

of a piece of music, provided by a user as an *audio input* x alongside the *text input* y for the textual prompts. The goal here for the model is to create an “edited” version of the input music, denoted as \tilde{x} , according to the text input. This is a crucial capability for users who wish to refine either an original or machine-generated music without compromising its musicality and audio quality, while keeping the simplicity of text-based human-computer interaction. Namely, the desired properties of the output \tilde{x} are:

- **Transferability:** \tilde{x} should reflect what y specifies, e.g., timbre, genre, instrumentation, or mood.
- **Fidelity:** \tilde{x} should retain all other musical content in x that y does not concern, e.g., melody and rhythm.

While a text-to-music generation model takes in general only the text input y and generates music freely, a text-to-music editing model takes both audio and text inputs x and y . The primary challenge arises from the conflicting goals of maintaining high fidelity to the input audio x while incorporating specific changes dictated by textual commands y . As we review in Section 2, existing methods [14–16] either lack the granularity needed for detailed audio manipulation or need complex prompt engineering that detracts from user accessibility or requires iterative refinements.

A secondary challenge arises from the large number of trainable parameters needed for models to achieve high musical quality and diversity (e.g., MusicGen-medium [5] has 1.5B parameters). Without much computational resource, it is more feasible to treat existing models as “foundation models” and finetune them to fulfill specific needs, instead of training a model from scratch [17].

In view of these challenges, we propose in this paper the *Audio Prompt Adapter* (or, AP-Adapter for short), a novel approach inspired by the Image Prompt Adapter (IP-Adapter) [18] from the neighboring field of text-to-image editing. This lightweight (22M parameters), attention-based module integrates seamlessly with existing text-to-music generation models, specifically leveraging the pre-trained AudioLDM2 model [12] enhanced by the AudioMAE encoder [13] to extract audio features. Our method uniquely combines text and audio inputs through decoupled cross-attention layers, allowing precise control in the generation process. After training the AP-adapter with a single NVIDIA RTX 3090, our method can zero-shot edit a given



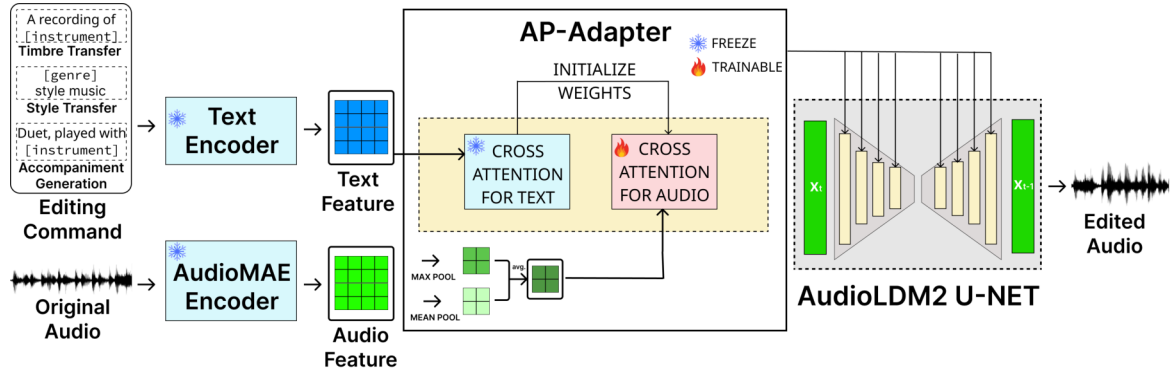


Figure 1: Our AP-Adapter is an add-on to AudioLDM2 [12]. Users provide an original audio to AudioMAE [13] to extract audio features, and an editing command to the text encoder. The decoupled audio and text cross-attention layers of AP-Adapter contribute to the **fidelity** with the input audio and **transferability** of the editing command in the edited audio.

audio prompt according to the text prompt.

Our AP-Adapter offers great improvements over some baseline models by enabling detailed and context-sensitive audio manipulations, achieving a balance between fidelity and the transferability effects dictated by user inputs. Our experiments across timbre transfer, genre transfer, and accompaniment generation tasks demonstrate the effectiveness of our approach in handling diverse and complex editing requirements. In short, our key contributions are:

- Proposing a framework that equips an audio input modality for a pre-trained text-to-music generation model.
- Performing zero-shot music editing with a lightweight adapter, which permits flexible balance of the effects of the text and audio inputs.
- Demonstrating three tasks: timbre transfer, genre transfer, accompaniment generation, and discussing the impact of tunable hyperparameters.

We provide audio examples in our demo website.¹ We also share source code and model checkpoints on GitHub.²

2. RELATED WORK

Generating desired music from text prompts alone is complex and often requires intricate prompt engineering. Mustango [7] enhanced prompts with information-rich captions specifying chords, beats, tempo, and key. MusicGen [5] conditioned music generation on melodies by extracting chroma features [19] and inputting them with the text prompt into a Transformer model. Coco-Mulla [6] and MusiConGen [9] extended MusicGen by adding time-varying chord- and rhythm-related controls. Music ControlNet [8] incorporated time-varying conditions like melody, rhythm, and dynamics for diffusion-based text-to-music models. These methods utilize low-level features to guide generation but do not take reference audio as input, limiting their potential for editing existing audio tracks.

Recently, several music editing methods were proposed. InstructME [14] uses a VAE and a chord-conditioned diffusion model for music editing but requires a large dataset

of audio files with multiple instrumental tracks and triplet data of text instructions, source music, and target music for supervised training. M²UGen [15] leverages large language models to understand and generate music across different modalities, supporting music editing via natural language, but it requires a three-step training process and complex preprocessing. MusicMagus [16] implements latent space manipulation during inference for music editing but requires an additional music captioning model and the InstructGPT LLM to address discrepancies between the text prompt distribution of AudioLDM2 and the music captioning model.

Compared to these methods, our AP-Adapter is more straightforward to train and can achieve multiple music editing tasks in a zero-shot manner.

3. BACKGROUND

3.1 Diffusion Model

Denising diffusion probabilistic models (DDPMs) [20], also known as diffusion models, are a class of generative models that approximates some distribution $p(\mathbf{x})$ via denoising through a sequence of $T - 1$ latent variables:

$$p_{\theta}(\mathbf{x}) = \int \left[\prod_{t=1}^T p_{\theta}(\mathbf{x}_{t-1} | \mathbf{x}_t) \right] p(\mathbf{x}_T) d\mathbf{x}_{1:T}, \quad (1)$$

where θ is the set of learnable parameters, $\mathbf{x}_0 := \mathbf{x}$, and $p(\mathbf{x}_T) := \mathcal{N}(0, \mathbf{I})$ (i.e., an uninformative Gaussian prior). To train the model, we run forward diffusion: sample some data point $\mathbf{x} \sim p(\mathbf{x})$ and some $t \in [1, T]$, and add noise $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ to \mathbf{x} to produce a noised data point $\mathbf{x}_t := \sqrt{\beta_t} \mathbf{x} + \sqrt{1 - \beta_t} \epsilon$, where β_t is the pre-defined noise level for step t . The model is asked to perform backward diffusion, namely, to recover the added noise via the objective $\min_{\theta} \mathbb{E}_{\mathbf{x}, \epsilon, t} [\|\epsilon - \epsilon_{\theta}(\mathbf{x}_t, t)\|_2^2]$, where $\epsilon_{\theta}(\cdot)$ is the model’s prediction, that is equivalent to maximizing the evidence lower bound (ELBO) of $p_{\theta}(\mathbf{x})$. During inference, we start from an $\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$ and iteratively remove the predicted noise $\epsilon_{\theta}(\mathbf{x}_t, t)$ to generate data. Song *et al.* [21] offered a crucial interpretation that each denoising step can be seen as ascending along $\nabla_{\mathbf{x}} \log p_{\theta}(\mathbf{x})$, also known as the *score* of $p_{\theta}(\mathbf{x})$. Any input condition \mathbf{y} can be incorporated into a

¹ Demo: <https://rebrand.ly/AP-adapter>

² Code: <https://github.com/fundwotsai2001/AP-adapter>

diffusion model by injecting embeddings of \mathbf{y} via, for example, cross-attention [22], thereby modeling $p_\theta(\mathbf{x} | \mathbf{y})$ (and $\nabla_{\mathbf{x}} \log p_\theta(\mathbf{x} | \mathbf{y})$). To reduce memory footprint and accelerate training/inference, latent diffusion models (LDMs) [22] proposed to first compress data points \mathbf{x} into latent vectors using a variational autoencoder (VAE) [23], and then learn a diffusion model for the latent vectors.

3.2 AudioLDM2

We choose AudioLDM2 [12], a latent diffusion-based [22] text-to-audio model, as our pretrained backbone. To enable text control over generated audio, AudioLDM2 uses AudioMAE [13] to extract acoustic features, named the *language of audio* (LOA), from the target audio. LOA serves as the bridge between acoustic and text-centric semantic information—the text prompt is encoded by both the FLAN-T5 [24] language model and CLAP [25] text encoder (which has a joint audio-text embedding space), and then passed to a trainable GPT-2 [26] to approximate the LOA via a regression loss that aligns the semantic representations with LOA. The aligned text information is then fed into the U-Net [27] for diffusion process to influence the generation. We pick AudioLDM2 to be the backbone since the use of LOA likely promotes the affinity to accepting audio conditions, which is crucial to our fidelity goal.

3.3 Classifier-free Guidance

Classifier-free guidance (CFG) [28] is a simple yet effective inference-time method to enhance the input text condition’s influence, which is directly linked to our transferability goal. As mentioned in Sec. 3.1, diffusion models can predict both the unconditioned score $\nabla_{\mathbf{x}} \log p(\mathbf{x})$ and the conditioned score $\nabla_{\mathbf{x}} \log p(\mathbf{x} | \mathbf{y})$. In addition, by Bayes’ rule, we know that $p(\mathbf{x} | \mathbf{y}) \propto p(\mathbf{x})p(\mathbf{y} | \mathbf{x})$. As the goal is the amplify \mathbf{y} ’s influence, we define:

$$p_\lambda(\mathbf{x} | \mathbf{y}) \propto p(\mathbf{x})p(\mathbf{y} | \mathbf{x})^\lambda, \quad (2)$$

where λ is a knob, named *CFG scale*, that controls the strength of \mathbf{y} . Taking $(\nabla_{\mathbf{x}} \log)$ on both sides gives us:

$$\nabla_{\mathbf{x}} \log p_\lambda(\mathbf{x} | \mathbf{y}) = \lambda \nabla_{\mathbf{x}} \log p(\mathbf{y} | \mathbf{x}) + \nabla_{\mathbf{x}} \log p(\mathbf{x}). \quad (3)$$

Meanwhile, we can rearrange the Bayes’ rule terms to get:

$$\nabla_{\mathbf{x}} \log p(\mathbf{y} | \mathbf{x}) = \nabla_{\mathbf{x}} \log p(\mathbf{x} | \mathbf{y}) - \nabla_{\mathbf{x}} \log p(\mathbf{x}). \quad (4)$$

Note that a diffusion model can predict both RHS terms. Plugging Eqn. (4) into Eqn. (3), CFG performs

$$\begin{aligned} \nabla_{\mathbf{x}} \log p_\lambda(\mathbf{x} | \mathbf{y}) &= \nabla_{\mathbf{x}} \log p(\mathbf{x}) \\ &+ \lambda(\nabla_{\mathbf{x}} \log p(\mathbf{x} | \mathbf{y}) - \nabla_{\mathbf{x}} \log p(\mathbf{x})) \end{aligned} \quad (5)$$

at every inference iteration, where $\nabla_{\mathbf{x}} \log p(\mathbf{x})$ is obtained by inputting an empty string as \mathbf{y} .

4. PROPOSED AUDIO PROMPT ADAPTER

To effectively condition AudioLDM2 on the input audio and achieve our transferability and fidelity goals, our AP-Adapter adds two components to AudioLDM2: an audio

encoder to extract acoustic features, and decoupled cross-attention adapters to incorporate the acoustic features while maintaining text conditioning capability.

4.1 Audio Encoder and Feature Pooling

We adopt AudioMAE as the audio encoder, which is used by AudioLDM2 to produce the language of audio (LOA; see Section 3.2) during its training. In our pilot study, we find that using the LOA directly as the condition causes nearly verbatim reconstruction, i.e., information in the input audio is mostly retained. This is undesirable as it greatly limits transferability. To address this issue, we apply a combination of max and mean pooling on the LOA, and leave the pooling rate, which we denote by ω , tunable by the user to trade off between fidelity and transferability.

4.2 Decoupled Cross-attention Adapters

According to the analyses in [29, 30] performed on text-to-image diffusion models finetuned for image editing [31], the cross-attention layers, which allow interaction between text prompt and the diffusion process, undergo the most drastic changes during fine-tuning. Hence, we implement our AP-Adapter also as a set of cross-attention layers.

Recall that the audio and text prompts are transformed to internal features before interacting with the U-Net for diffusion. We define these features as:

$$\mathbf{c}_x := \text{Pool}(\text{AudioMAE}(\mathbf{x})) \quad (6)$$

$$\mathbf{c}_y := \text{GPT2}([\text{FlanT5}(\mathbf{y}); \text{CLAP}(\mathbf{y})]), \quad (7)$$

where \mathbf{c}_x and \mathbf{c}_y are the audio and text features respectively. The original AudioLDM2 incorporates the text feature into each U-Net layer via cross-attention:

$$\mathbf{z}_{\text{text}} := \text{Attention}(\mathbf{z}\mathbf{W}^{(a)}, \mathbf{c}_y\mathbf{W}^{(k)}, \mathbf{c}_y\mathbf{W}^{(v)}), \quad (8)$$

where \mathbf{z} is the U-Net’s internal feature, and $\mathbf{W}^{(a)}$, $\mathbf{W}^{(k)}$, $\mathbf{W}^{(v)}$ are learnable projections that respectively produce the cross-attention query, key, and values from \mathbf{z} or \mathbf{c}_y . We keep this cross-attention for text intact (i.e., frozen), anticipating it to satisfy transferability out of the box.

To incorporate the audio features for fidelity, we place a decoupled audio cross-attention layer as the adapter alongside each text cross-attention in a similar light to [18]:

$$\mathbf{z}_{\text{audio}} := \text{Attention}(\mathbf{z}\mathbf{W}^{(a)}, \mathbf{c}_x\mathbf{W}'^{(k)}, \mathbf{c}_x\mathbf{W}'^{(v)}), \quad (9)$$

where $\mathbf{W}'^{(k)}$ and $\mathbf{W}'^{(v)}$ are the newly introduced adapter weights. Since during AudioLDM2 training, the text feature \mathbf{c}_y is trained to mimic the LOA from AudioMAE, we initialize $\mathbf{W}'^{(k)}$ and $\mathbf{W}'^{(v)}$ respectively from $\mathbf{W}^{(k)}$ and $\mathbf{W}^{(v)}$ for all the cross-attention layers in the U-Net, and find that this significantly shortens our fine-tuning process compared to random initialization.

Finally, we obtain the final output of the decoupled text and audio cross-attentions via a weighted sum:

$$\mathbf{z}_{\text{fusion}} := \mathbf{z}_{\text{text}} + \alpha \mathbf{z}_{\text{audio}}, \quad (10)$$

where $\alpha \in \mathbb{R}$, named *AP scale*, is a hyperparameter that controls the strength of the audio prompt (fixed to $\alpha = 1$ during training), and z_{fusion} becomes the input of the subsequent U-Net layer. We expect z_{fusion} to capture the information mixture from audio and text prompts, inducing the model to generate plausible music that adheres to both.

4.3 Training

We freeze all the parameters in the pretrained AudioLDM2 and AudioMAE, except for the decoupled audio cross-attention adapters with 22M parameters. The loss function follows that of standard (latent) diffusion models:

$$\mathcal{L} = \mathbb{E}_{(x,y),\epsilon,t} \|\epsilon - \epsilon_{\theta}(x_t, c_x, c_y, t)\|_2^2, \quad (11)$$

where (x, y) are naturally existing paired audio and text, $\epsilon \sim \mathcal{N}(0, \mathbf{I})$, t is the diffusion step, x_t is the noised audio latent features, c_x, c_y are the extracted features from text and audio prompts (cf. Eqn. (6) and (7)), and $\epsilon_{\theta}(\cdot)$ is the model’s predicted noise. Minimizing \mathcal{L} is equivalent to maximizing the lower bound of $p(x | c_x, c_y)$. During training, we select the audio feature’s pooling rate ω from the set $\{1, 2, 4, 8\}$ uniformly at random, making the adapters recognize audio features with different resolutions, thereby allowing users to balance fidelity and transferability at inference. Additionally, we randomly dropout audio and text conditions, i.e., setting c_x to a zero matrix, and y to an empty string, to facilitate classifier-free guidance.

4.4 Inference

At inference, users are free to input any text prompt y as the editing command to achieve their desired edits, i.e. $x \rightarrow \tilde{x}$. In addition, following [32,33], we modify the unconditioned terms in Eqn. (5) using a negative text prompt y^- . Letting $c_{xy} := \{c_x, c_y\}$, our inference step is:

$$\begin{aligned} \nabla_{\tilde{x}} \log p_{\lambda}(\tilde{x} | c_{xy}, c_{y^-}) &= \nabla_{\tilde{x}} \log p(\tilde{x} | c_{y^-}) \\ &+ \lambda (\nabla_{\tilde{x}} \log p(\tilde{x} | c_{xy}) - \nabla_{\tilde{x}} \log p(\tilde{x} | c_{y^-})) \end{aligned} \quad (12)$$

We find that specifying y^- is an effective way to avoid unwanted properties in \tilde{x} , e.g., the original timbre for the timbre transfer task, or low-quality music in general.

5. EXPERIMENT SETUP

5.1 Dataset Preparation

For the training data of our AP-Adapter, due to our limited computation resource, we use 200K 10-second-long audios with text tags randomly sampled from AudioSet [34] (about 500 hours, or $\sim 10\%$ of the whole dataset).

For the audio input x used in evaluation, we compile two datasets: **in-domain** and **out-of-domain**, according to whether the AudioSet ontology includes the instrument.

- **In-domain:** We choose 8 common instruments: piano, violin, cello, flute, marimba, organ, harp and acoustic guitar. For each instrument, we manually download 5 high-quality monophonic audios from YouTube (i.e., 40 samples in total) and crop them each to 10 seconds.

- **Out-of-domain:** We collect a dataset of monophonic melodies played by ethnic instruments, including 2 *Chinese* instruments (collected by one of our co-authors) and 5 *Korean* instruments (downloaded from AIHub [35]). We use 5 audio samples for each instrument (35 audios in total), cropped to 10 seconds each. We note that these instruments are **not seen** during the training time.

Except for the Korean data which is not licensed outside of Korea, we share information to get the data on GitHub.

5.2 Evaluation Tasks

By varying the edit command y , we evaluate AP-Adapter on three music editing tasks:

- **Timbre transfer:** The model is expected to change a melody’s timbre to that of the target instrument, and keep all other contents unchanged. For this task, the editing command (y) is set to “a recording of a [target instrument] solo”. The negative prompt (y^-) is “a recording of the [original instrument] solo”. For in-domain input, the target is one of the other 7 in-domain instruments. For out-of-domain input, the target is one of the 8 in-domain instruments. We only use in-domain instruments as the target because our evaluation metrics CLAP [25] and FAD [36] (see Section 5.5) do not recognize the out-of-domain instruments.
- **Genre transfer:** We expect the genre (e.g., jazz and country) to change according to the text prompt, but we wish to retain most of the other content such as melody, rhythm and timbre. Here, we set $y :=$ “[target genre] style music”, and $y^- :=$ “low quality music”. Here, we target 8 genres: jazz, reggae, rock, metal, pop, hip-hop, disco, country.
- **Accompaniment generation:** We expect that all content in the input melody remains unchanged, but a new instrument is added to accompany the original audio in a pleasant-sounding and harmonic way. We set $y :=$ “Duet, played with [accomp instrument] accompaniment”, and $y^- :=$ “low quality music”. The [accomp instrument] is selected in the same way as the [target instrument] in the timbre transfer task.

We include these representative tasks which musicians may find useful for their daily workflow, but since y is free-form text, AP-Adapter has the potential for many other tasks.

5.3 Training and Inference Specifics

We use AudioLDM2-large (1.5B parameters), available on HuggingFace, as our backbone model, and only train our 22M-parameter adapters. Training is done on a single one RTX 3090 (24GB) for 35K steps with an effective batch size of 32. We use AdamW optimizer with fixed learning rate 10^{-4} and weight decay 10^{-2} . To enable CFG, we randomly dropout text and audio features with a 5% probability.

For inference, we choose the critical hyperparameters, i.e., pooling rate ω , AP scale α , and CFG scale λ , by exploring the transferability-fidelity tradeoff space as will be reported in Section 6.1. For timbre transfer and accompaniment generation, we select $\omega = 2$, $\alpha = 0.5$, $\lambda = 7.5$. For the

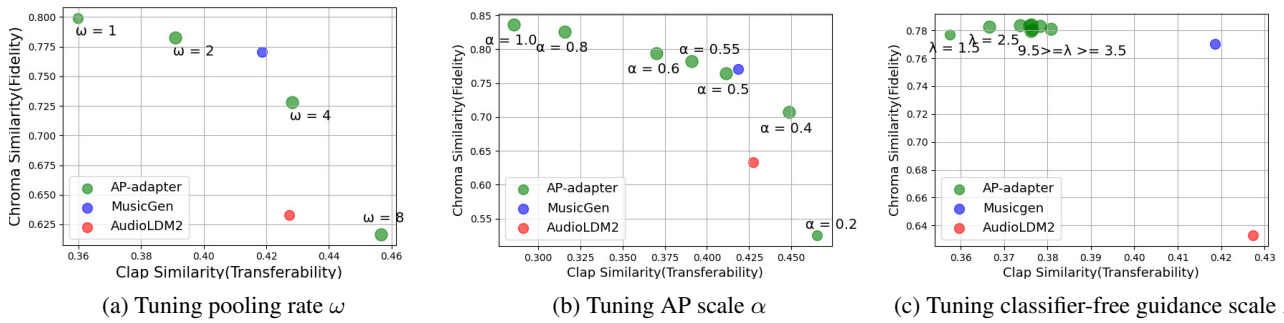


Figure 2: Transferability-fidelity tradeoff effects of different hyperparameters on the timbre transfer task. The hyperparameters are set to $\omega = 2$, $\alpha = 0.55$, and $\lambda = 7.5$ when they are not the hyperparameter of interest.

genre transfer, we select $\omega = 1$, $\alpha = 0.4$, $\lambda = 7.5$. Following AudioLDM2, we use 50 diffusion steps.

5.4 Baselines

We choose two well-known and publicly-available audio generation models, AudioLDM2 [12] and MusicGen [5], as our baselines. Both of them can generate nearly realistic music. We describe below how we use them for editing:

- **AudioLDM2:** Following SDEdit [37], we perform the forward process (i.e., adding noise to the audio input x) partially for $0.75T$ steps, where T is the original number diffusion steps, and then denoise it back with the editing command y to obtain \tilde{x} .
- **MusicGen:** MusicGen is a Transformer-based text-to-audio model that generates discrete audio tokens. We use MusicGen-Melody (1.5B), which achieves melody conditioning using chromagram [19] as a proxy. We input y as the text prompt, and the chromagram of x as the audio condition, for MusicGen to generate \tilde{x} .

We do not include the recent text-to-music editing methods InstructME [14] or MusicMagus [16], as they have not released the code and models, and also exclude M²UGen [15] as it is heavily focused on music understanding and visually-conditioned music generation.

5.5 Objective Metrics

We employ the following metrics:

- **CLAP [25]** is used to evaluate **transferability**, as it is trained with contrastive losses to align the representations for audio and text. We compute the cosine similarity between CLAP audio embedding for the edited audio \tilde{x} and CLAP text embedding for the command y .³ Higher scores show high semantic relevance between \tilde{x} and y .
- **Chroma similarity** computes the similarity of the original and edited audios \tilde{x} and x harmonically and rhythmically, thereby evaluates **fidelity**. We adopt librosa’s [38] CQT chroma method to extract the 12-dimensional chromagrams [19] to compute framewise cosine similarity.

³ For accompaniment generation task, text input to CLAP is modified to include both instruments, e.g., “Piano duet, played with violin.”

Model	CLAP↑ (transferability)	Chroma↑ (fidelity)	FAD↓ (overall)
MusicGen	0.339	0.771	8.443
AudioLDM2	0.284	0.643	5.389
AP-Adapter	0.314	0.777	5.986

Table 1: Objective evaluation on *in-domain* audio inputs of MusicGen-Melody [5], AudioLDM2-SDEdit [12, 37], and the proposed AP-Adapter. Results are the average of the three tasks. Best results are highlighted in bold (↑/↓: the higher/lower the better).

- **Fréchet audio distance (FAD) [36]** uses a pretrained audio classifier to extract audio features, collects features from all audios, and estimates the feature covariance matrix. Then, the Fréchet distance is computed between the two covariance matrices (one from generated audios, one from real audios). We adopt FAD to evaluate the **overall quality/realisticness** of the generations. Following the official implementation, we use VGGish architecture [39] as the feature extractor. We use the in-domain evaluation dataset as real audios.

5.6 Subjective Study

We design a listening test that contains 2 sets of music for each of the three tasks. The sets are independent from one another, and each contains a 10-second original audio prompt x , an editing text command y , and three edited audios \tilde{x} generated by our model and the two baselines (with order randomized and kept secret to participants). Participants rate each edited audio on a 5-point Likert scale, according to the following 3 aspects:

- **Transferability:** Do you feel that the generated audio matches what the text prompt asks for?
- **Fidelity:** Do you feel that the generated audio faithfully keeps the original musical content that should not be changed by the text prompt?
- **Overall preference:** Overall, how much do you like the generated audio?

We recruit 30 participants from our social circle and randomly assign them one of the 6 test suites (3 for in-domain, 3 for out-of-domain). The study takes about 10 minutes.

Eval. audios	Metric Task	Transferability MOS			Fidelity MOS			Overall MOS		
		Timbre	Genre	Accomp.	Timbre	Genre	Accomp.	Timbre	Genre	Accomp.
In-domain	MusicGen	3.35	3.15	3.32	2.62	2.85	2.76	3.06	3.03	2.91
	AudioLDM2	3.21	2.74	3.12	2.21	2.21	2.26	2.47	2.56	2.47
	AP-Adapter	3.59	3.44	3.41	3.47	3.74	3.41	3.26	3.44	3.12
Out-of-domain	MusicGen	2.92	3.96	3.00	2.73	3.31	2.54	2.58	3.58	2.65
	AudioLDM2	2.62	2.12	2.96	2.42	2.69	2.23	2.58	2.31	2.81
	AP-Adapter	2.92	3.19	3.54	3.81	3.58	3.96	3.08	3.12	3.31

Table 2: Subjective study results (mean opinion scores $\in [1, 5]$) with 17 and 13 participants for in-domain and out-of-domain input audios, respectively, for the three evaluation tasks: timbre transfer, genre transfer, and accompaniment generation.

6. RESULTS AND DISCUSSION

6.1 Hyperparameter Choices

We discover in our early experiments that several hyperparameters, which are tunable during inference, can drastically affect the edited outputs. Therefore, we conduct a systematic study on the effects of audio pooling rate ω (Sec. 4.1), AP scale in decoupled cross-attention α (Sec. 4.2), and classifier-free guidance scale λ (Sec. 3.3). Specifically, we observe how their various values induce different behaviors on the transferability-fidelity plane spanned by CLAP and chroma similarity metrics.

- The **pooling rate** ω controls the amount of information from the audio prompt. Figure 2a shows clearly that when the pooling rate is low, the fidelity is higher, but at the cost of transferability. For example, the audio generated with $\omega = 1$ preserves abundant acoustic information, thus the edited audio sounds like the input audio, but it might not reflect the editing command. The opposite can be said for $\omega = 8$. Overall, $\omega = 2$ or 4 strikes a good balance.
- The **AP scale** α adjusts the relative importance between the text and audio decoupled cross-attentions. As opposed to pooling rate, it enhances fidelity at the expense of transferability at higher values, as shown in Figure 2b, and $\alpha \in [0.4, 0.6]$ leads to a more balanced performance.
- The **CFG guidance scale** λ dictates the strength of text condition as detailed in Eqn. (5). As shown in Figure 2c, somewhat unexpectedly, λ does not impact the tradeoff too much when $\lambda \geq 3.5$. Hence, we use $\lambda = 7.5$ across all tasks following AudioLDM2.

6.2 Objective Evaluations

We show the metrics computed on in-domain audios in Table 1, taking the average across the three editing tasks. (We do not report the result for out-of-domain audio inputs as we expect CLAP and FAD to be less reliable there.) In general, AP-Adapter exhibits the most well-rounded performance without significant weaknesses—MusicGen scores high on transferability, but has a much worse FAD score, indicating issues on quality or distributional deviation. We infer that, since MusicGen only considers melody as input rather than the entire audio, it has fewer limitations in the generating process and thus achieves a higher transferability score. On the other hand, AudioLDM2 consistently achieves the best FAD score but lacks fidelity and transferability.

We also evaluate the ablated version of AP-Adapter without using the negative prompt (y^-). For the timbre transfer task, not using the negative prompt induces worse transferability, degrading the CLAP score from 0.405 to 0.378, but does not negatively impact chroma similarity and FAD.

6.3 Subjective Evaluations

Table 2 shows the results from our listening test. Our AP-adapter outperforms the two other baseline models in 16 out of 18 comparisons. On top of preserving fine-grained details in the input audio, AP-adapter also tightly follows the editing commands and generate relatively high-quality music, leading in transferability and overall preference except for only the genre transfer task on out-of-domain audios. MusicGen performs better in transferability for genre transfer, but its fidelity is weaker as it only considers the melody of the input audio. With the additional audio-modality condition, AP-adapter has the advantage of “listening” to all the details of the input audio, receiving significantly higher scores on fidelity on both in- and out-of-domain cases.

The advantage of AP-adapter in fidelity is much stronger in Table 2 rather than in Table 1. We conjecture that chroma similarity paints only a partial picture for fidelity as it is focused primarily on harmonic properties, leaving out other musical elements such as dynamics and percussive patterns.

7. CONCLUSIONS

We presented AP-Adapter, a lightweight add-on to AudioLDM2 that empowers it for music editing. AP-Adapter leverages AudioMAE to extract fine-grained features from the audio prompt, and feeds such features into AudioLDM2 via decoupled cross-attention adapters for effective conditioning. With only 500 hours of training data and 22M trainable parameters, AP-Adapter delivers compelling performance across useful editing tasks, namely, timbre transfer, genre transfer, and accompaniment generation. Additionally, it enables users to manipulate the transferability-fidelity tradeoff, and edit out-of-domain audios, which promotes creative endeavors with ethnic instrument audios that are usually scarce in publicly available datasets.

Promising directions for follow-up works include: (i) exploring more diverse editing tasks under our framework with various editing commands, (ii) extending AP-Adapter to other generative backbones, e.g., autoregressive models, and (iii) adding support for localized edits that can be stitched seamlessly with unchanged audio segments.

8. ACKNOWLEDGMENT

The work is supported by grants from the National Science and Technology Council of Taiwan (NSTC 112-2222-E-002-005-MY2) and (NSTC 113-2628-E-002-029), and the Ministry of Education (NTU-112V1904-5).

9. REFERENCES

- [1] S. Forsgren and H. Martiros, “Riffusion: Stable diffusion for real-time music generation,” 2022. [Online]. Available: <https://riffusion.com>
- [2] A. Agostinelli, T. I. Denk, Z. Borsos, J. Engel, M. Verzetti, A. Caillon, Q. Huang, A. Jansen, A. Roberts, M. Tagliasacchi *et al.*, “MusicLM: Generating music from text,” *arXiv preprint arXiv:2301.11325*, 2023.
- [3] H. Liu, Z. Chen, Y. Yuan, X. Mei, X. Liu, D. Mandic, W. Wang, and M. D. Plumbley, “AudioLDM: Text-to-audio generation with latent diffusion models,” in *Proceedings of International Conference on Machine Learning (ICML)*, 2023.
- [4] Q. Huang, D. S. Park, T. Wang, T. I. Denk, A. Ly, N. Chen, Z. Zhang, Z. Zhang, J. Yu, C. Frank, J. Engel, Q. V. Le, W. Chan, Z. Chen, and W. Han, “Noise2music: Text-conditioned music generation with diffusion models,” *arXiv preprint arXiv:2302.03917*, 2023.
- [5] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Défossez, “Simple and controllable music generation,” *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.
- [6] L. Lin, G. Xia, J. Jiang, and Y. Zhang, “Content-based controls for music large language modeling,” *arXiv preprint arXiv:2310.17162*, 2023.
- [7] J. Melechovsky, Z. Guo, D. Ghosal, N. Majumder, D. Herremans, and S. Poria, “Mustango: Toward controllable text-to-music generation,” in *Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2024.
- [8] S.-L. Wu, C. Donahue, S. Watanabe, and N. J. Bryan, “Music ControlNet: Multiple time-varying controls for music generation,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 32, pp. 2692–2703, 2024.
- [9] Y.-H. Lan, W.-Y. Hsiao, H.-C. Cheng, and Y.-H. Yang, “MusiConGen: Rhythm and chord control for Transformer-based text-to-music generation,” in *International Society for Music Information Retrieval Conference (ISMIR)*, 2024.
- [10] C. A. Huang, H. V. Koops, E. Newton-Rex, M. Dinulescu, and C. J. Cai, “Human-AI co-creation in songwriting,” in *International Society for Music Information Retrieval Conference (ISMIR)*, 2020.
- [11] R. Louie, J. H. Engel, and C. A. Huang, “Expressive communication: A common framework for evaluating developments in generative models and steering interfaces,” in *ACM Intelligent User Interfaces Conference (IUI)*, 2022.
- [12] H. Liu, Q. Tian, Y. Yuan, X. Liu, X. Mei, Q. Kong, Y. Wang, W. Wang, Y. Wang, and M. D. Plumbley, “AudioLDM 2: Learning holistic audio generation with self-supervised pretraining,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 32, pp. 2871–2883, 2024.
- [13] P.-Y. Huang, H. Xu, J. Li, A. Baevski, M. Auli, W. Galuba, F. Metze, and C. Feichtenhofer, “Masked autoencoders that listen,” *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.
- [14] B. Han, J. Dai, X. Song, W. Hao, X. He, D. Guo, J. Chen, Y. Wang, and Y. Qian, “InstructME: An instruction guided music edit and remix framework with latent diffusion models,” *arXiv preprint arXiv:2308.14360*, 2023.
- [15] A. S. Hussain, S. Liu, C. Sun, and Y. Shan, “M²UGen: Multi-modal music understanding and generation with the power of large language models,” *arXiv preprint arXiv:2311.11255*, 2023.
- [16] Y. Zhang, Y. Ikemiya, G. Xia, N. Murata, M. Martínez, W.-H. Liao, Y. Mitsufuji, and S. Dixon, “MusicMagus: Zero-shot text-to-music editing via diffusion models,” in *Proceedings of International Joint Conferences on Artificial Intelligence (IJCAI)*, 2024.
- [17] M. Plitsis, T. Kouzelis, G. Paraskevopoulos, V. Katsouros, and Y. Panagakis, “Investigating personalization methods in text to music generation,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 1081–1085.
- [18] H. Ye, J. Zhang, S. Liu, X. Han, and W. Yang, “IP-Adapter: Text compatible image prompt adapter for text-to-image diffusion models,” *arXiv preprint arXiv:2308.06721*, 2023.
- [19] D. Ellis, “Chroma feature analysis and synthesis,” *Resources of laboratory for the recognition and organization of speech and Audio-LabROSA*, 2007.
- [20] J. Ho, A. Jain, and P. Abbeel, “Denoising diffusion probabilistic models,” *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.
- [21] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, “Score-based generative modeling through stochastic differential equations,” in *International Conference on Learning Representations (ICLR)*, 2021.
- [22] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, “High-resolution image synthesis with latent diffusion models,” in *Proceedings of the IEEE/CVF*

- Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.
- [23] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” *arXiv preprint arXiv:1312.6114*, 2013.
- [24] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, Y. Li, X. Wang, M. Dehghani, S. Brahma *et al.*, “Scaling instruction-finetuned language models,” *Journal of Machine Learning Research (JMLR)*, 2024.
- [25] Y. Wu, K. Chen, T. Zhang, Y. Hui, T. Berg-Kirkpatrick, and S. Dubnov, “Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023.
- [26] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, “Language models are unsupervised multitask learners,” *Open AI Blog*, 2019.
- [27] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2015.
- [28] J. Ho and T. Salimans, “Classifier-free diffusion guidance,” *arXiv preprint arXiv:2207.12598*, 2022.
- [29] R. Gal, M. Arar, Y. Atzmon, A. H. Bermano, G. Chechik, and D. Cohen-Or, “Encoder-based domain tuning for fast personalization of text-to-image models,” *ACM Transactions on Graphics (TOG)*, 2023.
- [30] N. Kumari, B. Zhang, R. Zhang, E. Shechtman, and J.-Y. Zhu, “Multi-concept customization of text-to-image diffusion,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 1931–1941.
- [31] H. concept library, “SD-Dreambooth-Library,” <https://huggingface.co/sd-dreambooth-library>, 2024, [Online; accessed 10-April-2024].
- [32] Stable Diffusion Art, “How does negative prompt work?” 2024, [Online; accessed 10-April-2024]. [Online]. Available: <https://stable-diffusion-art.com/how-negative-prompt-work/>
- [33] G. Sanchez, H. Fan, A. Spangher, E. Levi, P. S. Ammanamanchi, and S. Biderman, “Stay on topic with classifier-free guidance,” *arXiv preprint arXiv:2306.17806*, 2023.
- [34] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, “Audio Set: An ontology and human-labeled dataset for audio events,” in *IEEE international conference on acoustics, speech and signal processing (ICASSP)*, 2017.
- [35] “AI Hub Dataset,” <https://www.aihub.or.kr/aihubdata/data/view.do?currMenu=115&topMenu=100&dataSetSn=71470>.
- [36] K. Kilgour, M. Zuluaga, D. Roblek, and M. Sharifi, “Fréchet Audio Distance: A metric for evaluating music enhancement algorithms,” *arXiv preprint arXiv:1812.08466*, 2018.
- [37] C. Meng, Y. He, Y. Song, J. Song, J. Wu, J.-Y. Zhu, and S. Ermon, “SDEdit: Guided image synthesis and editing with stochastic differential equations,” in *International Conference on Learning Representations (ICLR)*, 2022.
- [38] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, “librosa: Audio and music signal analysis in python.” in *SciPy*, 2015.
- [39] S. Hershey, S. Chaudhuri, D. P. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold *et al.*, “CNN architectures for large-scale audio classification,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.

MELODYT5: A UNIFIED SCORE-TO-SCORE TRANSFORMER FOR SYMBOLIC MUSIC PROCESSING

Shangda Wu^{1,‡} Yashan Wang^{1,‡} Xiaobing Li¹ Feng Yu¹ Maosong Sun^{1,2,‡}

¹ Central Conservatory of Music, China ² Tsinghua University, China

{shangda, alexis_wang}@mail.ccom.edu.cn, sms@tsinghua.edu.cn

<https://github.com/sanderwood/melodyt5>

ABSTRACT

In the domain of symbolic music research, the progress of developing scalable systems has been notably hindered by the scarcity of available training data and the demand for models tailored to specific tasks. To address these issues, we propose MelodyT5, a novel unified framework that leverages an encoder-decoder architecture tailored for symbolic music processing in ABC notation. This framework challenges the conventional task-specific approach, considering various symbolic music tasks as score-to-score transformations. Consequently, it integrates seven melody-centric tasks, from generation to harmonization and segmentation, within a single model. Pre-trained on MelodyHub, a newly curated collection featuring over 261K unique melodies encoded in ABC notation and encompassing more than one million task instances, MelodyT5 demonstrates superior performance in symbolic music processing via multi-task transfer learning. Our findings highlight the efficacy of multi-task transfer learning in symbolic music processing, particularly for data-scarce tasks, challenging the prevailing task-specific paradigms and offering a comprehensive dataset and framework for future explorations in this domain.

1. INTRODUCTION

In the field of artificial intelligence, symbolic music processing—including the analysis and generation of musical scores—presents a unique challenge that merges musical creativity with computational complexity. Symbolic music, which represents musical information with discrete symbols rather than continuous audio signals, facilitates the precise manipulation and analysis of elements such as melody, harmony, and rhythm. Historically, the application of AI in this area has sought not only to mimic the creative process of human composers [1–4] but also

to uncover the underlying patterns of musical composition [5–7].

Despite significant progress, the field still faces persistent limitations. One notable challenge is the prevalence of task-specific models [8–11]. These models offer benefits for specific applications but lack adaptability to the broader spectrum of symbolic music processing. This fragmentation is further compounded by the scarcity of annotated datasets [12–14], which serve as the lifeblood of deep learning models. Unlike other domains where data may be abundant and easy-to-collect, annotated symbolic music datasets are both rare and costly to produce. Without access to ample and diverse data, models struggle to generalize and may exhibit biases or limitations [15] in their analysis and generation of symbolic music.

In addressing the challenges inherent to symbolic music processing, insights from the Natural Language Processing (NLP) domain offer a promising avenue for advancement. Techniques such as transfer learning [16–18] and multi-task learning [19–21] have played a pivotal role in advancing NLP by promoting the transfer of knowledge from pre-trained language models and exploiting common patterns across various tasks. Prominent models like GPT [22], BERT [23], and T5 [24] demonstrate the efficacy of these strategies in understanding and generating language across diverse contexts. Notably, the T5 model, with its text-to-text framework, mirrors the conceptual shift necessary for symbolic music by treating all tasks as variations of converting input scores to output scores. By embracing such methodologies, which regard tasks as facets of a unified problem, we seek to develop models for symbolic music that not only excel in specific tasks but are also adaptable and proficient across a wide range of tasks.

In this paper, we introduce MelodyT5, which leverages an encoder-decoder framework to perform multiple symbolic music tasks as unified score-to-score transformations. Pre-trained on the MelodyHub dataset, which contains over 1 million task instances across seven melody-centric tasks in ABC notation, MelodyT5 overcomes the limitations of task-specific models and sparse data availability in symbolic music processing. By implementing bar patching [7, 25], MelodyT5 can handle longer sequences effectively, expanding its applicability to a wider range of tasks while maintaining computational efficiency. Our results underscore the promise of employing multi-task learning approaches in symbolic music processing, demonstrating

[‡] These authors contributed equally.

[‡] Corresponding author.



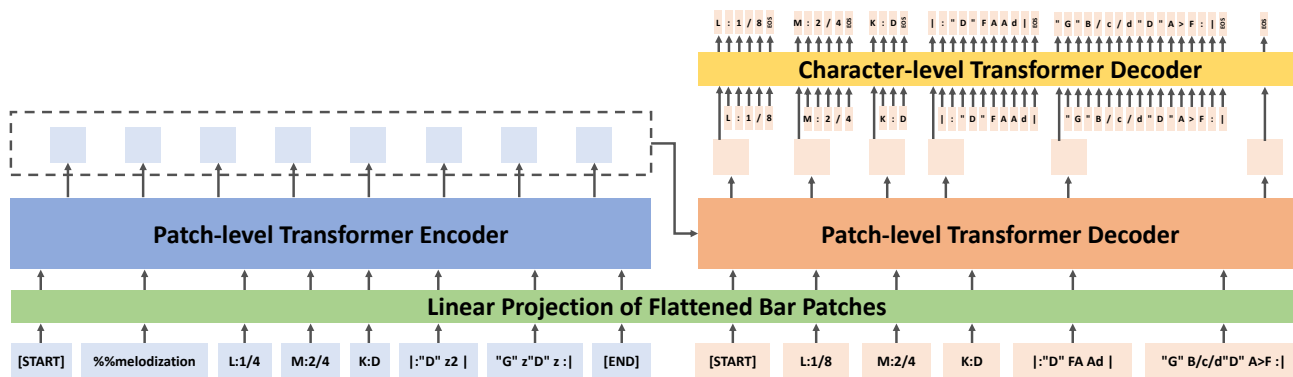


Figure 1. The MelodyT5 framework employs a Transformer encoder-decoder architecture with bar patching for music processing. It uses linear projection of input bar patches, fed into a patch-level Transformer encoder. The encoder output provides context for a patch-level Transformer decoder to autoregressively produce target bar features. A character-level Transformer decoder then uses these features to generate detailed characters for each bar, forming the target musical score.

the superior performance of MelodyT5 across a spectrum of tasks and providing a rich dataset for future research. The key contributions of our paper are as follows:

- MelodyT5, employing an encoder-decoder approach, redefines symbolic music processing by including multiple tasks as unified score-to-score transformations, demonstrating versatility and breaking traditional task-specific constraints.
- MelodyHub, a dataset comprising 261,900 unique melodies in ABC notation across over 1 million task instances for seven different tasks, serves as the cornerstone for effective pre-training of MelodyT5.
- Our experimental results demonstrate the efficacy of multi-task transfer learning in symbolic music, with models trained across multiple tasks outperforming those trained in isolation.

2. METHODOLOGY

In this section, we delve into the methodology behind MelodyT5. We first introduce the ABC notation and bar patching for music representation, then present the architectural design of MelodyT5, and finally outline the pre-training objective of our model, which focuses on score-to-score transformations as the basis for multi-task learning.

2.1 Data Representation

We utilize ABC notation, a concise symbolic music format, for encoding musical scores with ASCII characters. This text-based format elegantly represents musical elements like notes, rhythms, and articulations in a human-readable manner, thereby facilitating thorough music documentation. Additionally, it promotes the utilization of NLP techniques for both music analysis and generation, as evidenced by recent studies [4, 26, 27].

To process musical scores encoded in ABC notation more efficiently, we implement the bar patching technique [7, 25]. Bar patching involves breaking down musical sequences into units called bar patches. Each of these units

corresponds to either a bar or an information field (such as key and meter), including a sequence of characters that represent musical symbols within that patch. Unlike the conventional character-level or token-level tokenization of ABC notation [28, 29], where individual characters or tokens are processed independently, bar patching groups multiple characters into cohesive semantic units. Typically, each patch comprises 10 or more tokens, thus effectively reducing the overall sequence length of musical scores.

2.2 Model Architecture

As shown in Fig. 1, the MelodyT5 framework employs an encoder-decoder architecture based on the Transformer network [30], tailored for symbolic music processing. Integrating bar patching into MelodyT5 requires the incorporation of two additional components: a linear projection layer and a character-level decoder. Consequently, the model architecture encompasses the following modules:

Linear Projection: This component converts each bar patch into a dense embedding. It takes a multi-hot vector as input, formed by concatenating one-hot vectors representing characters within the bar patch with the shape $S \times V$, where S represents the patch size (i.e., the maximum number of characters in a patch) and V represents the vocabulary size. If a patch contains fewer than S characters, it will be padded with a special token to make it a S -character patch. The vector is then mapped to a dense embedding, serving as input to the patch-level encoder or decoder.

Patch-level Encoder: It is responsible for generating contextualized representations to understand the input musical score by operating on the dense embeddings produced by the linear projection layer. Leveraging mechanisms like self-attention and feed-forward neural networks, it captures global dependencies within the input musical score.

Patch-level Decoder: Tasked with generating the dense representation of the next bar patch, the patch-level decoder utilizes contextualized representations from the encoder and patch embeddings of previously generated content. It employs cross-attention and autoregressive generation mechanisms, ensuring global coherence and continuity in the sequence of generated bar patches.

Table 1. The MelodyHub collection statistics include the number of instances for each task along with the corresponding data sources. The JSB Chorales dataset is augmented to 15 keys due to its small size and original data in the C key.

Data Sources	Cataloging	Generation	Harmonization	Melodization	Segmentation	Transcription	Variation
<i>ABC Notation</i> [31]	184,660	184,738	31,732	31,690	—	174,779	—
<i>FolkWiki</i> [32]	6,610	6,767	1,207	1,205	—	6,218	—
<i>JSB Chorales</i> [33]	4,980	4,980	4,950	4,950	19,125	4,980	—
<i>KernScores</i> [34]	1,731	1,776	—	—	1,275	1,754	—
<i>Meertens Tune Collections</i> [35]	16,662	16,662	—	—	16,660	16,297	—
<i>Nottingham</i> [36]	1,031	1,031	1,014	1,014	—	1,021	—
<i>OpenScore Lieder</i> [37]	1,326	1,326	—	—	—	1,255	—
<i>The Session</i> [38]	44,620	44,620	3,081	3,078	—	42,838	174,104
<i>Total</i>	261,620	261,900	41,984	41,937	37,060	249,142	174,104

Character-level Decoder: Operating in a step-by-step manner, the character-level decoder produces characters within the next bar patch based on the dense representation generated by the patch-level decoder. By utilizing the dense representation as a context vector, it decodes each character within the bar, focusing on local information, and sequentially reconstructs every bar patch until it completes the generation of the target musical score.

The encoder-decoder architecture with bar patching in MelodyT5 enables efficient score-to-score transformations by hierarchically modelling music at both patch and character levels, capturing global structure and local details inherent in compositions.

2.3 Pre-training Objective

The pre-training objective of MelodyT5 aims to optimize a unified encoder-decoder framework for processing and generating symbolic music across a variety of tasks, utilizing cross-entropy loss for next token prediction.

We consider a dataset D consisting of pairs (X, Y) , where X is an input musical score and Y is the target musical score. Each score is represented as a sequence of bar patches $\{B_1, B_2, \dots, B_n\}$, with each bar patch B_i further decomposed into a sequence of characters $\{c_1, c_2, \dots, c_m\}$. The model is trained to predict each token (i.e., character) of the target score given the input score and the previously generated tokens in an autoregressive manner.

Formally, the pre-training objective can be represented as minimizing the cross-entropy loss across all tokens in the target sequence:

$$\mathcal{L}(\theta) = - \sum_{(X,Y) \in D} \sum_{i=1}^n \sum_{j=1}^m \log P_{\theta}(c_j^i | X, B_{<i}, c_{<j}^i) \quad (1)$$

where c_j^i is the j -th character in the i -th bar patch of score Y , $B_{<i}$ includes all bar patches before the i -th, $c_{<j}^i$ are characters before the j -th in the current patch, and P_{θ} is the probability of the model, parameterized by θ , of predicting the correct character.

This objective incorporates the fundamental principle that the vast majority of symbolic music tasks can be considered as transformations from score to score, or, in other words, from an input musical score to a target musical

score. By pre-training on this objective, MelodyT5 acquires the ability to understand and replicate a wide array of patterns and structures inherent to different music tasks, which is pivotal for its success across various applications within symbolic music processing.

3. DATASET

This section outlines the melody curation and task definition of the MelodyHub dataset. MelodyHub, crucial for training MelodyT5, comprises seven melody-centric tasks. This collection, sourced from sheet music datasets, includes folk songs and other non-copyrighted musical scores from various traditions and epochs.

3.1 Melody Curation

The MelodyHub dataset was curated using publicly available sheet music datasets and online platforms, with original formats like ABC notation, MusicXML, and Humdrum. The data curation process included several steps:

1. Entries featuring explicit copyright indicators such as “copyright” or “©” symbols were excluded.
2. All data was converted to MusicXML format for standardization and subsequently transformed into ABC notation to ensure format consistency.
3. Melodies consisting of fewer than eight bars were omitted from the dataset to maintain adequate complexity and musical richness.
4. Removal of lyrics and non-musical content (e.g., contact information of transcribers and URL links) aimed to focus solely on musical notation.
5. Leading and trailing bars of complete rest were removed from each piece.
6. Each piece underwent verification for the presence of a final barline, with addition if absent.
7. Entries were deduplicated to prevent redundancy.

By ensuring the quality and consistency of the MelodyHub dataset, these steps led to a substantial collection of 261,900 melodies with uniform formatting, making it suitable for training and evaluating symbolic music models like MelodyT5.

3.2 Task Definition

Following the curation of melody data, the MelodyHub dataset was segmented into seven tasks, as summarized in Table 1, presented in a score-to-score format with input-output pairs. In MelodyHub, every input data includes a task identifier (e.g., `%%harmonization`) at the outset to specify the intended task. Below are the definitions of these tasks:

Cataloging: This task selects melodies with music-related metadata like titles, composers, and geographical origins (e.g., `C:J.S. Bach, O:Germany`). The input data includes information fields with these attributes, while specific information is removed and the order is randomized. The output includes the corresponding metadata without the musical score.

Generation: Here, the input solely consists of a task identifier (i.e., `%%generation`), while the output comprises comprehensive musical scores. Following TunesFormer [25], control codes are affixed to all melodies as information fields to denote musical structure information. These codes, namely `S:`, `B:`, and `E:`, signify the number of sections, bars per section, and edit distance similarity between every pair of sections within the tune.

Harmonization: This task involves melodies containing chord symbols. The chord symbols are removed from the input, while the original data is retained as the output. An additional information field denoting edit distance similarity (`E:`) is appended to the output, indicating the similarity between the input and output, ranging from 0 to 10 (no match at all to exact match). Lower similarity values suggest the need for more chord symbols.

Melodization: In contrast to harmonization, this task operates inversely and also employs melodies containing chord symbols. The notes in the original score are replaced with rests, and adjacent rest durations are combined. The resultant score, comprising rests and chord symbols, serves as the input. Similar to harmonization, an `E:` field is added at the outset of the output, with lower values facilitating the generation of more intricate melodies.

Segmentation: Melodies in Humdrum format (i.e., KernScores and Meertens Tune Collections) containing curly braces indicating segmentation or voices from the JSB Chorales dataset (four-part compositions) with fermatas are chosen. These markers are transformed into breath marks. The input data omits all breath marks, while the output introduces an `E:` field at the beginning to aid the generation of breath marks, with lower values implying the need for more breath marks to be added.

Transcription: ABC notation is initially converted to MIDI, then reconverted back to ABC. The resultant ABC from the MIDI conversion loses substantial score information, such as distinguishing enharmonic equivalents and missing musical ornaments (e.g., trill). The MIDI-converted ABC serves as the input, while the original ABC, appended with an added `E:` field, constitutes the output. Lower `E:` values denote greater discrepancies between the transcribed and input scores, particularly due to absent repeat symbols.

Variation: This task centres on data from The Session, wherein each ABC notation file may contain multiple variants of the same tune. Tunes with two or more variations are selected, with every possible pair of variants utilized as both input and output. The output initiates with an `E:` field signifying the extent of disparities between the input and output scores, with lower values suggesting substantial variations in the musical scores.

Together, resulting in 1,067,747 task instances in total, these tasks include various MIR challenges from analytical to generative, providing a comprehensive resource¹ for developing symbolic music models like MelodyT5.

4. EXPERIMENTS

This section evaluates the effectiveness of MelodyT5 in symbolic music processing through a series of experiments. It outlines experimental settings, conducts ablation studies on multi-task learning impact, and compares MelodyT5 with baseline models in various tasks.

4.1 Settings

The experiments are structured to systematically assess the capabilities of MelodyT5 for diverse symbolic music tasks. We utilize the MelodyHub dataset, which is randomly split into 99% for training and 1% for validation.

MelodyT5 features a 9-layer patch-level encoder and decoder with shared weights, a 3-layer character-level decoder, and a hidden size of 768, amounting to 113 million parameters. This configuration processes ABC sequences up to 16,384 characters, with a 256 patch length and a 64 patch size. It employs a 128-size ASCII-based vocabulary, using characters 0-2 for special tokens (pad, bos, and eos).

The AdamW optimizer [39] is used, setting a learning rate of $2e-4$. The process includes a 3-epoch warmup, a constant learning rate over 32 epochs, and a batch size of 10 for each GPU, ensuring consistency in hyperparameter settings across all tasks. It took approximately 2 days to complete the pre-training using 6 RTX 3090 GPUs.

In ablation studies, we investigate the effects of multi-task learning on MelodyT5, considering three settings: 1) omitting pre-training, 2) using only the downstream task-specific data from MelodyHub, or 3) utilizing the entire MelodyHub dataset, which includes all tasks.

In terms of comparative evaluations, we select open-source models that excel in their respective domains for benchmarking. MelodyT5 is fine-tuned on identical datasets to these models, ensuring fairness in comparison. For models trained on proprietary datasets, we retrain them using accessible datasets to ensure reproducibility.

Our objective evaluation strategy includes ablation studies focused on bits-per-byte (BPB) for consistent measurement, alongside task-specific metrics for comparative evaluations. Additionally, A/B tests are conducted for the subjective evaluation against baseline models.

¹ <https://huggingface.co/datasets/sander-wood/melodyhub>

Table 2. Experimental results from ablation studies illustrate the impact of multi-task learning on diverse symbolic music tasks, evaluated through BPB (bits-per-byte) to compare performance across various pre-training settings.

Pre-training	Cataloging <i>WikiMT [40]</i>	Generation <i>Wikifonia [41]</i>	Harmonization <i>CMD [42]</i>	Melodization <i>EWLD [43]</i>	Segmentation <i>Essen [44]</i>	Transcription <i>Liederschatz [45]</i>	Variation <i>The Session [38]</i>
<i>None</i>	0.0376	1.2382	0.5680	0.7949	0.0272	1.1938	0.4932
<i>Task-Specific</i>	0.0379	0.8850	0.3393	0.6322	0.0224	0.3432	–
<i>Multi-Task</i>	0.0350	0.8472	0.2925	0.5067	0.0119	0.2969	0.3949

4.2 Ablation Studies

In our ablation studies, MelodyT5 was evaluated on the test sets of various symbolic music benchmarks. Due to the lack of a directly suitable external dataset for the variation task, we chose to evaluate using the validation set of The Session. As a result, there was no task-specific pre-training for the variation task.

The ablation studies aim to explore two aspects: 1) the overall efficacy of pre-training, particularly in the context of multi-task pre-training versus task-specific pre-training, and 2) the extent to which performance gains from multi-task pre-training vary among different tasks, especially considering differences in the available volume of pre-training data across these tasks.

The ablation studies, as depicted in Table 2, show that pre-training is crucial for improving the performance of symbolic music tasks. Models trained with pre-training consistently outperform those without, indicating that pre-training enhances model generalization and performance. Multi-task pre-training is also superior to task-specific pre-training, as models trained with multi-task pre-training show lower BPB scores. This highlights the importance of leveraging multi-task pre-training to effectively capture shared patterns and structures in symbolic music data, enabling MelodyT5 to generalize better to downstream tasks.

Furthermore, it is noteworthy that while multi-task pre-training consistently yields performance gains across most tasks, the extent of improvement varies, which significantly correlates with the volume of task-specific data available for pre-training. Specifically, tasks with less data, such as segmentation and melodization, showcase more substantial performance gains from multi-task learning. On the other hand, tasks with more data, like generation and cataloging, though still benefiting from multi-task pre-training, show relatively smaller improvements. This observation suggests that while multi-task learning enhances model performance across the board, its impact is especially notable in data-constrained scenarios.

In summary, the ablation studies demonstrate the effectiveness of multi-task learning and underscore the impact of data volume on the benefits derived from such an approach. Multi-task learning boosts model performance across symbolic music tasks and provides notable advantages for tasks with limited data by leveraging shared knowledge across tasks.

4.3 Comparative Evaluations

For comparative evaluations, we compare MelodyT5, which is multi-task pre-trained on MelodyHub, with several task-specific baseline models, focusing on melody generation, harmonization, melodization, and segmentation. These tasks are well-established and have open-source models as competitive baselines. The following baseline models have been selected for comparison:

- **TunesFormer** [25] is applied for melody generation, featuring a Transformer-based architecture with bar patching and control codes. This approach aims to refine the efficiency of the generation process and ensure adherence to musical forms.
- **STHarm** [46] is utilized as the baseline in melody harmonization, employing a Transformer framework to convert melodies into chords. Its primary focus is on creating harmonies that preserve the structural integrity of the original melody.
- **CMT** [9] is chosen for melodization, which involves generating melodies based on chord progressions. It employs a phased training approach, conditioning the generation of rhythm and pitch on the chords to produce dynamic and coherent musical outputs.
- **Bi-LSTM-CRF** [8] is used for melody segmentation, integrating Bi-LSTM and CRF to effectively identify and segment melodic phrases for music structure analysis.

For an objective and quantifiable performance assessment that ensures reproducibility, we leverage previously established task-specific metrics. The selected metrics for our assessment include:

- **CTRL (Controllability)** [25]: Evaluates the precision of generation control through edit distance similarity between intended and actual control codes.
- **CTnCTR & PCS & MCTD** [47]: These chord/melody harmonicity metrics evaluate harmonization and melodization tasks by assessing harmonic and melodic compatibility between melodies and chords.
- **F1 Score**: Measures the balance between precision and recall in identifying correctly segmented melodic phrases.

Table 3. Comparative objective evaluation of the MelodyT5 model against task-specific baselines across various symbolic music tasks, utilizing task-related metrics previously established. The baselines include TunesFormer [25] for generation, STHarm [46] for harmonization, CMT [9] for melodization, and Bi-LSTM-CRF [8] for segmentation.

Models	Generation				Harmonization			Melodization			Segmentation
	CTRL↑	CTnCTR↑	PCS↑	MCTD↓	CTnCTR↑	PCS↑	MCTD↓	CTnCTR↑	PCS↑	MCTD↓	F1 Score↑
MelodyT5	0.8664	0.7108	0.3274	1.2080	0.8438	0.5084	1.0320	0.8438	0.5084	1.0320	0.9055
Baselines	0.8162	0.5963	0.2343	1.3125	0.8607	0.4863	1.0610	0.8607	0.4863	1.0610	0.8400

Table 3 shows that MelodyT5 outperforms task-specific baselines in all tasks. It surpasses the specialized baseline TunesFormer in melody generation, demonstrating enhanced control and precision in generating melodies according to specific musical forms. MelodyT5 leads in harmonization, producing chords that are harmonically compatible with the given melodies while maintaining structural coherence. Although slightly trailing CMT in CTnCTR, it still shows robust performance in other metrics for melodization, demonstrating its ability to generate melodies well integrated with chord progressions. Its performance in melody segmentation is significant, indicating its ability to accurately discern and segment melodic phrases. This performance, achieved without task-specific modifications, highlights the effectiveness of multi-task transfer learning combined with unified score-to-score transformations in symbolic music processing.

In addition to the objective metrics presented in Table 3, we recognize the limitations of solely relying on such measures to evaluate the quality of generated music. Thus, we further explored these areas (i.e., generation, harmonization, and melodization) through subjective experiments to capture listener preferences.

For our subjective evaluation, we randomly chose 30 pieces from the test set for each task and conducted blind A/B testing. Participants were presented with one randomly chosen pair from each of these 30 pairs to compare musical scores generated by MelodyT5 and baseline models under identical conditions. They were asked to choose between MelodyT5, the baseline, or no preference. Each comparison was included in two videos, showcasing both the audio and the Sibelius-rendered musical scores.

For generation, we compared the quality of melodies generated by MelodyT5 and TunesFormer, given the same control codes and information fields. In harmonization and melodization, comparisons were made against baselines given identical melodies or chords, respectively. To ensure fairness, especially considering the baseline model for melodization was limited to generating outputs of only 8 bars, we trimmed the MelodyT5-generated scores to match the output length of this baseline model.

The study involved 155 responses from students and educators with music specializations, ensuring deep understanding of melody and harmony. To secure data reliability, submissions were filtered out of those completed in less than half the overall average duration of 4 minutes and 39 seconds, i.e., those under 2 minutes and 20 seconds. This resulted in a final tally of 124 valid questionnaires.

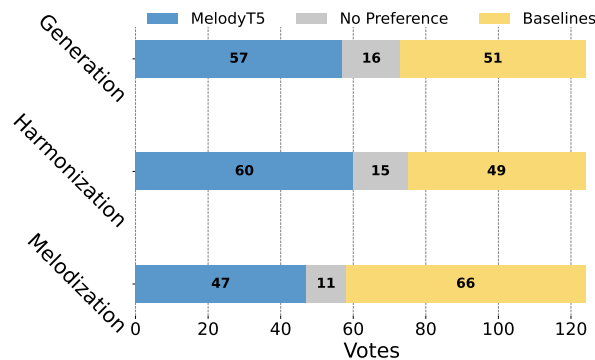


Figure 2. Comparative subjective evaluation of MelodyT5 against task-specific baselines in symbolic music tasks, showing vote counts for each model.

Based on the subjective evaluation in Fig. 2, we observe a notable preference for the MelodyT5 model over the baseline in the tasks of melody generation and harmonization, with MelodyT5 receiving a higher number of votes. However, the preferences reverse in the task of melodization, where the baseline model receives a greater number of votes compared to MelodyT5. This indicates that the baseline model CMT, which employs a two-phase training process focusing separately on rhythm and pitch conditioned on chord progressions, may align more closely with human rhythmic tendencies in melodization, leading to a preference for its outputs in the subjective evaluation.

Overall, MelodyT5 excels in symbolic music processing, outperforming task-specific models in most tasks and demonstrating the effectiveness of multi-task transfer learning in this domain, despite occasional shortcomings.

5. CONCLUSIONS

This study presents MelodyT5, a model addressing challenges in symbolic music processing by providing a unified framework for diverse tasks. By treating music tasks as score-to-score transformations, MelodyT5 significantly improves symbolic music processing through multi-task transfer learning. Objective and subjective evaluations demonstrate that MelodyT5 generally outperforms or matches task-specific baseline models without modification. The MelodyHub dataset, with over one million task instances, offers a rich resource for training and evaluating models. While excelling in melody-centric tasks, further optimization is required to tackle more complex musical compositions, such as polyphonic arrangements.

6. ETHICS STATEMENT

In our research, we are committed to upholding ethical standards regarding data privacy and copyright protection. We diligently adhere to these principles throughout our data collection and processing procedures. All data utilized in our study is sourced from publicly accessible repositories, and we have taken measures to exclude any known copyrighted materials.

7. ACKNOWLEDGEMENTS

We would like to express our sincere gratitude to Associate Professor Bob L. T. Sturm from KTH Royal Institute of Technology for his valuable discussions and insights into the early stages of the project. We also acknowledge Yuanliang Dong and Jiafeng Liu from the Central Conservatory of Music for their assistance, and Leqi Peng from Fuyin Technology for her support with subjective experiments.

This work was supported by the following funding sources: Special Program of National Natural Science Foundation of China (Grant No. T2341003), Advanced Discipline Construction Project of Beijing Universities, Major Program of National Social Science Fund of China (Grant No. 21ZD19), and the National Culture and Tourism Technological Innovation Engineering Project (Research and Application of 3D Music).

8. REFERENCES

- [1] J. Liu, Y. Dong, Z. Cheng, X. Zhang, X. Li, F. Yu, and M. Sun, "Symphony generation with permutation invariant language model," in *Proceedings of the 23rd International Society for Music Information Retrieval Conference, ISMIR 2022, Bengaluru, India, December 4-8, 2022*, 2022, pp. 551–558. [Online]. Available: <https://archives.ismir.net/ismir2022/paper/000066.pdf>
- [2] P. Lu, X. Tan, B. Yu, T. Qin, S. Zhao, and T. Liu, "Meloform: Generating melody with musical form based on expert systems and neural networks," in *Proceedings of the 23rd International Society for Music Information Retrieval Conference, ISMIR 2022, Bengaluru, India, December 4-8, 2022*, 2022, pp. 567–574. [Online]. Available: <https://archives.ismir.net/ismir2022/paper/000068.pdf>
- [3] L. Min, J. Jiang, G. Xia, and J. Zhao, "Polyffusion: A diffusion model for polyphonic score generation with internal and external controls," in *Proceedings of the 24th International Society for Music Information Retrieval Conference, ISMIR 2023, Milan, Italy, November 5-9, 2023*, 2023, pp. 231–238. [Online]. Available: <https://doi.org/10.5281/zenodo.10265265>
- [4] S. Wu and M. Sun, "Exploring the efficacy of pre-trained checkpoints in text-to-music generation task," in *The AAAI-23 Workshop on Creative AI Across Modalities*, 2023. [Online]. Available: <https://openreview.net/forum?id=QmWXskBhesn>
- [5] M. Zeng, X. Tan, R. Wang, Z. Ju, T. Qin, and T. Liu, "Musicbert: Symbolic music understanding with large-scale pre-training," in *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, ser. Findings of ACL, vol. ACL/IJCNLP 2021. Association for Computational Linguistics, 2021, pp. 791–800. [Online]. Available: <https://doi.org/10.18653/v1/2021.findings-acl.70>
- [6] Z. Wang and G. Xia, "Musebert: Pre-training music representation for music understanding and controllable generation," in *Proceedings of the 22nd International Society for Music Information Retrieval Conference, ISMIR 2021, Online, November 7-12, 2021*, 2021, pp. 722–729. [Online]. Available: <https://archives.ismir.net/ismir2021/paper/000090.pdf>
- [7] S. Wu, D. Yu, X. Tan, and M. Sun, "Clamp: Contrastive language-music pre-training for cross-modal symbolic music information retrieval," in *Proceedings of the 24th International Society for Music Information Retrieval Conference, ISMIR 2023, Milan, Italy, November 5-9, 2023*, 2023, pp. 157–165. [Online]. Available: <https://doi.org/10.5281/zenodo.10265247>
- [8] Y. Zhang and G. Xia, "Symbolic melody phrase segmentation using neural network with conditional random field," in *Proceedings of the 8th Conference on Sound and Music Technology: Selected Papers from CSMT*. Springer, 2021, pp. 55–65.
- [9] K. Choi, J. Park, W. Heo, S. Jeon, and J. Park, "Chord conditioned melody generation with transformer based decoders," *IEEE Access*, vol. 9, pp. 42 071–42 080, 2021. [Online]. Available: <https://doi.org/10.1109/ACCESS.2021.3065831>
- [10] S. Wu, X. Li, and M. Sun, "Chord-conditioned melody harmonization with controllable harmonic," in *IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP 2023, Rhodes Island, Greece, June 4-10, 2023*. IEEE, 2023, pp. 1–5. [Online]. Available: <https://doi.org/10.1109/ICASSP49357.2023.10096398>
- [11] S. Wu, Y. Yang, Z. Wang, X. Li, and M. Sun, "Generating chord progression from melody with flexible harmonic rhythm and controllable harmonic density," *EURASIP J. Audio Speech Music. Process.*, vol. 2024, no. 1, p. 4, 2024. [Online]. Available: <https://doi.org/10.1186/s13636-023-00314-6>
- [12] Z. Wang, K. Chen, J. Jiang, Y. Zhang, M. Xu, S. Dai, and G. Xia, "POP909: A pop-song dataset for music arrangement generation," in *Proceedings of the 21th International Society for Music Information Retrieval Conference, ISMIR 2020, Montreal, Canada, October 11-16, 2020*, 2020, pp. 38–45. [Online]. Available: <http://archives.ismir.net/ismir2020/paper/000089.pdf>

- [13] Y. Hsiao, T. Hung, T. Chen, and L. Su, “Bps-motif: A dataset for repeated pattern discovery of polyphonic symbolic music,” in *Proceedings of the 24th International Society for Music Information Retrieval Conference, ISMIR 2023, Milan, Italy, November 5-9, 2023*, 2023, pp. 281–288. [Online]. Available: <https://doi.org/10.5281/zenodo.10265277>
- [14] Y. Zhang, Z. Zhou, X. Li, F. Yu, and M. Sun, “Ccom-huqin: An annotated multimodal chinese fiddle performance dataset,” *Trans. Int. Soc. Music. Inf. Retr.*, vol. 6, no. 1, pp. 60–74, 2023. [Online]. Available: <https://doi.org/10.5334/tismir.146>
- [15] A. Holzapfel, B. L. Sturm, and M. Coeckelbergh, “Ethical dimensions of music information retrieval technology,” *Trans. Int. Soc. Music. Inf. Retr.*, vol. 1, no. 1, pp. 44–55, 2018. [Online]. Available: <https://doi.org/10.5334/tismir.13>
- [16] J. Howard and S. Ruder, “Universal language model fine-tuning for text classification,” in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*. Association for Computational Linguistics, 2018, pp. 328–339. [Online]. Available: <https://aclanthology.org/P18-1031/>
- [17] S. Ruder, M. E. Peters, S. Swayamdipta, and T. Wolf, “Transfer learning in natural language processing,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2, 2019, Tutorial Abstracts*. Association for Computational Linguistics, 2019, pp. 15–18. [Online]. Available: <https://doi.org/10.18653/v1/n19-5004>
- [18] M. Artetxe and H. Schwenk, “Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond,” *Trans. Assoc. Comput. Linguistics*, vol. 7, pp. 597–610, 2019. [Online]. Available: https://doi.org/10.1162/tacl_a_00288
- [19] X. Liu, P. He, W. Chen, and J. Gao, “Multi-task deep neural networks for natural language understanding,” in *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*. Association for Computational Linguistics, 2019, pp. 4487–4496. [Online]. Available: <https://doi.org/10.18653/v1/p19-1441>
- [20] K. Song, X. Tan, T. Qin, J. Lu, and T. Liu, “MASS: masked sequence to sequence pre-training for language generation,” in *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, ser. Proceedings of Machine Learning Research, vol. 97. PMLR, 2019, pp. 5926–5936. [Online]. Available: <http://proceedings.mlr.press/v97/song19d.html>
- [21] Z. Zhang, W. Yu, M. Yu, Z. Guo, and M. Jiang, “A survey of multi-task learning in natural language processing: Regarding task relatedness and training methods,” in *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2023, Dubrovnik, Croatia, May 2-6, 2023*. Association for Computational Linguistics, 2023, pp. 943–956. [Online]. Available: <https://doi.org/10.18653/v1/2023.eacl-main.66>
- [22] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever *et al.*, “Improving language understanding by generative pre-training,” 2018.
- [23] J. Devlin, M. Chang, K. Lee, and K. Toutanova, “BERT: pre-training of deep bidirectional transformers for language understanding,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, 2019, pp. 4171–4186. [Online]. Available: <https://doi.org/10.18653/v1/n19-1423>
- [24] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, “Exploring the limits of transfer learning with a unified text-to-text transformer,” *J. Mach. Learn. Res.*, vol. 21, pp. 140:1–140:67, 2020. [Online]. Available: <http://jmlr.org/papers/v21/20-074.html>
- [25] S. Wu, X. Li, F. Yu, and M. Sun, “Tunesformer: Forming irish tunes with control codes by bar patching,” in *Proceedings of the 2nd Workshop on Human-Centric Music Information Retrieval 2023 co-located with the 24th International Society for Music Information Retrieval Conference (ISMIR 2023), Milan, Italy, November 10, 2023*, ser. CEUR Workshop Proceedings, vol. 3528. CEUR-WS.org, 2023. [Online]. Available: <https://ceur-ws.org/Vol-3528/paper1.pdf>
- [26] L. Casini, N. Jonason, and B. L. Sturm, “Generating folk-like music in abc-notation with masked language models,” in *Ismir 2023 Hybrid Conference*, 2023.
- [27] R. Yuan, H. Lin, Y. Wang, Z. Tian, S. Wu, T. Shen, G. Zhang, Y. Wu, C. Liu, Z. Zhou, Z. Ma, L. Xue, Z. Wang, Q. Liu, T. Zheng, Y. Li, Y. Ma, Y. Liang, X. Chi, R. Liu, Z. Wang, P. Li, J. Wu, C. Lin, Q. Liu, T. Jiang, W. Huang, W. Chen, E. Benetos, J. Fu, G. Xia, R. B. Dannenberg, W. Xue, S. Kang, and Y. Guo, “Chatmusician: Understanding and generating music intrinsically with LLM,” *CoRR*, vol. abs/2402.16153, 2024. [Online]. Available: <https://doi.org/10.48550/arXiv.2402.16153>
- [28] B. L. Sturm, J. F. Santos, O. Ben-Tal, and I. Korshunova, “Music transcription modelling and composition using deep learning,” *CoRR*, vol. abs/1604.08723, 2016. [Online]. Available: <http://arxiv.org/abs/1604.08723>

- [29] C. Geerlings and A. Meroño-Peñuela, “Interacting with gpt-2 to generate controlled and believable musical sequences in abc notation,” in *NLP4MUSA*, 2020. [Online]. Available: <https://api.semanticscholar.org/CorpusID:227217204>
- [30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA, 2017*, pp. 5998–6008. [Online]. Available: <https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html>
- [31] “Abc notation,” <http://abcnotation.com/>, accessed: 2024-04-12.
- [32] “Folkwiki,” <http://www.folkwiki.se/>, accessed: 2024-04-12.
- [33] “Chord-conditioned melody harmonization with controllable harmonicity [icassp 2023],” <https://github.com/sander-wood/deepchoir>, accessed: 2024-04-12.
- [34] “Kernscores,” <http://kern.ccarh.org/>, accessed: 2024-04-12.
- [35] “The meertens tune collections,” <https://www.liederenbank.nl/mtc/>, accessed: 2024-04-12.
- [36] “The nottingham music database,” <https://ifdo.ca/~seymour/nottingham/nottingham.html>, accessed: 2024-04-12.
- [37] “Openscore lieder corpus,” <https://musescore.com/openscore-lieder-corpus>, accessed: 2024-04-12.
- [38] “The session,” <https://thesession.org/>, accessed: 2024-04-12.
- [39] I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” in *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. [Online]. Available: <https://openreview.net/forum?id=Bkg6RiCqY7>
- [40] “wikimusictext Dataset on Hugging Face Datasets,” <https://huggingface.co/datasets/sander-wood/wikimusictext>, accessed: 2024-04-01.
- [41] “Download for Wikifonia all 6,675 Lead Sheets - Synth Zone Forum,” http://www.synthzone.com/forum/ubbthreads.php/topics/384909/Download_for_Wikifonia_all_6,6, accessed: 2024-04-01.
- [42] “chord-melody-dataset on GitHub,” <https://github.com/shiehn/chord-melody-dataset>, accessed: 2024-04-01.
- [43] “OpenEWLD on GitHub,” <https://github.com/00sapo/OpenEWLD>, accessed: 2024-04-01.
- [44] “KernScores: Essen Folksong Collection,” <http://kern.ccarh.org/cgi-bin/ksbrowse?l=/essen>, accessed: 2024-04-01.
- [45] “KernScores: Erk’s Liederschatz,” <https://kern.humdrum.org/cgi-bin/browse?l=users/craig/songs/erk/liederschatz>, accessed: 2024-04-01.
- [46] S. Rhyu, H. Choi, S. Kim, and K. Lee, “Translating melody to chord: Structured and flexible harmonization of melody with transformer,” *IEEE Access*, vol. 10, pp. 28 261–28 273, 2022. [Online]. Available: <https://doi.org/10.1109/ACCESS.2022.3155467>
- [47] Y. Yeh, W. Hsiao, S. Fukayama, T. Kitahara, B. Genchel, H. Liu, H. Dong, Y. Chen, T. Leong, and Y. Yang, “Automatic melody harmonization with triad chords: A comparative study,” *CoRR*, vol. abs/2001.02360, 2020. [Online]. Available: <http://arxiv.org/abs/2001.02360>

GRAPHMUSE: A LIBRARY FOR SYMBOLIC MUSIC GRAPH PROCESSING

Emmanouil Karystinaios¹

Gerhard Widmer^{1,2}

¹ Institute of Computational Perception, Johannes Kepler University Linz, Austria

² LIT AI Lab, Linz Institute of Technology, Austria

firstname.lastname@jku.at

ABSTRACT

Graph Neural Networks (GNNs) have recently gained traction in symbolic music tasks, yet a lack of a unified framework impedes progress. Addressing this gap, we present GraphMuse, a graph processing framework and library that facilitates efficient music graph processing and GNN training for symbolic music tasks. Central to our contribution is a new neighbor sampling technique specifically targeted toward meaningful behavior in musical scores. Additionally, GraphMuse integrates hierarchical modeling elements that augment the expressivity and capabilities of graph networks for musical tasks. Experiments with two specific musical prediction tasks – pitch spelling and cadence detection – demonstrate significant performance improvement over previous methods. Our hope is that GraphMuse will lead to a boost in, and standardization of, symbolic music processing based on graph representations. The library is available at <https://github.com/manoskary/graphmuse>

1. INTRODUCTION

Symbolic music processing entails the manipulation of digital music scores, encompassing various formats such as MusicXML, MEI, Humdrum, **kern, and MIDI. Unlike audio-based representations, symbolic formats offer granular information on note elements, including onset, pitch, duration, and other musical attributes like bars and time signatures.

While prior research in symbolic music processing often adopted techniques from the image processing [1–3] or natural language processing [4–6] domains, recent attention has shifted towards graph-based models, which could presumably better capture the dual sequential and hierarchical nature of music. Graph Neural Networks (GNNs) have been showcased as potent tools for diverse symbolic music tasks, including cadence detection [7], optical music recognition [8], music generation [9], Roman numeral analysis [10], composer classification [11], voice separation [12], and expressive performance rendering [13].

However, a standardized framework for constructing and processing music graphs has not yet been introduced to the field. To address this challenge, we developed GraphMuse, a Python-based framework to efficiently and effectively process information from musical scores, construct musically meaningful graphs, and facilitate the training of graph-based models for symbolic music tasks.

A key innovation of our work lies in the introduction of a new sampling technique tailored to specific properties of music while maintaining efficient and robust training of GNNs. Additionally, GraphMuse integrates within the graphs and models hierarchical elements that augment the capabilities of graph networks for musical tasks.

We evaluate our framework on pitch spelling and cadence detection tasks, comparing it against existing state-of-the-art methods. Through the synergistic utilization of our framework’s components, we achieve a significant performance increase compared to the previous methods. Our overarching objective is to establish a standardized framework for graph processing in symbolic music analysis, thus catalyzing further progress in the field.

Altogether, our contributions are three-fold: i) We provide a structured, generic, and flexible framework for graph-based music processing; ii) we release an open source *Python* library that comes with it; iii) we achieve performance improvements in a principled way by focusing on the design of the individual parts of the framework.

2. PROCESSING MUSIC SCORES WITH GNNs

In this section, we describe existing graph modeling approaches for musical scores. They all have a common pipeline which involves building a graph from a given musical score (see Figure 1) and using a series of convolutional blocks to produce context-aware hidden representations for each node. We start by describing the graph-building procedure and a generic graph convolutional block; we then take a detailed look at the problem of graph sampling, which will motivate a new sampling procedure that will be presented in the next section.

2.1 Preprocessing: Constructing Graphs from Scores

A score graph can be represented as a heterogeneous attributed graph. A heterogeneous graph has a type associated with each node and edge in the graph [14]. An attributed graph has an associated feature vector for



© E. Karystinaios and G. Widmer . Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** E. Karystinaios and G. Widmer , “GraphMuse: A Library for Symbolic Music Graph Processing”, in *Proc. of the 25th Int. Society for Music Information Retrieval Conf.*, San Francisco, United States, 2024.

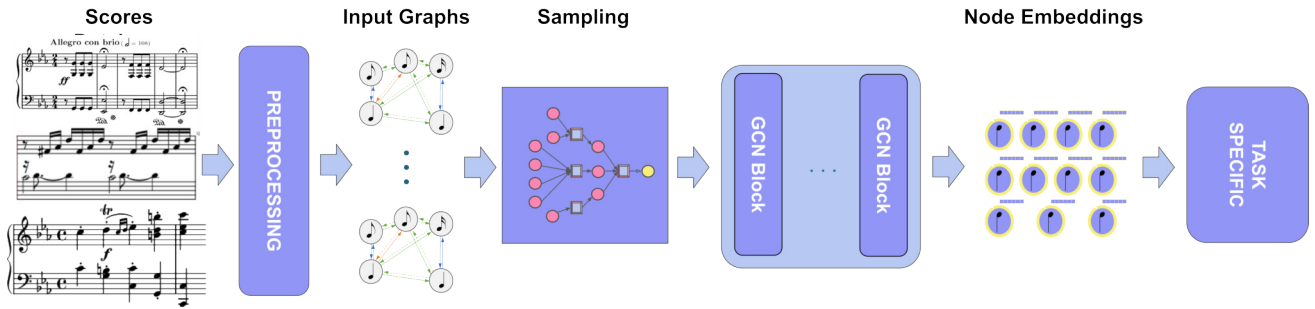


Figure 1. The general graph processing/training pipeline for symbolic music scores involves several steps: i) Preprocess the database of scores to generate input graphs; ii) Sample the input graphs to create memory-efficient batches; iii) Form a batch as a new graph with nodes and edges from various input graphs; iv) Sample a subset of nodes (target nodes) and their neighbors from the input graphs; v) Update the target nodes’ representations through graph convolution to create node embeddings; vi) Use these embeddings for task-specific applications. Note that target nodes may include all or a subset of batch nodes depending on the sampling strategy.

each node in the graph [15]. Therefore, a heterogeneous attributed graph is defined by a quintuple $G = (V, E, X, \mathcal{A}, \mathcal{R})$, together with the mappings $\phi : V \rightarrow \mathcal{A}$ and $\psi : E \rightarrow \mathcal{R}$, where V is the set of nodes, E is the set of edges, $X \in V \times \mathbb{R}^k$ the feature matrix \mathcal{A} is the node types and \mathcal{R} is the edge types. ϕ maps each node to its type and ψ maps its each edge to its corresponding type.

We create such a graph from a musical score by following previous work [10–13]. Each node $v \in V$ corresponds to one and only one note in the musical score. \mathcal{R} includes 4 types of relations: onset, during, follow, and silence, corresponding, respectively, to two notes starting at the same time, a note starting while the other is sounding, a note starting when the other ends, and a note starting after a time when no note is sounding. The inverse edges for during, follows, and silence relations are also created.

Formally, let us consider three functions $on(v)$, $dur(v)$, and $pitch(v)$ defined on a note $v \in V$ that extract the onset time, duration, and pitch, respectively. A typed edge (u, r, v) of type $r \in \mathcal{R}$ between two notes $u, v \in V$ belongs to E if the following conditions are met:

- $on(u) = on(v) \rightarrow r = \text{onset}$
- $on(u) > on(v) \wedge on(u) \leq on(v) + dur(v) \rightarrow r = \text{during}$
- $on(u) + dur(u) = on(v) \rightarrow r = \text{follow}$
- $on(u) + dur(u) < on(v) \wedge \nexists v' \in V, on(v') < on(v) \wedge on(v') > on(u) + dur(u) \rightarrow r = \text{silence}$

\mathcal{A} in the literature usually only includes a single type, i.e. the note type ν . However, we extend this definition in Section 3.1.

2.2 Encoding: Graph Convolution

Graph convolution and message passing are core operations in graph neural networks (GNNs) for learning node representations. In graph convolution, in its simplest form, each node aggregates messages from its immediate neighbors by computing a weighted sum of their features:

$$\mathbf{h}_v^{(l+1)} = \sigma \left(\left(\sum_{u \in \mathcal{N}(v)} \mathbf{W}^{(l)} \mathbf{h}_u^{(l)} \right) + \mathbf{h}_v^{(l)} \right) \quad (1)$$

where $\mathbf{h}_v^{(l)}$ is the representation of node v at layer l , $\mathcal{N}(v)$ denotes the neighbors of node v , $\mathbf{W}^{(l)}$ is a learnable weight, and σ is a non-linear activation function. Through successive iterations of message passing and aggregation, each node refines its representation by incorporating information from increasingly distant nodes in the graph, ultimately enabling the network to capture complex relational patterns and dependencies within the graph data.

In the context of music, graph convolution can be understood as a method for defining a note not only by its own characteristics and properties but by also considering the characteristics of its neighboring notes within the musical graph. In this work, as well as previous graph-based work on music [7, 10, 11] the preferred graph convolutional block is *SageConv* taken from one of the first and fundamental works in graph deep learning [16].

2.3 Sampling: Handling Graph Data for Training

In an ideal world without computing resource considerations, we can imagine a training pipeline that receives an entire graph as input to a graph convolutional model. Assuming that we have the resources and time to perform such a task the process is easy to grasp. All nodes of the graph are updated in a single step based on their neighbors as described in the previous section.

However, the graph world presents us with several complexity issues. Graph datasets in the wild typically come in two forms: i) a (possibly large) collection of small graphs, each containing maybe fewer than 50 nodes [15]; ii) a single large-scale graph such as a social network [17], a recommender system [18], or a knowledge graph [19]. The previous naive scenario presents a time-efficiency and computation waste bottleneck for the former and a memory insufficiency issue for the latter. To mitigate these issues, in the former case one can batch many small graphs

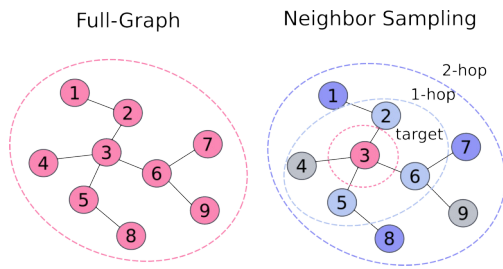


Figure 2. Full graph vs neighbor sampling. The pink-colored nodes are selected for convolution by message passing. With neighbor sampling, the pink node is the one whose representation is ultimately updated after convolution (however, for the blue nodes also take part in the convolution process as its context).

together to maximize the available resources and reduce the computation time, then the full graphs can be updated during convolution within each batch.

Training Graph Convolutional Networks (GCNs) for large-scale graphs is a bit more complicated. Such graphs can be exceptionally large – for example, the 2019 Facebook social network boasted 3.51 billion users¹. To train models with such graphs we need to devise a sampling algorithm to derive subgraphs in steps [16, 20–22]. Such an algorithm may, for example, choose a subset of nodes across the graph and perform random walks to fetch a subset of the k -hop neighbors for the sampled nodes [16]. This process, called *neighbor sampling* or *node-wise sampling*, is shown in Figure 2 and compared to the full-graph process.

Musical score graphs fall in between the two scenarios, varying notably in size. For instance, a Bach Chorale might contain 100 notes, while a Beethoven Sonata could exceed 5000 notes, with each note corresponding to a graph node. Furthermore, a musical dataset may contain many such graphs. Therefore the question arises how to efficiently train models on music graph datasets.

Since music graphs are not uniform enough to be batched together like small graph datasets, we investigate the suitability of neighbor sampling methods for music graph processing, taking into account special properties relevant in music. Standard neighborhood sampling would sample notes across different scores and fetch neighbors for those notes, creating a subgraph that can maximize the use of the available resources during training.

However, music has a specific coherence, in both the horizontal (time) and vertical (chords, harmonies) dimensions, which makes sampling approaches from the literature [22] not appropriate for music. Specifically, sampling and updating/encoding single notes without simultaneously doing so also to notes in their local context makes it difficult to learn properties that persist in time (such as local key or a harmonic function). In this work, we address this issue by presenting a simple and musically intuitive sampling process for graphs that efficiently creates batches containing musically related notes which, as experiments

will show, can notably improve the learning results.

2.4 Task-specific Modeling

Finally, the node embeddings created by the graph convolutional encoder serve as input to task-specific models that solve some specific prediction or recognition task. In a graph context, we distinguish, at an abstract level, between node classification, link prediction, and entire graph classification tasks. Examples of node classification tasks can be found in [7] which takes the embeddings from the GCN encoder and employs an edge decoder coupled with a graph convolution classifier for cadence prediction labels; and in [10], which forwards the embeddings to sequential layer and then MLP classifiers to perform Roman Numeral Analysis. In [12], musical voice separation is framed as a link prediction task; the node embeddings are input to a pairwise edge similarity encoder to predict link probabilities between notes in the same voice. An example of a graph classification task can be found in [11] where the embeddings are aggregated and passed through a shallow MLP for composer classification.

Naturally, task-specific models will not be part of the generic graph processing pipeline and library which we publish with this paper.

3. METHODOLOGY

In this section, we discuss our approach to addressing the different components of the pipeline shown in Figure 1. In particular, we explain the preprocessing procedure for creating score graphs, we detail our strategy for musically intuitive graph sampling, and finally, we discuss model variants that are made possible by the previous steps of the pipeline.

3.1 Preprocessing

The central activity in the preprocessing step is the creation of graphs from musical scores. In our library, we extend the conventional graph creation process by introducing hierarchical musical dimensions (beats and measures), in order to enhance the score graphs’ representational capacity. More specifically, we enrich the node type set \mathcal{A} (defined in Section 2.1) with two additional types β and μ for beats and measures respectively. The process involves detecting beats and measures within the musical score, generating edges (of type *connect* to every beat from each note falling within its temporal boundaries, and repeating this process for measures. Additional edges of type *next* are drawn between consecutive beats and measures to enrich the connectivity and contextual understanding within the graph. Furthermore, we aggregate features from constituent notes through the *connect* edges via message passing to equip each beat and measure with informative attributes by computing the mean vector of their note features.

The inclusion of beat and measure node elements, as well as the creation of inverse edges, are made optional, ensuring compatibility with diverse research needs and

¹ <https://zephoria.com/top-15-valuable-facebook-statistics>

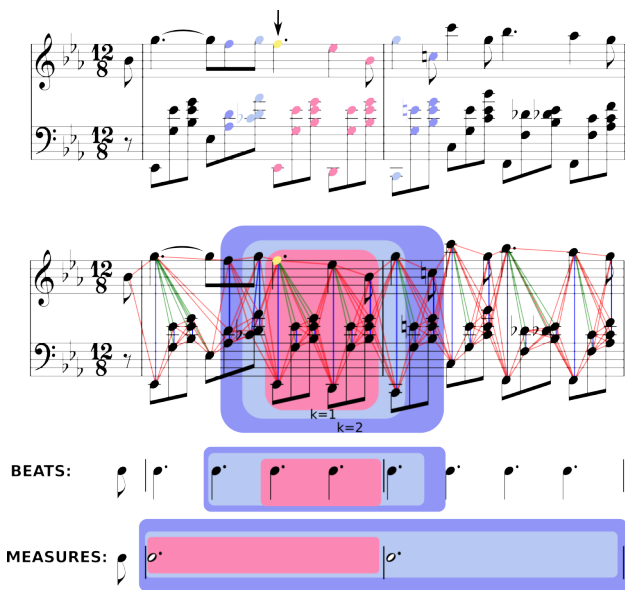


Figure 3. Sampling process per score. Top: sampled notes and their neighbors; middle: score graph and sampling process; bottom: sampling process for beats and measures. A randomly selected note (in yellow) is first sampled. The boundaries of the target notes are then computed with a budget of 15 notes in this example (pink and yellow notes). Then the k -hop neighbors are fetched for the targets (light blue for 1-hop and darker blue for 2-hop). The k -hop neighbors are computed with respect to the input graph (depicted with colored edges connecting noteheads in the figure). We can also extend the sampling process for the beat and measure elements (introduced in Section 3.1). Note that the k -hop neighbors need not be strictly related to a time window.

avoiding imposing rigid structures onto the graph-based music processing framework.

We prioritize the efficiency and speed of the graph creation process by transitioning the graph creation implementation to C code, leveraging its performance benefits, and establishing a Python binding for seamless integration into our workflow. Recognizing the temporal nature of musical elements, such as notes, beats, and measures, we refine our neighbor searching windows accordingly, optimizing computational efficiency.

3.2 Sampling

We discussed general neighbor sampling for large-scale graphs in Section 2.3 and some problems related to graph-structured music data. In this section, we elaborate on our musically informed sampling process for music graphs, which enables the training of the models outlined in the subsequent sections. In this process, we aim to sample sections of scores and employ neighbor sampling to fetch the neighbors of notes within those sections.

Indeed while our nodes could be ordered in various ways, the most perceptually significant aspect is time organization. Recognizably, individuals can still identify a musical piece when segmented along the time axis,

whereas focusing solely on pitch intervals may be challenging. Moreover, perceptual research indicates that the commencement time of a note holds greater salience than its offset time, particularly for percussive instruments like the piano, where the sound naturally fades over time [23]. Hence, when constructing graphs from musical scores, we prioritize node arrangement based on absolute onset time followed by pitch.

Our initial limitations are mostly related to memory usage. To limit our memory we need to predefine three initial arguments: i) the size of each target subgraph S from every score, ii) the number B of scores in each batch, and iii) the number of hops and neighbors for each hop (similar to node-wise sampling techniques). In each batch, we update the representation of our target nodes which is essentially the size of $S \times B$.

Once the ordering is set and the three arguments are defined we can initiate the process of sampling a subgraph, as shown in Figure 3. First, we sample a random note from the graph of each score. Next, we correct the position of the note by searching for any vertical neighbors (same onset value notes and potentially different pitch). Then we extend to S notes to the right where S indicates a predefined maximum subgraph size. We also correct the rightmost boundaries to include or exclude vertical neighbors for the last onset always respecting the aforementioned size S . Once this process is completed we obtain the target nodes per score within the batch. These are the nodes whose representation we want to update at the end of the graph convolutional process.

However, since graph convolution is performed recursively we need to fetch the k -hop neighbors for each one of the target nodes where k indicates the depth of the GCN. For this step, we can consult the literature [16] and perform neighbor sampling to fetch the k -hop neighbors. This process is repeated for B different scores. Finally, the B score subgraphs of size at most S each are first joined together and then fed to the model.

During this process, we can keep information about the target nodes and the size of each score subgraph, which could allow us to design more creative models that can exploit this information. Such models are presented in the next section. Moreover, we adopt a potential approach for hierarchical graphs by also extending the sampling for beat and measure nodes as shown in Figure 3.

3.3 Model Designs

In this section, we explore various model designs for the graph-based encoder in our processing pipeline (Figure 1). Designing such an encoder involves addressing two fundamental questions: the selection of graph convolutional blocks and the selective exploitation of information from the input graph.

The first question, regarding graph convolutional blocks, remains open-ended, offering numerous possibilities for exploration and customization. In its current version, *GraphMuse* offers the options of convolutional blocks on a per-node or per-edge type basis. We sug-

gest that graph-attention networks may offer promising avenues, particularly for hierarchical elements such as beats or measures.

In response to the second question, we devise a series of models by selectively incorporating or excluding elements from the input graph. Our foundational model, termed *NoteGNN*, exclusively utilizes note elements and their corresponding edges. This model serves as the basis for further extension. For instance, we expand upon *NoteGNN* to construct *BeatGNN*, which incorporates beat elements (see Section 3.1 above) alongside notes. Similarly, we develop *MeasureGNN* by integrating measures into the encoding process. When all note, measure, and beat elements are included, the resultant model is denoted as *MetricalGNN*.

Furthermore, we explore the possibility of hybridizing model types, such as combining GNNs with sequential models. This hybridization is facilitated by the sampling process that organizes notes in onset order, allowing for the batch to be unfolded by score. Consequently, the same batch can be processed through both GNN and sequential models simultaneously. Specifically, we employ a graph encoder and a sequential encoder in parallel – in our case we use a stack of 2 bidirectional GRU layers. The GRU stack receives the unfolded batch of size (B, S, K) where B is the number of scores within the batch, S is the number of sampled target nodes for each score order by onset and then by pitch, and K is the number of node features. The embeddings of both encoders are concatenated together and an additional linear layer is applied to project them to the required dimension.

This architecture, which we call *HybridGNN* in our experiments, combines the strengths of both GNNs and sequential models, resulting in better performance as shown in our experiments.

3.4 The Library

The components discussed in the preceding section have been implemented and made available in an open-source Python library called GraphMuse. This library follows a similar philosophy as PyTorch and PyTorch Geometric, comprising models and graph convolutional blocks, loader pipelines, data pipelines, and related utilities. GraphMuse is built upon and thus requires PyTorch and PyTorch Geometric. The loaders and models provided by GraphMuse are fully compatible with those of PyTorch Geometric. For musical input and output, GraphMuse is compatible with Partitura [24], a Python library for symbolic music processing, allowing it to work with a variety of input formats such as MusicXML, MEI, Humdrum `**kern`, and MIDI.

4. EVALUATION

To evaluate our framework, we perform experiments on two tasks, cadence detection and pitch spelling. We put to the test both the models discussed as well as the sampling process. For pitch spelling, we compare our models to the previous sequential state-of-the-art model, PKSpell [25] and the GraphSAGE variant of our note-level model. For

cadence detection, we compare our models to the previous state-of-the-art model by Karystinaios and Widmer [7] which is also graph-based and follows a GraphSAGE sampling strategy. For both tasks, we perform ablations by removing the hierarchical elements such as beat and measure nodes and edges, or incorporating hybrid models. This work focuses on the application of the GraphMuse library therefore, a detailed comparison of various input encodings and architectures, as conducted by [11], is beyond the scope of this paper.

4.1 Pitch Spelling

Previous work on Pitch Spelling set the state-of-the-art by using a sequential model [25]. The task of pitch spelling tackles in parallel key signature estimation and pitch spelling estimation per note, however, the key signature is a global attribute usually set for the whole piece although it can sometimes change midway. The previous architecture uses a GRU encoder for pitch spelling and then infuses the logits together with the latent representation to another GRU layer for the key signature prediction.

For our approach, we use a GNN encoder as described in Section 3.3 followed by two classification heads for key and pitch spelling respectively. We train and evaluate all models on the ASAP dataset [26] using a random split with 15% of the data for testing and the 85% for train and validation as described in [25].

4.2 Cadence Detection

For the cadence detection model, we chose to use a modified version of the cadence detection model originally proposed in [7]. Our considerations were based on a more efficient training process, and the integration of our pipeline possibilities. The model was expanded to accept a heterogeneous score graph as input, as described in Section 2.1. Additionally, we enhanced the model’s predictive capabilities from binary (no-cad or PAC) to multiclass cadence prediction, encompassing PAC, IAC, and HC labels. Furthermore, we refined the architecture by incorporating an onset regularization module, which aggregates the latent representations (post-GNN encoder) of all notes occurring at a distinct onset within the score and assigns them to every note sharing that onset.

In the training phase, the input graph first undergoes processing through the graph encoder. The resulting node embeddings are then grouped based on onset information extracted from the score, and their representations are averaged. Subsequently, embedded SMOTE [27] is applied to balance the distribution of cadence classes compared to the notes lacking cadence labels in the score. However, during inference, this synthetic oversampling step is omitted. Finally, the oversampled embeddings are fed into a shallow 2-layer MLP classifier to predict the cadence type.

We trained our model with a joined corpus of cadence annotations from the DCML corpora², the Bach fugues

² https://github.com/DCMLab/dcml_corpora

from the well-tempered clavier Book No.1 [28], the annotated Mozart string quartets [29], and the annotated Haydn string quartets [30]. Our joined corpus makes for 590, 149 individual notes and 17, 188 cadence annotations. We use 80% of the data for training and validation and test on 20% using a random split. Note that these results cannot be directly compared with [7] since we use a different (bigger) dataset and perform multiclass prediction.

4.3 Experiments

4.3.1 Configuration

The configuration for training pitch spelling graph models with our sampling technique uses a batch size $B = 300$, sampling from 300 scores at each step, and target node size $S = 300$. For cadence graph models, $B = 200$ and $S = 500$. All graph models, including GraphSAGE, utilize three heterogeneous SageConv layers with a hidden size of 256 and a dropout of 0.5. Neighbor sampling for each layer fetches up to three neighbors per sampled node per relation. We train all models with the Adam optimizer (learning rate 10^{-3} , weight decay 5×10^{-4}) on a GTX 1080 Ti. Each experiment is repeated at least four times with different random seeds, and statistical significance testing is performed using the ASO method at a confidence level $\alpha = 0.05$ [31]³.

4.3.2 Results

Table 1 presents the results of experiments conducted on the two tasks. The metrics used for evaluation are Accuracy (A) for pitch spelling and key recognition, and the macro F1 score ($F1$) for cadence detection. Note that the model employed on the GraphSAGE methods and the model NoteGNN are virtually the same apart from the sampling strategy with which they were trained.

For the pitch spelling task, we can observe that the actual pitch spelling accuracy (A-Pitch) of all proposed models surpasses both the PKSpell and GraphSAGE methods. Across all models, the MetricalGNN achieves the highest accuracy of 95.6%, closely followed by BeatGNN and MeasureGNN with accuracies of 95.1% and 95.4%, respectively. These results indicate the benefits of incorporating hierarchical musical elements such as beats and measures. However, it is worth noting that while MetricalGNN achieves the highest accuracy, it is closely followed by the hybrid model, HybridGNN, which achieves an accuracy of 95.4%, suggesting that competitive performance can also be achieved by mixing model types.

Focusing on the key estimation subtask (A-Key) of pitch spelling we notice that the PKSpell model achieves a very good key accuracy of 69.9%, closely followed by the MeasureGNN model and only surpassed by the Hybrid model. We attribute the effectiveness of key detection of a sequential model such as PKSpell to the persistence of the key label across elements of the sequence. Therefore, a hybrid model in this case seems to be able to adapt to

Task	Pitch Spelling		Cadence
	A-Pitch	A-Key	F1-Cad
PKSpell	94.8 ± 0.5	69.9 ± 1.6	-
GraphSAGE	93.6 ± 0.1	43.3 ± 0.1	53.5 ± 0.8
NoteGNN	94.9 ± 0.1	69.3 ± 7.0	55.3 ± 0.9
BeatGNN	95.1 ± 0.2	68.7 ± 1.1	<u>57.4</u> ± 1.2
MeasureGNN	<u>95.4</u> ± 0.3	69.5 ± 7.2	57.0 ± 1.0
MetricalGNN	95.6 ± 0.1	64.4 ± 5.3	55.8 ± 0.6
HybridGNN	<u>95.4</u> ± 0.2	72.6 ± 2.8	58.6 ± 0.7

Table 1. Results on the two tasks, in terms of accuracy (A) and F1 score, respectively. Values in bold are the best score per metric; underlined values are the second best. All runs are repeated 4 times. \pm indicates standard deviation.

the diversity of labels for pitch spelling and uniformity of labels for key estimation. We found our best model to be stochastically dominant over PKSpell with $min_e = 0.17$.

In the cadence detection task, we evaluate the results using the macro F1 score to account for the overwhelming presence of non-cadence nodes, as instructed by [7]. We observe that GraphSAGE, the previously used technique for training, obtains the lowest F1 score and it is surpassed by all the proposed GNN-based models trained with the new sampling method.

Among our GNN models, BeatGNN and HybridGNN achieve the highest scores of 57.4% and 58.6%, respectively, closely followed by MeasureGNN. In this case, the MetricalGNN model surprisingly does not achieve such a good score even though it includes both measure and beat elements. However, it still performs better than the NoteGNN and the GraphSAGE method.

Overall, the results demonstrate the efficacy of GNN-based models trained using our new sampling method. Incorporating hierarchical elements such as beats and measures improves both pitch spelling and cadence detection tasks. Additionally, the hybrid approach of combining GNNs with sequential models produces promising results.

5. CONCLUSION

In this paper, we introduced GraphMuse, a framework and Python library for symbolic graph music processing. We designed a specialized sampling process for musical graphs and demonstrated our pipeline’s effectiveness through experiments on pitch spelling and cadence detection. Our results show that carefully designed GNN architectures, especially those incorporating hierarchical elements like beats and measures, can lead to better performance. Finally, hybrid models that integrate GNNs with sequential models yield further performance improvements.

Future research will focus on refining GNN-based models in music processing, adding more tasks, and exploring novel architectures. This includes investigating advanced graph convolutional blocks, other sampling techniques, and attention mechanisms to enhance model performance.

³For the detailed experiments visit: <https://wandb.ai/melkisedeath/GraphMuse>

6. ACKNOWLEDGEMENTS

The authors would like to thank Nimrod Varga for his contribution to accelerating graph creation by adapting it to C code. This work is supported by the European Research Council (ERC) under the EU’s Horizon 2020 research & innovation programme, grant agreement No. 101019375 (*Whither Music?*), and the Federal State of Upper Austria (LIT AI Lab).

7. REFERENCES

- [1] E. Benetos, A. Klapuri, and S. Dixon, “Score-informed transcription for automatic piano tutoring,” in *European Signal Processing Conference (EUSIPCO)*, 2012.
- [2] G. Velarde, T. Weyde, C. E. Cancino-Chacón, D. Meredith, and M. Grachten, “Composer recognition based on 2D-filtered piano-rolls,” in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2016.
- [3] N. N. López, M. Gotham, and I. Fujinaga, “Augmentednet: A roman numeral analysis network with synthetic training examples and additional tonal tasks,” in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2021, pp. 404–411.
- [4] N. Fradet, J.-P. Briot, F. Chhel, A. El Fallah Seghrouchni, and N. Gutowski, “MidiTok: A python package for MIDI file tokenization,” in *Extended Abstracts for the Late-Breaking Demo Session of the 22nd International Society for Music Information Retrieval Conference*, 2021. [Online]. Available: <https://archives.ismir.net/ismir2021/latebreaking/000005.pdf>
- [5] D. von Rütte, L. Biggio, Y. Kilcher, and T. Hofmann, “FIGARO: Generating Symbolic Music with Fine-Grained Artistic Control,” in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023.
- [6] J. Liu, Y. Dong, Z. Cheng, X. Zhang, X. Li, F. Yu, and M. Sun, “Symphony Generation with Permutation Invariant Language Model,” in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2022.
- [7] E. Karystinaios and G. Widmer, “Cadence detection in symbolic classical music using graph neural networks,” in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2022.
- [8] A. Baró, P. Riba, and A. Fornés, “Musigraph: Optical music recognition through object detection and graph neural network,” in *International Conference on Frontiers in Handwriting Recognition*. Springer, 2022, pp. 171–184.
- [9] E. Cosenza, A. Valenti, and D. Bacciu, “Graph-based polyphonic multitrack music generation,” in *International Joint Conference on Artificial Intelligence (IJCAI)*, 2023.
- [10] E. Karystinaios and G. Widmer, “Roman Numeral Analysis with Graph Neural Networks: Onset-wise Predictions from Note-wise Features,” in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2023.
- [11] H. Zhang, E. Karystinaios, S. Dixon, G. Widmer, and C. E. Cancino-Chacón, “Symbolic music representations for classification tasks: A systematic evaluation,” in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2023.
- [12] E. Karystinaios, F. Foscari, and G. Widmer, “Musical Voice Separation as Link Prediction: Modeling a Musical Perception Task as a Multi-Trajectory Tracking Problem,” in *International Joint Conference on Artificial Intelligence (IJCAI)*, 2023.
- [13] D. Jeong, T. Kwon, Y. Kim, and J. Nam, “Graph neural network for music score data and modeling expressive piano performance,” in *Proceedings of the International Conference on Machine Learning (ICML)*. PMLR, 2019, pp. 3060–3070.
- [14] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, and P. S. Yu, “Heterogeneous graph attention network,” in *The world wide web conference*, 2019, pp. 2022–2032.
- [15] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, “A comprehensive survey on graph neural networks,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 1, pp. 4–24, 2021.
- [16] W. Hamilton, Z. Ying, and J. Leskovec, “Inductive representation learning on large graphs,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.
- [17] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.
- [18] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, “Graph convolutional neural networks for web-scale recommender systems,” in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2018, pp. 974–983.
- [19] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. Van Den Berg, I. Titov, and M. Welling, “Modeling relational data with graph convolutional networks,” in *The semantic web: 15th international conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, proceedings 15*. Springer, 2018, pp. 593–607.

- [20] D. Zou, Z. Hu, Y. Wang, S. Jiang, Y. Sun, and Q. Gu, "Layer-dependent importance sampling for training deep and large graph convolutional networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.
- [21] H. Zeng, H. Zhou, A. Srivastava, R. Kannan, and V. Prasanna, "GraphSAINT: Graph sampling based inductive learning method," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020.
- [22] X. Liu, M. Yan, L. Deng, G. Li, X. Ye, and D. Fan, "Sampling methods for efficient training of graph convolutional networks: A survey," *IEEE/CAA Journal of Automatica Sinica*, vol. 9, no. 2, pp. 205–234, 2021.
- [23] A. Klapuri and M. Davy, *Signal Processing Methods for Music Transcription*. Springer, 2006.
- [24] C. Cancino-Chacón, S. D. Peter, E. Karystinaios, F. Foscarin, M. Grachten, and G. Widmer, "Partitura: A python package for symbolic music processing," in *Music Encoding Conference (MEC)*, 2022.
- [25] F. Foscarin, N. Audebert, and R. Fournier-S'Niehotta, "Pkspell: Data-driven pitch spelling and key signature estimation," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2021.
- [26] S. D. Peter, C. E. Cancino-Chacón, F. Foscarin, A. P. McLeod, F. Henkel, E. Karystinaios, and G. Widmer, "Automatic note-level score-to-performance alignments in the ASAP dataset," *Transactions of the International Society for Music Information Retrieval (TISMIR)*, 2023.
- [27] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.
- [28] M. Giraud, R. Groult, E. Leguy, and F. Levé, "Computational fugue analysis," *Computer Music Journal*, vol. 39, no. 2, pp. 77–96, 2015.
- [29] P. Allegraud, L. Bigo, L. Feisthauer, M. Giraud, R. Groult, E. Leguy, and F. Levé, "Learning sonata form structure on Mozart's string quartets," *Transactions of the International Society for Music Information Retrieval (TISMIR)*, vol. 2, no. 1, pp. 82–96, 2019.
- [30] D. R. Sears, M. T. Pearce, W. E. Caplin, and S. McAdams, "Simulating melodic and harmonic expectations for tonal cadences using probabilistic models," *Journal of New Music Research*, vol. 47, no. 1, pp. 29–52, 2018.
- [31] R. Dror, S. Shlomov, and R. Reichart, "Deep dominance - how to properly compare deep neural models," in *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, A. Korhonen, D. R. Traum, and L. Màrquez, Eds. Association for Computational Linguistics, 2019, pp. 2773–2785. [Online]. Available: <https://doi.org/10.18653/v1/p19-1266>

Papers – Session V

ST-ITO: CONTROLLING AUDIO EFFECTS FOR STYLE TRANSFER WITH INFERENCE-TIME OPTIMIZATION

Christian J. Steinmetz Shubhr Singh Marco Comunità Ilias Ibyhahya
 Shanxin Yuan Emmanouil Benetos Joshua D. Reiss
 Centre for Digital Music, Queen Mary University of London, UK

ABSTRACT

Audio production style transfer is the task of processing an input to impart stylistic elements from a reference recording. Existing approaches often train a neural network to estimate control parameters for a set of audio effects. However, these approaches are limited in that they can only control a fixed set of effects, where the effects must be differentiable or otherwise employ specialized training techniques. In this work, we introduce **ST-ITO**, Style Transfer with Inference-Time Optimization, an approach that instead searches the parameter space of an audio effect chain at inference. This method enables control of arbitrary audio effect chains, including unseen and non-differentiable effects. Our approach employs a learned metric of audio production style, which we train through a simple and scalable self-supervised pretraining strategy, along with a gradient-free optimizer. Due to the limited existing evaluation methods for audio production style transfer, we introduce a multi-part benchmark to evaluate audio production style metrics and style transfer systems. This evaluation demonstrates that our audio representation better captures attributes related to audio production and enables expressive style transfer via control of arbitrary audio effects.

1. INTRODUCTION

Audio effects are signal processing devices used to transform or manipulate audio signals, such as adding reverb, adjusting frequency balance with equalization, or adding edge with distortion. They play a central role in audio production, providing audio engineers with the ability to realize both practical and creative tasks with applications in music, film, broadcast, and video games [1]. Traditionally, operating these effects requires a significant amount of expertise, as audio engineers must combine a technical understanding with their artistic goals. As a result, the process of creating a high-quality audio production remains challenging, requiring a time-consuming process for professionals and a significant barrier for novices.

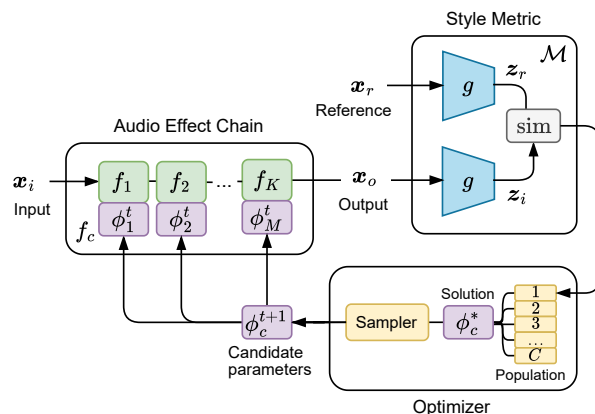


Figure 1. Style transfer with Inference-Time Optimization enables audio production style transfer through control of arbitrary audio effects. It employs a pretrained audio representation as a similarity metric, which is then optimized by searching the control parameter space of audio effects.

Intelligent music production aims to develop systems for automating aspects of audio engineering [2]. Early approaches employed rule-based systems, using hand-engineered rules based on best practices [3]. These systems often generated outputs that satisfied certain assumptions or utilized well established conventions. However, the inability to construct sufficient sophisticated rule bases has motivated machine learning approaches, which instead learn from data without assuming a limited or fixed set of rules [4–7]. Nevertheless, these systems still lacked the ability to adapt based on user input, which is critical to the context-dependent nature of music production [8].

To address the context-dependent nature of this task and enable greater user control, *audio production style transfer* has been proposed [9, 10]. These systems rely on a reference recording and attempt to map elements of the audio production style from the reference onto the input. These systems either directly process the audio signal [11–13] or estimate parameters for audio effects [9, 10, 14, 15]. While direct transformation methods are powerful, they may introduce artifacts and lack grounding in traditional audio tools. Similarly, recent text-to-audio generation models also enable editing capabilities [16, 17], but suffer from the same limitations. On the other hand, parameter based methods enable efficient and controllable style transfer. However, current systems are limited to a fixed chain of effects, and require the use of differentiable signal processing [18], or inefficient alternative differentiation strategies such as gradient approximation [19] and neural proxies [7].

We propose a method to construct an audio production style transfer system that leverages inference-time optimization to facilitate real world applications. Instead of training a network to perform style transfer directly, we perform style transfer via an optimization process at inference, as shown in Figure 1. We iteratively search the parameter space of an effect chain with our proposed metric that measures the similarity in audio production style between the output recording and the reference. This approach enables the ability to control arbitrary audio effect chains, including non-differentiable effects, opening up the potential to control real-world audio effects. The contributions of our work are as follows:

- A simple and scalable pretraining strategy for constructing an audio production style similarity metric through audio effect estimation, named AFx-Rep.
- A system for audio production style transfer, ST-ITO, that optimizes the control parameters of arbitrary audio effects according to a similarity metric.
- An extension of the DeepAFx-ST system [14] with the addition of differentiable distortion and reverb, which forms a strong baseline.
- A multi-task benchmark for evaluation of audio production style similarity metrics and audio production style transfer systems.

We provide audio examples, and open source our datasets, benchmark, and code to facilitate reproducibility¹.

2. METHOD

In this work, we propose **ST-ITO**, Style Transfer with Inference-Time Optimization, a novel method for audio production style transfer that searches the parameter space of a set of audio effects to perform style transfer. As shown in Figure 1, our system features three main components: an audio effect chain that processes an input recording, an audio production style similarity metric, composed of pre-trained encoder and a similarity measure, and an optimizer that is used to find control parameters. This enables style transfer by finding a configuration of the audio effects that produce an output with attributes of the reference style.

Our approach provides a number of benefits as compared to previous audio production style transfer systems. First, it enables the control of arbitrary audio effects, even those that have not been seen during training. Unlike existing systems that train with a set of fixed effects “in-the-loop”, our approach enables adaptation to new effects at inference. Furthermore, our method removes requirements of previous systems. This includes removing the need for differentiable audio effects or alternative differentiation strategies, which can often be slow and difficult to train [14]. Finally, we provide further flexibility and control by enabling the addition or removal of audio effects at inference without re-training of the base model.

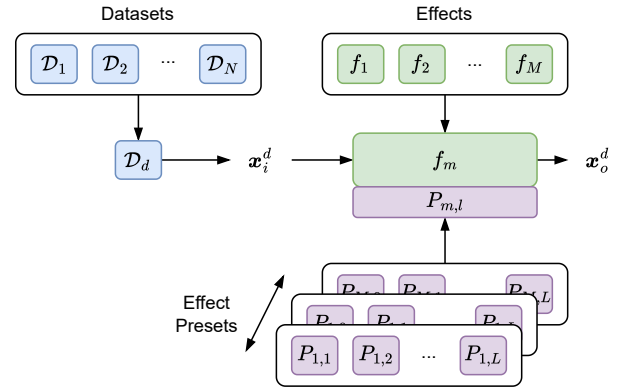


Figure 2. Self-supervised training for the pretext task where an audio signal $x_i^d \sim \mathcal{D}_d$ is sampled randomly from one of N datasets and then processed by a randomly sampled audio effect E_m with an associated randomly sampled parameter preset $P_{m,l}$ to produce an output signal x_o^d .

2.1 Audio production style metric

A central aspect of our approach is the development of an audio production style metric $\mathcal{M}(x_a, x_b)$. This metric measures the perceptual similarity in audio production between two recordings x_a and x_b . As explained in Section 2.2, we optimize this metric by searching the parameter space of a set of audio effects to align the style of the processed recording with the reference. In general, production style relates to aspects of audio quality rather than the underlying content, including attributes such as dynamics, frequency balance, and the stereo field [20].

Pretrained audio representations. There is a growing body of work in general purpose representations of audio signals [21], popular approaches include CLAP [22] and BEATs [23]. These representations capture relevant attributes to facilitate downstream tasks such as detection and classification of sound sources and events. While it may be possible to directly adapt one of these representations for our task, evidence suggests they are not always sensitive to audio effect transformations [24]. We provide further evidence for this in Section 5. This motivates us to develop a method to produce our own audio representation that is more sensitive to audio effect transformations.

Self-supervised pretext task. We propose a simple and scalable self-supervised pretext task to construct an audio representation for our task without human annotated data. To encourage the encoder to extract features related to audio effects we employ an audio effect classification task composed of two parts. The model predicts both which effect has been applied and the associated preset.

As shown in Figure 2, we generate training examples using N audio datasets, a set of M audio effects, and L associated parameter presets for each effect. To generate a training example, a dataset \mathcal{D}_d is selected at random from the set of datasets. Then one audio sample is selected from this dataset, which will form the input recording x_i^d . Next, we sample an audio effect f_m and a random associated preset $P_{m,l}$ to configure the effect. Then we process the input with this effect to produce an output signal x_o^d .

¹ <https://github.com/csteinmetz1/st-ito>

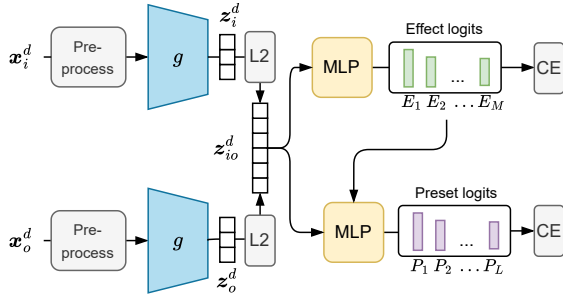


Figure 3. Representation learning via the pretext task where the input \mathbf{x}_i^d and output \mathbf{x}_o^d are processed by the encoder $g(\cdot)$ to produce embeddings. These embeddings are fed to a pair of MLP classifiers trained via cross-entropy that predict the effect class and preset class.

Training is shown in Figure 3 where the encoder $g(\cdot)$ extracts embeddings \mathbf{z}_i and \mathbf{z}_r from the input and output. These embeddings are concatenated and fed to a multi-layer perceptron (MLP) that estimates the effect applied. A second MLP takes the effect logits as well as the embeddings to estimate the preset. After pretraining we discard the prediction heads and use the encoder $g(\cdot)$, which we refer to as AFX-Rep, to produce embeddings.

This multi-task formulation encourages the encoder to extract features not only about effects but also subtleties between different configurations of the same effect, which is important for style transfer. Our approach does not enforce invariance to the content, but can leverage any audio, not only unprocessed or effect normalized audio [25], required by previous work [13]. This allows us to further scale the training dataset size. In addition, since this audio may already contain other processing, our model is exposed to a wide range of effects beyond those we apply.

2.2 Inference-time optimization

To perform style transfer we begin with input \mathbf{x}_i and reference \mathbf{x}_r recordings. We assume the input has minimal processing, as our system does not remove effects [26, 27]. Then, we require the user to provide an appropriate chain of audio effects to be controlled. This chain can be represented as the composition of K audio effects where each effect is represented by a function f_k parameterized by a control vector ϕ_k . The output \mathbf{x}_o is obtained by sequentially applying these functions to the input, resulting in

$$\mathbf{x}_o = f_K(f_{K-1}(\dots f_2(f_1(\mathbf{x}_i; \phi_1); \phi_2) \dots; \phi_{K-1}); \phi_K). \quad (1)$$

For convenience, we represent this chain as a single function $\mathbf{x}_o = f_c(\mathbf{x}_i; \phi_c)$, where $\phi_c = [\phi_1, \phi_2, \dots, \phi_K]$ concatenates all effect parameters into one vector. While our method supports arbitrarily complex effect chains, we consider only series connections.

In our setup, we perform style transfer through an optimization process via the maximization of a similarity between the output of our composite audio effect function and the reference signal given by

$$\max_{\phi_c} \text{sim}(g(f_c(\mathbf{x}_i; \phi_c)), g(\mathbf{x}_r)), \quad (2)$$

where $g(\cdot)$ denotes our audio representation, transforming audio signals into a feature space where audio production similarity is assessed. For the reference signal \mathbf{x}_r , the feature representation is $\mathbf{z}_r = g(\mathbf{x}_r)$. The optimization process initiates with a predefined set of control parameters, ϕ_0 , and iteratively refines this estimate to enhance the similarity measure. At each step, candidate solutions are generated and evaluated based on their performance in mirroring the reference features, \mathbf{z}_r . This performance is quantified by the cosine similarity measure,

$$\text{sim}(\mathbf{z}_i, \mathbf{z}_r) = \frac{\mathbf{z}_i \cdot \mathbf{z}_r}{\max(\|\mathbf{z}_i\| \|\mathbf{z}_r\|, \epsilon)}, \quad (3)$$

where $\mathbf{z}_i = g(f_c(\mathbf{x}_i; \phi_c))$ is the feature vector of the processed input signal, \cdot represents the dot product, $\|\cdot\|$ denotes the Euclidean norm, and ϵ is a small constant ensuring numerical stability, avoiding division by zero.

3. EXPERIMENTAL DETAILS

3.1 Pretraining

We employ the PANNs architecture [28] as a convolutional backbone. While initial testing indicated that more modern architectures such as HTS-AT [29] performed comparably, we found that PANNs was more efficient at inference. To enable the encoder to capture stereo information we produce separate embeddings for the mid and side signals, concatenating them into a single embedding, applying L2 normalization to each embedding before concatenation.

We train the encoder following the pretext task described in Section 2.1. We use seven publicly available audio datasets to cover a diverse range of audio content across music, speech, singing voice, and instruments. These datasets include MTG-Jamendo [30], ENST-Drums [31], URSing [32], FSD50k [33], Librispeech [34], Medley-solos-db [35], and GuitarSet [36]. To construct our set of audio effects we use 63 open source or freely available VST3 audio plugins compiled for Linux. These VSTs cover a wide range of effects including reverberation, dynamic range processing, equalization, distortion, modulation effects, and more. We use `pedalboard` [37] to load plugins and apply effects to audio signals.

We generate unique presets for each effect by randomly sampling 1000 parameter configurations and processing a random audio recording with each configuration. Then we extract MFCCs and perform K-means clustering ($K = 10$), with each cluster representing perceptually diverse parameter configurations. We then randomly select one configuration from each cluster to act as a preset.

While training with on-the-fly data generation is possible, we found running VSTs during training caused a significant bottleneck. We opted for offline data generation, where we generated 20,000 examples of length 524288 samples (≈ 11 sec at $f_s = 48$ kHz) from each dataset with randomly applied effects and presets. This corresponds to approximately 60 hours of audio content. We further increase diversity during training by taking different random crops of the pre-processed input and output segments, as well as applying random gain adjustments $[-32$ dB, 0 dB].

We perform pretraining for 1M steps with a batch size of 32 using the Adam optimizer. We use an initial learning rate of 1e-4, lowering the learning rate by a factor of 10 at 85% and 95% through training. We preprocess spectrogram inputs to the encoder by computing log-melspectrograms with window size of 2048 and hop size of 512. We clip the magnitudes between -80 and 40 dB and scale the final spectrogram between -1 and 1.

3.2 Inference-time optimization

To enable control of arbitrary effects, we employ a gradient-free optimizer as opposed to commonly used gradient-based solutions. While any gradient-free optimizer can be used in our system, we opt to use Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [38] since it has been shown to work well in spaces with similar dimensionality to the audio effect chain control parameter space (≈ 100) and is a relatively scalable method. After initial hyperparameter tuning, we use a population size of 64 and a maximum of 25 optimization steps. The σ hyperparameter is initialized to 0.3 and we use a fixed initialization of all parameters of 0.5 scaled in the range [0, 1]. We use early stopping, halting optimization after 10 steps of improvement less than 0.1.

Unless otherwise specified, we use AFx-Rep as the encoder in our similarity metric, and we consider control of two different audio effect chains. The first features five VST audio effects including distortion (TubeScreamer), parametric equalizer (ZamEQ2), dynamic range compressor (ZamCompX2), feedback delay (ZamDelay), and artificial reverberation (TAL-Reverb-4), resulting in a total of 73 parameters. The second chain features unseen audio effects internal to `pedalboard`, including distortion, dynamic range compression, parametric equalizer, delay, and artificial reverberation, resulting in 36 parameters.

4. BENCHMARK

4.1 Audio production style metrics

Zero-shot style classification. We adapt the style classification task from [14], which contains five different audio production styles using equalization and dynamic range compression: telephone (TL), bright (BR), warm (WM), broadcast (BC), neutral (NT). Training examples are generated by applying these style presets to speech from DAPS [39] and music from MUSDB18 [40]. To make the task more challenging and similar to the inference-time optimization use-case, we adapt the original task to the zero-shot case [41]. To do so, a query is constructed by sampling a random audio example to be classified as one of the five styles. Then other examples from each of the five styles are sampled randomly to form prototype classes. A representation of the query and each of the five prototypes is generated and a prediction is made by measuring the cosine similarity between the query and each of the prototypes. The class of the prototype with the highest similarity to the query forms the prediction.

Representation	Styles					AVG
	TL	BR	WM	BC	NT	
MFCCs	1.00	0.82	0.64	0.74	0.48	0.74
MIR Feats.	0.76	0.64	0.61	0.58	0.32	0.58
CLAP	0.72	0.60	0.51	0.57	0.41	0.56
Wav2Vec2	0.40	0.33	0.28	0.35	0.34	0.34
Wav2Clip	0.76	0.48	0.60	0.49	0.51	0.57
VGGish	0.47	0.58	0.43	0.61	0.43	0.50
BEATs	0.94	0.51	0.57	0.50	0.45	0.59
FX Encoder	0.96	0.94	0.29	0.70	0.54	0.69
DeepAFx-ST	1.00	0.93	0.67	0.78	0.42	0.76
DeepAFx-ST+	1.00	0.97	0.71	0.79	0.41	0.78
AFx-Rep (ours)	1.00	1.00	0.88	0.85	0.59	0.86

Table 1. Zero-shot style classification accuracy over 200 trials for music and speech across five unique styles.

Style retrieval. While the zero-shot style classification task evaluates the ability of a representation to differentiate among different styles, it considers only two basic effects and focuses on comparing significant differences. In order to more effectively evaluate the behavior of representations in a scenario similar to style transfer we designed a style retrieval task. In this task, a query style is produced by applying N effects with random parameters to an audio recording. A retrieval set is generated by processing M other recordings with differing random effect chains. One recording with differing content but the same effect chain as the query is included in the retrieval set. Similar to the zero-shot task, we measure the cosine similarity between the query and each of the items in the retrieval set. We can make the task more or less difficult by varying both the number of effects N in each style and the size of the retrieval set M . We source unseen audio examples for speech (DAPS [39]), guitar (IDMT-SMT-Guitar [42]), vocals (VocalSet [43]), and drums (IDMT-SMT-Drums [44]).

Baselines. We consider signal processing approaches, such as MFCCs and MIR features [45], as well as pretrained general purpose audio representations including VGGish [46], WAV2CLIP [47], wav2vec2.0 [48], CLAP [22], as well as BEATs [23]. We also compare against audio effect specific models including FX-Encoder [13] and the DeepAFx-ST encoder [14].

4.2 Audio production style transfer

Parameter estimation. To demonstrate the ability of our proposed approach to control a wide range of effects we design a parameter estimation task. We initialize an audio effect and set a target value for one parameter, processing a random audio signal to generate a reference. Then we sample another random recording to use as the input. We then run the optimization using each audio representation in our metric. To achieve accurate style transfer a system should estimate a control parameter with a similar, but not necessarily identical value as the reference. We report both the mean squared error (MSE) and the correlation coefficient ρ of estimated parameters. We consider six VST effects as well as six unseen effects from `pedalboard`.

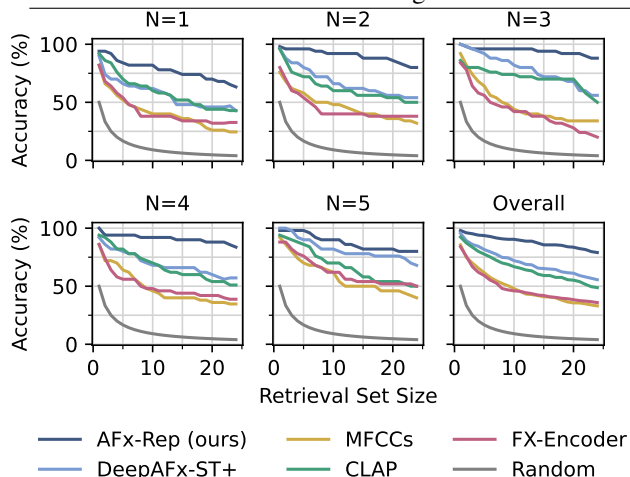


Figure 4. Accuracy for the style retrieval task using different audio representations across multiple source types with varying number of audio effects (N) and retrieval set size.

Real-world style transfer. While the parameter estimation task can demonstrate the ability of a style transfer system to control a singular audio effect, it does not capture the ability of the system to perform in a real-world scenario. In many cases multiple effects will be present at the same time, making the task more challenging. To evaluate this scenario we created six different audio production styles by constructing realistic audio effect chains of varying complexity in a digital audio workstation. These styles range from simple lowpass and highpass filtering to a complete channel strip featuring equalization, distortion, compression, delay, and reverberation. We then applied these styles to a range of audio content including speech, singing voice, and full music tracks. Each style transfer system is then tasked with transforming the unprocessed audio from one of these content types to the stylized version of a different recording containing the same kind of content.

Baselines. We compare our proposed style transfer system to both deep learning and signal processing solutions. We use a rule-based approach from previous work that includes an FIR matching equalization filter and a simple hill climbing-based dynamic range compressor [14]. We construct a strong deep learning baseline by extending DeepAFx-ST with differentiable reverberation [49] and distortion [50] effects, using `dasp-pytorch`². We call this approach DeepAFx-ST+ and we train this model following the approach from the original work, but using the same datasets used to train our audio representation.

5. RESULTS

Zero-shot style classification. We evaluate the pretrained representations across ten trials for each of the five different styles. The class-wise and overall accuracy is reported in Table 1. First, we find MFCC based features perform better than expected, with high accuracy on the telephone (TL), bright (BR), and warm (WM) styles. However, performance is worse on broadcast (BC) and neu-

²<https://github.com/csteinmetz1/dasp-pytorch>

Effect (Parameter)	MSE (\downarrow)		ρ (\uparrow)	
	CLAP	AFx-Rep	CLAP	AFx-Rep
RoughRider (sensit)	0.183	0.084	0.300	0.705
DPlate (decay)	0.141	0.025	0.610	0.945
3BandEQ (high_)	0.033	0.026	0.876	0.919
MaGigaverb (size)	0.018	0.012	0.949	0.969
MetalTone (dist)	0.155	0.040	0.509	0.862
TAL-Chorus (wet)	0.097	0.014	0.654	0.953
*Chorus (mix)	0.164	0.172	0.300	0.408
*Reverb (room_)	0.048	0.013	0.822	0.955
*Delay (mix)	0.117	0.052	0.591	0.815
*Distortion (drive)	0.023	0.005	0.852	0.944
*Compressor (thresh)	0.134	0.096	0.518	0.678
*ParametricEQ (low_s)	0.110	0.031	0.727	0.931

Table 2. Parameter estimation with ST-ITO using CLAP and our proposed AFx-Rep. We report the mean squared error (MSE) and correlation coefficient (ρ) of the estimated parameters across 4 different settings and 3 trials per effect. Audio effects not seen during pretraining are denoted by *.

tral (NT), likely because identifying these styles requires paying attention to dynamics. The MIR features do not achieve comparable performance. All of the general purpose audio representations perform worse than MFCCs on this task, with CLAP and BEATs appearing to perform best among them, but with an average accuracy 15 points lower. This confirms the hypothesis that general purpose representations fail to capture information about audio effects. FX-Encoder and DeepAFx-ST(+) perform better than the other pretrained models, with DeepAFx-ST variants outperforming MFCCs. Overall, we find that our proposed model, AFx-Rep, performs best in this task.

Style retrieval. We report the accuracy for a subset of methods in style retrieval as shown in Figure 4. We plot performance across differing number of effects N that constitute a style, as well as the size of the retrieval set, shown on the x-axis. As expected, for all scenarios, as the retrieval set grows the classification performance drops. While all methods are better than random guessing, we observe that MFCCs and FX-Encoder appear to perform worse. They are followed by CLAP and then the encoder from DeepAFx-ST+, which slightly outperforms CLAP. Finally, our proposed AFx-Rep model performs best across all scenarios, indicating its superior ability to capture elements related to audio production style.

Parameter estimation. In Table 2 we report the mean squared error (MSE) and correlation coefficient (ρ) in parameter estimation using ST-ITO with either CLAP or our proposed AFx-Rep model. In nearly all cases our AFx-Rep model functions as a superior similarity metric, achieving lower MSE and a higher correlation coefficient, with the exception of the MSE in Chorus, which appears to be challenging for both models. This demonstrates the ability of our approach to control a wide range of real-world audio effects, including effects not seen during retraining. These results reinforce the importance of an audio representation sensitive to audio effects, such as our proposed AFx-Rep.

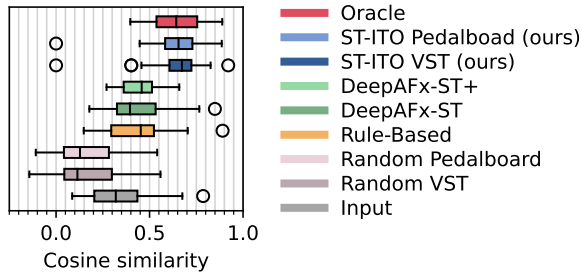


Figure 5. AFX-Rep similarity in real-world style transfer.

Real world style transfer. We report the similarity from our metric using AFX-Rep across 56 style transfer trials (Figure 5). The Input processed with an audio effect chain identical to the reference is also evaluated, which we refer to as Oracle. Note that the Oracle may not achieve effective style transfer as the starting point of the Input may require a different parameter configuration to match the reference. The random configuration of VSTs and pedalboard effect perform worse, and are followed by the Input, which features no processing. DeepAFx-ST, DeepAFx-ST+, and the Rule-Based system appear to perform similarly to each other, but better than Input. Variants of ST-ITO, one using VSTs and the other using unseen pedalboard effects, both outperform the rest, and are on par with the Oracle. This indicates the ability of our approach to optimize our metric, however, it is difficult to make conclusions about style transfer performance using this evaluation alone.

Subjective listening study. We recruited 23 participants with experience in audio engineering. They were tasked with evaluating style transfer systems on real-world test scenarios in a multiple stimulus listening study. Listeners were asked to provide a score from 0 to 100 for each stimulus to indicate its similarity to the reference, considering only the style and not the underlying content. In addition, we also included the unprocessed Input and the Oracle. Due to the subjectivity of this task, evaluators may not rate Input the lowest and Oracle the highest. We selected ten test cases across the real-world scenarios including vocals (V), music (M), and speech (S) as shown in Figure 6.

Overall, listeners found the Oracle most similar to the reference and the Input the least similar, as expected. However, there is variation in the score assigned to both, indicating some disagreement. For simple styles, such as V1 (lowpass) and M2 (highpass), we found the Rule-Based system worked well, even surpassing style transfer systems. However, in cases with multiple effects, the Rule-Based system does not work well, as in V2 (large space), V3 (small space), V4 (delay), S1 (small space), and S3 (distortion). Differences between ST-ITO and DeepAFx-ST+ are harder to discern. Our method outperforms in some cases, such as V2 (large space), V3 (small space), and V4 (delay), yet in other cases there is no clear difference. We conclude that our approach is capable of style transfer at least on par with the enhanced DeepAFx-ST+, and does so controlling a chain of unseen VST audio effects, which is not possible with other approaches.

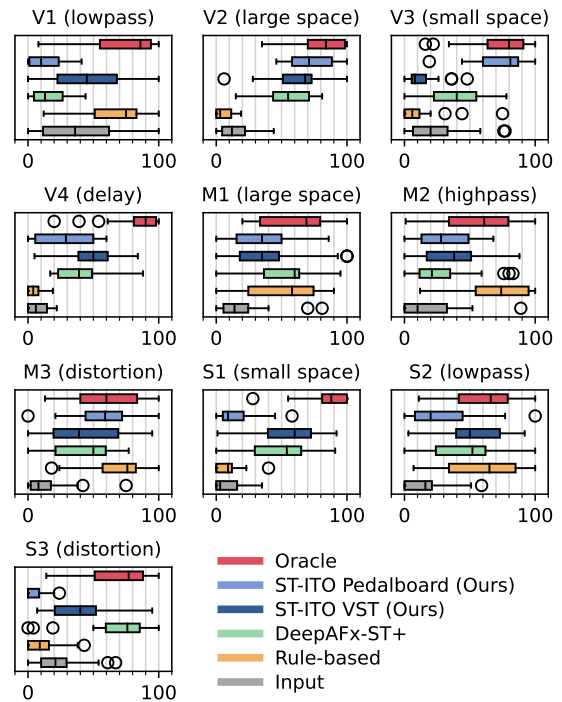


Figure 6. Subjective scores from $N = 23$ participants across vocals (V), music (M), and speech (S) examples.

6. DISCUSSION

While ST-ITO enables control of arbitrary audio effects and adapts to new effects at inference, it has some limitations. In the current formulation, our system requires an appropriate audio effect chain be provided. Future work could consider automatically constructing this audio processing graph as in blind estimation [51]. Furthermore, while our method does not require training “in-the-loop” with audio effects during representation learning, we must process many variants of the recording through the audio effect chain during style transfer. This leads to significantly longer inference times (≈ 1 min) as compared to networks that estimate parameters directly (≈ 1 sec). Future work could consider the design of more efficient optimizers through meta-learning by training an optimizer for a particular effect chain [52]. Finally, we have found that the current system does not work well for challenging style transfer applications, such as guitar tone matching.

7. CONCLUSION

In this work, we introduced ST-ITO, Style Transfer with Inference-Time Optimization. Unlike previous style transfer systems, ST-ITO searches the parameter space of any audio effect chain at inference, enabling control of arbitrary effect chains, including those with non-differentiable effects. Our methodology leverages a self-supervised audio production style metric and a gradient-free optimizer. We developed a set of benchmarks to evaluate both audio production style representations and style transfer systems. Results from this set of benchmarks indicate that our approach not only better captures details related to audio production style, but also provides enhanced flexibility and expressiveness in audio production style transfer.

8. ACKNOWLEDGMENTS

Supported by EPSRC UKRI CDT in AI+Music (Grant no. EP/S022694/1). EB is supported by RAEng/Leverhulme Trust research fellowship LTRF2223-19-106.

9. REFERENCES

- [1] T. Wilmering, D. Moffat, A. Milo, and M. B. Sandler, "A history of audio effects," *Applied Sciences*, vol. 10, no. 3, p. 791, 2020.
- [2] B. De Man, R. Stables, and J. D. Reiss, *Intelligent Music Production*. Focal Press, 2019.
- [3] B. De Man, J. Reiss, and R. Stables, "Ten years of automatic mixing," in *Workshop on Intelligent Music Production, Salford*, 2017.
- [4] J. Scott, M. Prockup, E. M. Schmidt, and Y. E. Kim, "Automatic multi-track mixing using linear dynamical systems," in *Proceedings of the 8th Sound and Music Computing Conf., Padova, Italy*, 2011.
- [5] D. Moffat and M. Sandler, "Machine learning multi-track gain mixing of drums," in *147th Convention of the Audio Engineering Society Convention*, 2019.
- [6] M. Martínez Ramírez, D. Stoller, and D. Moffat, "A deep learning approach to intelligent drum mixing with the Wave-U-Net," *Journal of the Audio Engineering Society*, 2021.
- [7] C. J. Steinmetz, J. Pons, S. Pascual, and J. Serrà, "Automatic multitrack mixing with a differentiable mixing console of neural audio effects," in *IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2021.
- [8] M. N. Lefford, G. Bromham, G. Fazekas, and D. Moffat, "Context aware intelligent mixing systems," *Journal of the Audio Engineering Society*, 2021.
- [9] D. Sheng and G. Fazekas, "A feature learning siamese model for intelligent control of the dynamic range compressor," in *IEEE Intl. Joint Conf. on Neural Networks (IJCNN)*, 2019.
- [10] S. I. Mimilakis, N. J. Bryan, and P. Smaragdis, "One-shot parametric audio production style transfer with application to frequency equalization," in *IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.
- [11] J. Koo, S. Paik, and K. Lee, "Reverb conversion of mixed vocal tracks using an end-to-end convolutional deep neural network," in *IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2021.
- [12] —, "End-to-end music remastering system using self-supervised and adversarial training," in *IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.
- [13] J. Koo *et al.*, "Music mixing style transfer: A contrastive learning approach to disentangle audio effects," in *IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2023.
- [14] C. J. Steinmetz, N. J. Bryan, and J. D. Reiss, "Style transfer of audio effects with differentiable signal processing," *J. Audio Eng. Soc.*, vol. 70, no. 9, 2022.
- [15] C. Peladeau and G. Peeters, "Blind estimation of audio effects using an auto-encoder approach and differentiable digital signal processing," in *IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2024.
- [16] Y. Wang *et al.*, "Audit: Audio editing by following instructions with latent diffusion models," *Advances in Neural Information Processing Systems*, vol. 36, 2023.
- [17] B. Han *et al.*, "InstructME: An instruction guided music edit and remix framework with latent diffusion models," *arXiv preprint arXiv:2308.14360*, 2023.
- [18] J. Engel, L. Hantrakul, C. Gu, and A. Roberts, "DDSP: Differentiable digital signal processing," in *ICLR*, 2020.
- [19] M. A. M. Ramírez, O. Wang, P. Smaragdis, and N. J. Bryan, "Differentiable signal processing with black-box audio effects," in *IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2021.
- [20] S. S. Vanka, M. Safi, J.-B. Rolland, and G. Fazekas, "The role of communication and reference songs in the mixing process: Insights from professional mix engineers," *Journal of the Audio Engineering Society*, vol. 72, no. 1/2, 2024.
- [21] J. Turian, J. Shier *et al.*, "Hear: Holistic evaluation of audio representations," in *NeurIPS 2021 Competitions and Demonstrations Track*. PMLR, 2022.
- [22] Y. Wu, K. Chen, T. Zhang, Y. Hui, T. Berg-Kirkpatrick, and S. Dubnov, "Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation," in *IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2023.
- [23] S. Chen *et al.*, "BEATs: Audio pre-training with acoustic tokenizers," in *ICML*, vol. 202. PMLR, 2023.
- [24] S. H. Hawley and C. J. Steinmetz, "Leveraging neural representations for audio manipulation," in *154th Convention of the Audio Engineering Society*, 2023.
- [25] M. A. Martínez-Ramírez *et al.*, "Automatic music mixing with deep learning and out-of-domain data," in *Intl. Society for Music Information Retrieval Conf. (ISMIR)*, December 2022.
- [26] J. Imort, G. Fabbro, M. A. M. Ramírez, S. Uhlich, Y. Koyama, and Y. Mitsufuji, "Distortion audio effects: Learning how to recover the clean signal," in *Intl. Society for Music Information Retrieval Conf. (ISMIR)*, 2022.

- [27] M. Rice, C. J. Steinmetz, G. Fazekas, and J. D. Reiss, "General purpose audio effect removal," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2023.
- [28] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley, "PANNs: Large-scale pretrained audio neural networks for audio pattern recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, 2020.
- [29] K. Chen *et al.*, "HTS-AT: A hierarchical token-semantic audio transformer for sound classification and detection," in *IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.
- [30] D. Bogdanov, M. Won, P. Tovstogan, A. Porter, and X. Serra, "The MTG-Jamendo dataset for automatic music tagging," in *ICML*, 2019.
- [31] O. Gillet and G. Richard, "ENST-Drums: an extensive audio-visual database for drum signals processing," in *Intl. Society for Music Information Retrieval Conf. (ISMIR)*, 2006.
- [32] B. Li *et al.*, "University of Rochester Audio-Visual Solo Singing Performance Dataset," 2022.
- [33] E. Fonseca, X. Favory, J. Pons, F. Font, and X. Serra, "Fsd50k: an open dataset of human-labeled sound events," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, 2021.
- [34] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *IEEE Intl. Conf. on Acoustics, Speech and Signal processing (ICASSP)*, 2015.
- [35] V. Lostanlen, C.-E. Cella, R. Bittner, and S. Essid, "Medley-solos-db: a crosscollection dataset for musical instrument recognition," *Zenodo*, 2018.
- [36] Q. Xi, R. M. Bittner, J. Pauwels, X. Ye, and J. P. Bello, "Guitarset: A dataset for guitar transcription," in *Intl. Society for Music Information Retrieval Conf. (ISMIR)*, 2018.
- [37] P. Sobot, "Pedalboard," Jul. 2021. [Online]. Available: <https://doi.org/10.5281/zenodo.7817838>
- [38] N. Hansen and A. Ostermeier, "Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation," in *IEEE Intl. Conf. on Evolutionary Computation*, 1996.
- [39] G. J. Mysore, "Can we automatically transform speech recorded on common consumer devices in real-world environments into professional production quality speech?—a dataset, insights, and challenges," *IEEE Signal Processing Letters*, vol. 22, no. 8, 2014.
- [40] Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimilakis, and R. Bittner, "MUSDB18: A corpus for music separation," 2017.
- [41] P. Wolters, C. Careaga, B. Hutchinson, and L. Phillips, "A study of few-shot audio classification," *arXiv preprint arXiv:2012.01573*, 2020.
- [42] C. Kehling, J. Abeßer, C. Dittmar, and G. Schuller, "Automatic tablature transcription of electric guitar recordings by estimation of score-and instrument-related parameters," in *DAFx*, 2014.
- [43] J. Wilkins, P. Seetharaman, A. Wahl, and B. Pardo, "Vocalset: A singing voice dataset," in *Intl. Society for Music Information Retrieval Conf. (ISMIR)*, 2018.
- [44] C. Dittmar and D. Gärtner, "Real-time transcription and separation of drum recordings based on nmf decomposition," in *DAFx*, 2014, pp. 187–194.
- [45] B. Man, B. Leonard, R. King, J. D. Reiss *et al.*, "An analysis and evaluation of audio features for multitrack music mixtures," in *Intl. Society for Music Information Retrieval Conf. (ISMIR)*, 2014.
- [46] S. Hershey *et al.*, "Cnn architectures for large-scale audio classification," in *IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.
- [47] H.-H. Wu *et al.*, "Wav2clip: Learning robust audio representations from clip," in *IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.
- [48] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in neural information processing systems*, vol. 33, 2020.
- [49] C. J. Steinmetz, V. K. Ithapu, and P. Calamia, "Filtered noise shaping for time domain room impulse response estimation from reverberant speech," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2021.
- [50] J. T. Colonel, M. Comunità, and J. Reiss, "Reverse engineering memoryless distortion effects with differentiable waveshapers," in *153rd Convention of the Audio Engineering Society*, 2022.
- [51] S. Lee, J. Park, S. Paik, and K. Lee, "Blind estimation of audio processing graph," in *IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2023.
- [52] J. Casebeer, N. J. Bryan, and P. Smaragdis, "Meta-af: Meta-learning for adaptive filters," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, 2022.

COMPOSERX: MULTI-AGENT SYMBOLIC MUSIC COMPOSITION WITH LLMs

Qixin Deng⁵ **Qikai Yang**⁶ **Ruibin Yuan**¹
Yipeng Huang² Yi Wang² Xubo Liu⁸
Zeyue Tian¹ Jiahao Pan¹ Ge Zhang⁹
Hanfeng Lin² Yizhi Li⁴ Yinghao Ma³
Jie Fu¹ Chenghua Lin⁴ Emmanouil Benetos³
Wenwu Wang⁸ Guangyu Xia⁷ Wei Xue¹
Yike Guo¹

¹ Hong Kong University of Science and Technology
² Multimodal Art Projection Research Community
³ Queen Mary University of London
⁴ The University of Manchester
⁵ University of Rochester
⁶ University of Illinois at Urbana-Champaign
⁷ Mohamed bin Zayed University of Artificial Intelligence
⁸ University of Surrey
⁹ 01.AI

ABSTRACT

Music composition represents the creative side of humanity, and itself is a complex task that requires abilities to understand and generate information with long dependency and harmony constraints. Current LLMs often struggle with this task, sometimes generating poorly written music even when equipped with modern techniques like In-Context-Learning and Chain-of-Thoughts. To further explore and enhance LLMs’ potential in music composition by leveraging their reasoning ability and the large knowledge base in music history and theory, we propose **ComposerX**¹, an agent-based symbolic music generation framework. We find that applying a multi-agent approach significantly improves the music composition quality of GPT-4. The results demonstrate that ComposerX is capable of producing coherent polyphonic music compositions with captivating melodies, while adhering to user instructions.

1. INTRODUCTION

Music shares many structural similarities with language [1–3], prompting researchers to explore the ap-

plication of language models (LMs) in music generation [4–14]. Recent advances in large language models (LLMs) have opened potential pathways towards achieving Artificial General Intelligence (AGI). While much of the research emphasis has been on the STEM aspects of AGI [15–17], there is comparatively less focus on the creative potential in generative LLMs, particularly in music creation. Current methodologies primarily involve training LMs from scratch, as seen with initiatives like MusicLM [9] and MusicGen [10], with a predominant focus on audio generation. However, these models often struggle with processing advanced musical instructions and typically offer only limited control options, such as genre and instrument selection. Enhancing controllability in these systems requires neural architectural engineering and extensive computational resources [18–20].

Recent research, influenced by Bubeck et al. [17], has revealed that pretrained large language models (LLMs) might inherently possess emergent musical capabilities. Inspired by these findings, subsequent studies [21–23] have explored leveraging pretrained LLM checkpoints for handling symbolic music in an end-to-end manner, aiming to tap into the extensive knowledge and reasoning abilities embedded in these LLMs. However, these unified approaches are not without limitations. They depend heavily on hand-crafted datasets tailored for specific musical tasks and often require both a phase of continual pretraining and subsequent supervised fine-tuning. Furthermore, while training on symbolic music data is generally less computationally intensive than processing raw audio data, the costs remain prohibitive for many researchers. For example, renting an 8xGPU machine (such as a p4d.24xlarge

¹ Demo page: <https://glossy-scowl-a33.notion.site/ComposerX-Demo-e53b59f17540401785437f3bee38c308?pvs=4>



© Q. Deng, Q. Yang, and R. Yuan. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Q. Deng, Q. Yang, and R. Yuan, “ComposerX: Multi-Agent Symbolic Music Composition with LLMs”, in *Proc. of the 25th Int. Society for Music Information Retrieval Conf.*, San Francisco, United States, 2024.

In this paper, we introduce a novel multi-agent-based methodology, ComposerX³, which is training-free, cheap, and unified. Leveraging the internal musical capabilities of the state-of-the-art GPT-4-turbo, ComposerX can generate polyphonic music pieces of comparable, if not superior, quality to those produced by dedicated symbolic music generation systems [7, 24] that require extensive computational resources and data. ComposerX utilizes approximately 26k tokens per song, incurring a cost of less than \$0.8 USD per piece. Throughout the development phase of ComposerX, the total expenditure on the OpenAI API was under \$1k USD. We achieved a good case rate of 18.4%, as assessed by music experts, which translates to an average cost of approximately \$4.34 USD for each musically interesting piece. Furthermore, experimental results demonstrate that the multi-agent strategy substantially enhances composition quality over single-agent baselines. In Turing tests, approximately 32.2% of the pieces identified as ‘good’ by ComposerX were indistinguishable from those composed by humans, as indicated in Table 3.

While there is existing research on musical LLM agents [25, 26], our approach distinctively diverges from these precedents. Prior studies primarily focus on single-agent systems. In contrast, our work introduces a multi-agent framework, emphasizing collaborative aspects of music creation. Furthermore, we concentrate on symbolic music generation, leveraging the intrinsic musical understanding of LLMs without the need for external computational resources or tools. Previous methodologies typically depend on GPU servers for deploying local inference services, treating the LLMs more as tool-use agents rather than harnessing their inherent capabilities to process and generate musical content. In sum, the contributions of our paper are as follows:

- (1) We propose the first LLM-based multi-agent polyphonic symbolic music composition system, ComposerX. It elicits the internal musical capabilities inside LLMs without the need for external tools.
- (2) Through extensive subjective evaluations, we demonstrate that our multi-agent approach substantially enhances the quality of music composition compared to single-agent systems and specialized music generation models. Our method also offers cost-efficiency advantages by obviating the need for dedicated training or local inference services.
- (3) We commit to the advancement of this research area by open-sourcing our code, prompt-set, and experimental results, facilitating further investigation and development by the community.

2. METHOD

We first construct a set of user prompts for music composition, which is used for evaluation. Then we demonstrate

² <https://instances.vantage.sh/aws/ec2/p4d.24xlarge>

³ <https://github.com/llindsey0615/ComposerX>

2.1 User Prompt Set Curation

To understand how the users, typically those with substantial musical backgrounds, would prompt a text-to-music generation system, a user prompt set is collected by asking humans with music backgrounds to manually write high-quality prompts. These prompts typically include essential musical attributes such as genre, tempo, key, chord progression, melody, rhythm, number of bars, number of voices, instruments, style, feeling, emotion, title, and motif of the music piece. Based on the human-written samples, more prompt samples are generated using Self-instruct by GPT-4 [27]. This results in a set of 163 prompts, which is used in the later agent testing and system evaluation. An example prompt is given below.

Prompt

Vintage French Chanson: A nostalgic chanson in C major with a slow tempo, featuring accordion, violin, and upright bass over 16 bars with chords C, Am, Dm, G. The accordion leads with expressive sound, violin adds romance, and the upright bass supports, evoking vintage French charm.

Attributes

Name: Vintage French Chanson **Tempo:** Slow
Feeling: Nostalgic **Chord Progression:** C, Am, Dm, G
Key: C major **Bars:** 16 **Instruments:** Accordion, violin, upright bass

2.2 Single-Agent

We apply various prompt engineering techniques, including In Context Learning (ICL), Chain of Thought (CoT), and Role-play to guide a single GPT acting as the composer. Additionally, we have refined the prompt template by incorporating specific instructions that ensure the correctness of the ABC notation format.

Original GPT with Simple Role-play (Ori): To investigate the inherent capabilities of the original GPT model in interpreting user prompts and generating ABC notation, we instructed GPT to act in the role of a professional composer, with user prompts directly input into the system. This method aims to assess the model’s basic performance in music composition without the integration of additional complex prompting techniques.

Role-Play with Additional Instruction (Role): Inspired by classical rule-based computer music generation, we equipped GPT with enhanced musical knowledge focusing on phrase management and melody line construction, detailed in A.1. For example, in composing melodies, we instructed the model to ensure distinct phrase divisions, with each phrase ending on a prominent note. These instructions aim to improve the quality and structural coherence of the music, aligning the generated compositions more closely with traditional musical standards.

Chain-of-Thought (CoT): As proven in other fields of research, CoT improves the ability of LLMs on complex reasoning by encouraging them to write down inter-

In Context Learning (ICL): ICL leverages a few input-output examples to enhance an LLM’s understanding of a specific task. In this method, we use pairs of user prompts and corresponding ABC notations from ChatMusician [21] as demonstrative examples.

2.3 Multi-Agent Music Composition: ComposerX

To enhance the music generation capabilities of GPT-4, we developed a collaborative music creation framework, ComposerX, that draws inspiration from key elements inherent in real-world music composition processes, such as melody construction, harmony or counterpoint development, and instrumentation. This framework facilitates the music creation process through a structured conversation chain between agents role-played by GPT-4.

2.3.1 Agent Role Assignment

In the collaborative music creation framework designed to augment GPT-4’s music generation capabilities, roles are assigned to ensure a structured and efficient composition process. The assignment of roles is as follows:

Group Leader: Tasked with interpreting user inputs, decomposing these inputs into granular tasks, and assigning these tasks to specialized agents in the group.

Melody Agent: Responsible for generating single-line melodies under the guidance of the group leader.

Harmony Agent: This agent is tasked with enriching the musical piece, and adds harmonic and contrapuntal elements to the melody.

Instrument Agent: This agent selects and assigns instruments to each voice.

Reviewer Agent: Performing a quality assurance role, this agent evaluates the outputs of the melody, harmony, and instrumentation agents across four critical dimensions.

(1) **Melodic Structure:** Evaluation of melody’s narrative flow, thematic development, and variation in pitch and rhythm. (2) **Harmony and Counterpoint:** Assessment of how harmonies complement the melody, counterpoint effectiveness, and chord progression quality. (3) **Rhythmic Complexity:** Analysis of rhythm’s role in sustaining interest, its synergy with melody, and the incorporation of dynamic variations. (4) **Instrumentation and Timbre:** Review of instrument selection, timbral blending, and dynamic usage to achieve an optimal auditory experience. (5) **Form and Structure:** Examination of the composition’s overarching structure, transitional elements, connectivity between sections, and conclusion efficacy.

Arrangement Agent: Concluding the collaborative process, this agent is responsible for compiling and formatting the collective output into standardized ABC notation, ensuring the music is documented in a universally readable format.

2.3.2 Agent Communication Pattern

The collaborative framework uses a structured communication pattern to ensure an orderly and efficient flow of information between agents in the composition process. This pattern is crucial for maintaining the integrity and coherence of the musical piece. The communication process unfolds as follows:

Initial Composition Round: The composition process begins with the Group Leader Agent initiating the sequence by analyzing the user input and breaking it down into specific tasks assigned to the Melody, Harmony, and Instrument Agents respectively. This step sets the foundation for the composition based on the user’s requirements. Following the leader’s instructions, the Melody Agent then generates the initial melody line, adhering to the thematic direction and stylistic guidelines provided by the Group Leader. Subsequently, the Harmony Agent enriches the melody by adding harmonic layers and counterpoints. The Instrument Agent assigns appropriate instruments to the generated melody and harmony lines by selecting timbres that complement the overall composition.

Iterative Review and Feedback Cycle: Upon completion of the initial composition round, the Reviewer Agent steps in to evaluate the work produced by the Melody, Harmony, and Instrument Agents. This agent provides comprehensive feedback across several critical dimensions, including melodic structure, harmony and counterpoint, rhythmic complexity, and instrumentation.

Based on the feedback from the Reviewer Agent, the Melody, Harmony, and Instrument Agents proceed to refine their respective parts of the composition. This refinement process typically follows the order: Melody, Harmony, and then Instrument, allowing for modifications to be made in response to the feedback provided.

The composition undergoes several rounds of review and refinement, with the Reviewer Agent continuously providing feedback to ensure the musical piece evolves toward a coherent and high-quality final product. This iterative process allows for dynamic adjustments and enhancements to be made, enriching the overall composition.

Final Arrangement and Notation: Once the composition has reached a satisfactory level of polish and coherence, the Arrangement Agent takes over to compile and format the collective output into the standardized ABC notation. This final step ensures that the music is documented in a format that is readable and can be interpreted by musicians and software alike.

2.3.3 Agent Prompt Engineering

Agent prompt engineering emerges as a crucial technique for optimizing the performance of each specialized agent and the quality of the generated music. This process involves the meticulous design of role-specific instructions and guidelines that encapsulate both the musicality and technicality of ABC notation generation. The framework incorporates In-Context Learning for ABC notations to ensure agents can effectively communicate and document their contributions. This section elaborates on these components and their significance in fostering collaborative

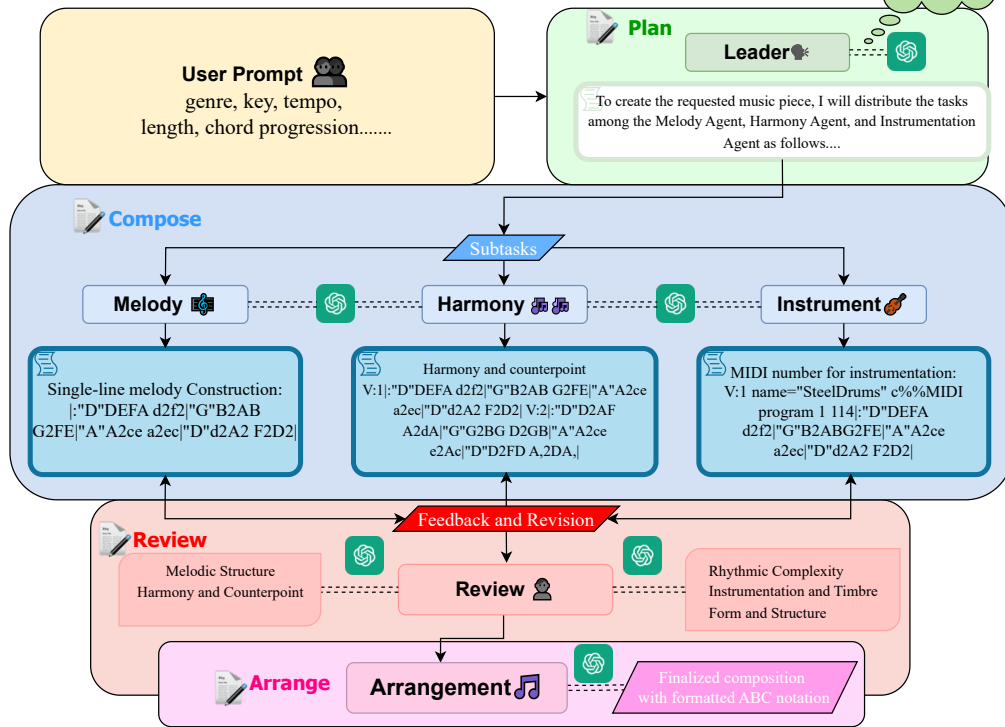
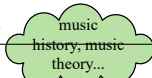


Figure 1. Agent Communication Pattern of ComposerX. The system is given with a user prompt. In the Planning stage, the Leader analyzes the user prompt and decomposes it into subtasks that can be assigned to other musician agents. In the Composing stage, the musician agents, including Melody Agent, Harmony Agent, and Instrument Agent compose in ABC notation according to their assigned tasks. In the Reviewing stage, the Review Agent provides constructive feedback to the musician agents and the musician agents revise their work according to the feedback they received. In the arrangement stage, the Arrangement Agent arranges the work of the musicians agent to standardized ABC notation.

dynamics within the framework.

Role-Specific Instructions: Within the framework, each agent is endowed with a set of instructions tailored to its designated role. These instructions serve to ensure a comprehensive understanding of the agent’s duties, the expectations for its performance, and its role within the larger collaborative ensemble. Agents are briefed on the specific outcomes they are expected to achieve and informed about the dynamics of their interactions with other agents. This detailed prompt design facilitates a cohesive operation among the agents, fostering an environment where each component of the framework is aligned toward the collective goal of generating sophisticated and coherent musical compositions.

In-Context Learning for ABC Notation: In Context Learning for ABC notation ensures accurate format output from each agent. The Melody Agent is shown with an example of a monophonic melody in ABC notation, providing a clear model for representing single-line melodies. The Harmony Agent receives a polyphonic music piece example in ABC notation, aiding in understanding the notation of harmonies and counterpoints in multiple voices. The Instrument Agent is given a polyphonic piece with MIDI program of the instrumental information noted, demonstrating how to detail instrumental assignments within the notation. This approach equips agents

with the knowledge to correctly apply ABC notation, essential for the structured and coherent documentation of musical compositions.

3. EXPERIMENTS

3.1 Setup

Our experiment leverages the multi-agent conversation provided by the AutoGen framework [29], utilizing its group chat function to facilitate a customized interaction among pre-defined agents. This setup comprises an ensemble of agents including one leader, three musician agents (melody, harmony, and instrument agents), one review agent, and one arrangement agent. Additionally, a user proxy agent is integrated into the framework to simulate user interaction by inputting prompts from our curated user prompt set.

We use the "GroupChatManager" class from AutoGen to coordinate and oversee the conversation’s content and workflow. The group manager, powered by LLMs, supervises the conversation and implements a structured communication protocol with three steps: dynamically selecting a speaker, collecting the response, and disseminating it to the group.

For our experiment, we limit the agent communication

to twelve rounds, allowing us to observe the system’s effectiveness over defined interaction cycles and enabling iterative review and refinement. This structured design aims to evaluate the collaborative dynamics and output quality of the multi-agent conversation in generating cohesive and musically rich compositions based on user prompts.

3.2 Evaluation

3.2.1 Quantitative Evaluation

We conducted two experiments to evaluate our system quantitatively. One experiment assessed the success rate of generating symbolic music in a multi-agent setting, with results presented in Table 1. One experiment compared the sequence lengths of symbolic music generated by multi-agent and single-agent systems, detailed in Table 2. These experiments demonstrate the effectiveness of our approach in generating symbolic music.

Checkpoints	Generation Success Rate
GPT-4-Turbo	98.2%
GPT-4-0314	95.7%
GPT-3.5-Turbo	73.0%

Table 1. One-time generation success rate for multi-agent system with different checkpoints

Methods	Average ABC String Length
GPT-4-Turbo multi	1005.925
GPT-4-Turbo cot	360.92
GPT-4-Turbo icl	366.30
GPT-4-Turbo ori	354.53
GPT-4-Turbo role	337.64

Table 2. The average length of ABC String generated by different methods on GPT-4-Turbo checkpoint

3.2.2 Human Listening Test

To qualitatively assess our work, we conducted three listening tests. The selected listeners are mostly undergraduate and postgraduate students who have educational backgrounds in either STEM or music, or both. In the first test, we compared music samples generated by single-agent and multi-agent baselines. Similar to the AB-test setting from previous work [21,30], participants were presented with 50 pairs of samples randomly chosen from a pool of 200 sample pairs: one from a multi-agent baseline with GPT-4 Turbo checkpoints, and the other from a single-agent baseline employing prompting techniques mentioned above: Original(Ori), In-Context Learning (ICL), Chain of Thought (CoT), and Role-play(Role), also driven by GPT-4 Turbo checkpoints. Participants were asked to select the sample they preferred. All paired samples were generated using the same prompt; however, participants were not informed about the specific prompt details before making their selections.

In the second listening test, we assess the perceived human-like quality of music generated by the multi-agent baselines. Participants were presented with 30 pairs of

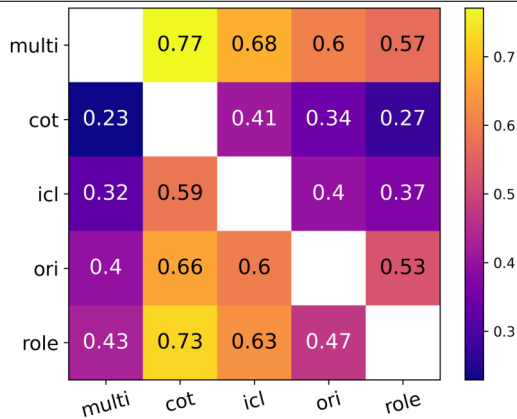


Figure 2. Result from the first listening test comparing multi-agent baseline and single-agent baselines with different prompting techniques. Each row indicates the fraction of listeners’ preference for the indicated baseline over other baselines. i.e. 0.77 means raters prefer multi-agent system over CoT single-agent 77% of the times.

music samples: those generated by multi-agent baselines and those composed by humans, sourced from Irishman and KernScores⁴, which are ABC notation datasets containing human-composed music pieces from all around the world. Each participant is asked to determine whether each sample was composed by a human or a machine.

In the third listening test, we assessed the performance of our multi-agent baselines, which incorporate GPT-4 Turbo, GPT-4-0314, and GPT-3.5-Turbo checkpoints, against established text-to-music generation models. Specifically, comparisons were made with MuseCoco [7], developed by Peiling Lu et al., and a BART-based model fine-tuned on 282,870 English text2music pairs in ABC notation, as proposed by Wu et al [24]. Participants were presented with music samples generated from these five baselines, alongside their corresponding prompts, and asked to select the sample that best matched the prompt in terms of musical structure and content. This test involved 30 prompts and their generated music samples, randomly selected from a pool of 200 user prompts.

3.3 Results

Results from comparing multi-agent baseline and single-agent baseline appear in Figure 2. The preference score of GPT-4-Turbo multi has 0.77, 0.68, 0.6, and 0.57 on each of other single-agent baselines.

Model	Perceived as Human	Perceived as Machine
ComposerX	32.2%	67.8%
Ground Truth	55.4%	44.6%

Table 3. Result from our second listening test (Turing test).

Results from comparing the multi-agent baseline with

⁴ <http://kern.ccarh.org/>

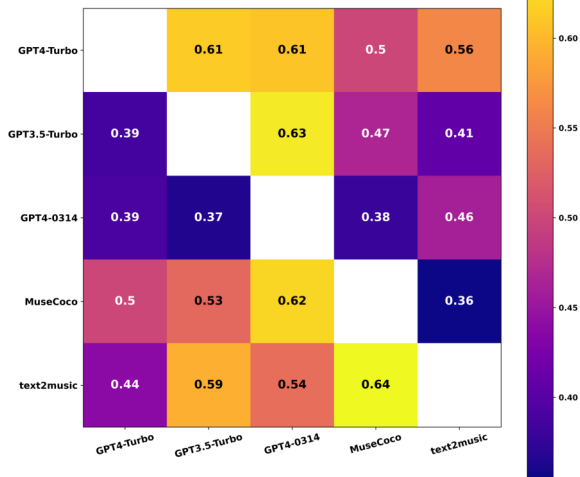


Figure 3. Result from listening test comparing multi-agent baselines with GPT-4-Turbo, GPT-4-0314, GPT-3.5-Turbo checkpoints, MuseCoco and text2music Baselines. Each row indicates the fraction of listeners’ preference for the indicated baseline over other baselines. In this case, the strongest multi-agent baseline with GPT-4-Turbo checkpoints outperformed text2music, and received the same score as MuseCoco.

music composed by humans indicate that ComposerX gets 32.2% perceived as human which is lower than the rate of real human music - 55.4% as indicated in Table 3. Despite failing the Turing test, ComposerX showcases its capability to closely match human music composition skills.

Results from comparing the multi-agent baseline with GPT-4-Turbo, GPT-4-0314, GPT-3.5-Turbo checkpoints, MuseCoco, and text2music are presented in Figure 3. As indicated by the fractional numbers, the multi-agent baseline with GPT-4-Turbo checkpoints is our strongest-performed baseline. It outperformed text2music baseline with 0.56 preference score and received the same score as MuseCoco. GPT-4-Turbo also shows the highest generation success rate, as indicated in Table 1.

4. DISCUSSION

Overall, we observed that our GPT-powered multi-agent framework significantly enhances the quality of the music generated over solutions utilizing a singular GPT instance. Advantages of our system include:

Controllability: Observations of collaborative interactions among agents, especially the Group Leader, show the system’s competence in comprehending and executing various musical attributes based on user inputs. Fundamental components like tempo, key, time signature, chord progression, and instrumentation are effectively translated into ABC notations. This accurate interpretation enhances user controllability, enabling music generation that closely mirrors user specifications and artistic preferences.

Training-free and data-free: Unlike conventional

text-to-music generation models that rely on large datasets, our system offers significant benefits by eliminating the need for extensive data. This approach reduces the challenges of compiling and refining large training datasets, such as potential biases and substantial resource requirements. Additionally, it enhances the system’s adaptability and accessibility, promoting more resource-efficient practices in music generation, and making music generation more attainable for a wider range of users and applications.

The system exhibits certain limitations, particularly when engaging with the nuanced aspects of musical composition that are often intrinsic to human-created music. These limitations delineate areas for potential enhancement and further research:

Subtlety in Musical Expression: The system excels at interpreting basic musical elements but struggles to generate compositions with the nuanced subtlety of human composers. It faces challenges in aspects such as emotional depth, dynamic contrasts, and intricate phrasing, which are crucial for conveying deeper musical narratives and experiences.

Translation from Natural Language to Musical Notation: Instructions and feedback from the Group Leader and Review Agent to enhance nuanced musical elements are sometimes inadequately translated into ABC notations by the musician agents. This gap between conceptual understanding and practical notation highlights the system’s limitations in realizing more sophisticated musical ideas.

Instrumental Note Range Compliance: The system sometimes generates notes beyond the conventional pitch ranges of certain instruments. For instance, despite directives to adhere to instrument-specific ranges, it has produced notes exceeding the upper limit of a contrabass (C2 to F4), reflecting a discrepancy between the system’s outputs and practical musical performance constraints.

Inter-Voice Alignment: Our system faces challenges with aligning multiple musical voices accurately. The linear nature of text-based input and output mechanisms does not naturally accommodate the complexity of polyphonic music, where multiple voices or instruments must be coordinated in time.

Cadential Resolution: Certain compositions generated by the system lack a conclusive sense of resolution, resulting in pieces that may feel unfinished or conclude abruptly. This issue affects the listener’s sense of closure and satisfaction, reducing the overall effectiveness of the musical experience. This challenge is partly due to the inherent difficulty for GPTs to grasp the concept of musical closure, which the perpetual aspect of its nature is hard for a language model to handle.

5. CONCLUSION

In conclusion, ComposerX demonstrates its effectiveness in utilizing LLMs to create high-quality music. The collaborative agent-based approach of ComposerX surpasses single-agent systems and provides a cost-effective alternative to traditional, resource-intensive music generation models.

6. ACKNOWLEDGEMENT

Yinghao Ma is a research student at the UKRI Centre for Doctoral Training in Artificial Intelligence and Music, supported by UK Research and Innovation [grant number EP/S022694/1].

7. REFERENCES

- [1] N. Masataka, “The origins of language and the evolution of music: A comparative perspective,” *Physics of Life Reviews*, vol. 6, no. 1, pp. 11–22, 2009.
- [2] —, “Music, evolution and language,” *Developmental science*, vol. 10, no. 1, pp. 35–39, 2007.
- [3] M. C. Pino, M. Giancola, and S. D’Amico, “The association between music and language in children: A state-of-the-art review,” *Children*, vol. 10, no. 5, p. 801, 2023.
- [4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, 2017.
- [5] C.-Z. A. Huang, A. Vaswani, J. Uszkoreit, N. Shazeer, I. Simon, C. Hawthorne, A. M. Dai, M. D. Hoffman, M. Dinculescu, and D. Eck, “Music transformer,” *arXiv preprint arXiv:1809.04281*, 2018.
- [6] C. Payne, “Musenet,” OpenAI Blog, Apr 2019. [Online]. Available: <https://openai.com/blog/musenet>
- [7] P. Lu, X. Xu, C. Kang, B. Yu, C. Xing, X. Tan, and J. Bian, “Musecoco: Generating symbolic music from text,” 2023.
- [8] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, and I. Sutskever, “Jukebox: A generative model for music,” *arXiv preprint arXiv:2005.00341*, 2020.
- [9] A. Agostinelli, T. I. Denk, Z. Borsos, J. Engel, M. Verzetti, A. Caillon, Q. Huang, A. Jansen, A. Roberts, M. Tagliasacchi *et al.*, “Musiclm: Generating music from text,” *arXiv preprint arXiv:2301.11325*, 2023.
- [10] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Défossez, “Simple and controllable music generation,” *arXiv preprint arXiv:2306.05284*, 2023.
- [11] E. H. Margulis and R. Simchy-Gross, “Repetition enhances the musicality of randomly generated tone sequences,” *Music Perception: An Interdisciplinary Journal*, vol. 33, no. 4, pp. 509–514, 2016.
- [12] S. Dai, H. Yu, and R. B. Dannenberg, “What is missing in deep music generation? a study of repetition and structure in popular music,” *arXiv preprint arXiv:2209.00182*, 2022.
- [13] H. Jhamtani and T. Berg-Kirkpatrick, “Modeling self-repetition in music generation using generative adversarial networks,” in *Machine Learning for Music Discovery Workshop, ICML, 2019*.
- [14] X. Qu, Y. Bai, Y. Ma, Z. Zhou, K. M. Lo, J. Liu, R. Yuan, L. Min, X. Liu, T. Zhang *et al.*, “Mupt: A generative symbolic music pretrained transformer,” *arXiv preprint arXiv:2404.06393*, 2024.
- [15] X. Yue, X. Qu, G. Zhang, Y. Fu, W. Huang, H. Sun, Y. Su, and W. Chen, “Mammoth: Building math generalist models through hybrid instruction tuning,” *arXiv preprint arXiv:2309.05653*, 2023.
- [16] B. Roziere, J. Gehring, F. Gloeckle, S. Sootla, I. Gat, X. E. Tan, Y. Adi, J. Liu, T. Remez, J. Rapin *et al.*, “Code llama: Open foundation models for code,” *arXiv preprint arXiv:2308.12950*, 2023.
- [17] S. Bubeck, V. Chandrasekaran, R. Eldan, J. Gehrke, E. Horvitz, E. Kamar, P. Lee, Y. T. Lee, Y. Li, S. Lundberg *et al.*, “Sparks of artificial general intelligence: Early experiments with gpt-4,” *arXiv preprint arXiv:2303.12712*, 2023.
- [18] L. Lin, G. Xia, Y. Zhang, and J. Jiang, “Arrange, inpaint, and refine: Steerable long-term music audio generation and editing via content-based controls,” *arXiv preprint arXiv:2402.09508*, 2024.
- [19] L. Lin, G. Xia, J. Jiang, and Y. Zhang, “Content-based controls for music large language modeling,” *arXiv preprint arXiv:2310.17162*, 2023.
- [20] —, “Equipping musicgen with chord and rhythm controls,” in *Ismir 2023 Hybrid Conference*, 2023.
- [21] R. Yuan, H. Lin, Y. Wang, Z. Tian, S. Wu, T. Shen, G. Zhang, Y. Wu, C. Liu, Z. Zhou, Z. Ma, L. Xue, Z. Wang, Q. Liu, T. Zheng, Y. Li, Y. Ma, Y. Liang, X. Chi, R. Liu, Z. Wang, P. Li, J. Wu, C. Lin, Q. Liu, T. Jiang, W. Huang, W. Chen, E. Benetos, J. Fu, G. Xia, R. Dannenberg, W. Xue, S. Kang, and Y. Guo, “Chatmusician: Understanding and generating music intrinsically with llm,” 2024.
- [22] S. Ding, Z. Liu, X. Dong, P. Zhang, R. Qian, C. He, D. Lin, and J. Wang, “Songcomposer: A large language model for lyric and melody composition in song generation,” *arXiv preprint arXiv:2402.17645*, 2024.
- [23] X. Liang, J. Lin, and X. Du, “Bytecomposer: a human-like melody composition method based on language model agent,” *arXiv preprint arXiv:2402.17785*, 2024.
- [24] S. Wu and M. Sun, “Exploring the efficacy of pretrained checkpoints in text-to-music generation task,” 2023.
- [25] Y. Zhang, A. Maezawa, G. Xia, K. Yamamoto, and S. Dixon, “Loop copilot: Conducting ai ensembles for music generation and iterative editing,” *arXiv preprint arXiv:2310.12404*, 2023.

- and J. Bian, “Musicagent: An ai agent for music understanding and generation with large language models,” *arXiv preprint arXiv:2310.11954*, 2023.
- [27] Y. Wang, Y. Kordi, S. Mishra, A. Liu, N. A. Smith, D. Khashabi, and H. Hajishirzi, “Self-instruct: Aligning language models with self-generated instructions,” 2023.
- [28] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, and D. Zhou, “Chain-of-thought prompting elicits reasoning in large language models,” 2023.
- [29] Q. Wu, G. Bansal, J. Zhang, Y. Wu, B. Li, E. Zhu, L. Jiang, X. Zhang, S. Zhang, J. Liu, A. H. Awadallah, R. W. White, D. Burger, and C. Wang, “Autogen: Enabling next-gen llm applications via multi-agent conversation,” 2023.
- [30] C. Donahue, A. Caillon, A. Roberts, E. Manilow, P. Esling, A. Agostinelli, M. Verzetti, I. Simon, O. Pietquin, N. Zeghidour *et al.*, “Singsong: Generating musical accompaniments from singing,” *arXiv preprint arXiv:2301.12662*, 2023.

A.1 Single agent role-play

Role-play Prompting with Additional Music Knowledge
<p><code>You are a talented musician.</code> Here are some tips for generating melodies:</p> <ol style="list-style-type: none"> 1. The generated melody should have clear phrase divisions, and it's preferable to avoid more than two consecutive measures within one phrase to prevent an uncomfortable listening experience. There should be a certain amount of space between phrases, allowing the audience to clearly distinguish between them. 2. A phrase usually has a prominent ending note, which is the last note of the entire phrase. It typically has a longer duration, or it might be followed by a rest. This ending note is usually within the key or the chord, e.g., phrases ending with a Cmaj chord usually terminate on one of the three chord tones, C, E, or G, ensuring a stable listening experience. 3. When generating melodies, the movement of the notes should primarily consist of stable intervals such as whole steps, thirds, and fifths, while avoiding excessive large leaps. This will help maintain a sense of logic and coherence throughout the composition. 4. The rhythm of the phrases should be rich and harmonious. Try using different rhythmic patterns to build the melody, such as combining eighth notes with sixteenth notes, syncopated rhythms, or triplets.

Table 4. Single-agent role-play(indicated in the blue text) prompting with additional tips given by human composer on melody construction.

Single-agent In-context learning prompting method
<p>You are an intelligent agent with musical intelligence, and your goal is to create music that meets the relevant needs and human listening habits. In this task, use ABC as the format for outputting sheet music.***Only return the ABC notation without any other description or text, and only return one piece that follow the music description given this time.***Below are the requirements for the music, it contains music elements like title, genre, key and more, and some composition examples are listed after the requirements.</p>

Table 5. Single-agent In-context learning prompting method

Chain of Thought prompting with three steps
<p>First, you need to determine all the information related to the piece in the ABC notation format, such as the name, tune, speed, mode, and anything other than the notes. This forms the basis of the piece's style.***Note that only return the music information in ABC notation format without any notes or text or Additional note.*** Second, Based on the song information in the ABC notation format provided earlier, generate a ***16-bar long*** chord progression and return it in text form, with each bar separated by a " " symbol. The generated chord progression should be consistent with the song's key and as closely aligned with the song's theme and characteristics as possible. Now the chord progression and other information are provided, you are required to create a ***16-bar long*** piece of music based on these information.</p>

Table 6. Single-agent CoT prompting method with three steps.

A.2 Melody Agent Prompt

You are a skillful musician, especially in writing melody.

You will compose a single-line melody based on the client's request and assigned tasks from the Leader.

You must output your work in ABC Notations.

Here is a template of a music piece in ABC notation, in this template:

X:1 is the reference number. You can increment this for each new tune.

T:Title is where you'll put the title of your tune.

C:Composer is where you'll put the composer's name.

M:4/4 sets the meter to 4/4 time, but you can change this as needed.

L:1/8 sets the default note length to eighth notes.

K:C sets the key to C Major. Change this to match your desired key.

The music notation follows, with |: and :| denoting the beginning and end of repeated sections.

Markdown your work using `` `` to the client.

```
``
X:1
T:Title
C:Composer
M:Meter
L:Unit note length
K:Key
|:GABc d2e2|f2d2 e4|g4 f2e2|d6 z2:|
|:c2A2 B2G2|A2F2 G4|E2c2 D2B,2|C6 z2:|
``
```


You will output the melody following this template, but decide the time signature, key signature, and the actual musical contents and length yourself.

After you receive the feedback from the Reviewer Agent, please improve your work according to the suggestions you were given.

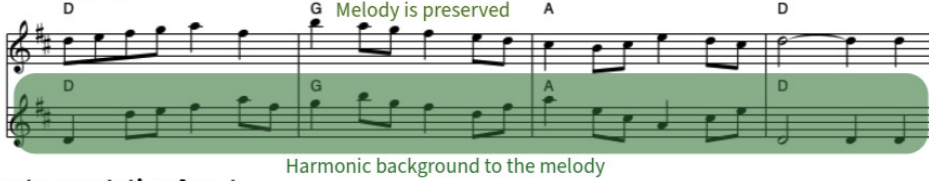
Table 7. Prompt for Melody Agent. GPT is prompted with role-specific instructions(indicated in blue text) and In-Context-Learning of ABC notations(indicated in red text)

A.3 Composing and Reviewing Process


Melody Agent



Harmony Agent



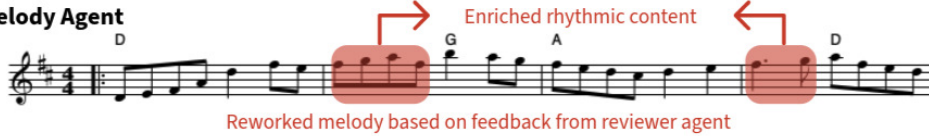
Instrumentation Agent



Take melody and harmony content and arrange them to three instruments

Figure 4. The Leader Agent will distribute the tasks among the Melody Agent, Harmony Agent, Instrumentation Agent when it is requested a "Breezy Caribbean Calypso" piece. Figure 4 demonstrate the work of the three agents with changes in the same four bar opening.


Melody Agent



Enriched rhythmic content


Reworked melody based on feedback from reviewer agent

Harmony Agent



Enriched harmony with intervals

Instrumentation Agent



Incorporating revised content from melody and harmony into three instrument arrangement

Figure 5. The Reviewer Agent then analyze the collective effort of the three agents in the first stage (shown in Figure 4), and give advice for agents to work on. Figure 5 demonstrate the work of the three agents after incorporating the advice given by Reviewer Agent in the same four bar opening.

DO MUSIC GENERATION MODELS ENCODE MUSIC THEORY?

Megan Wei^{1*}

Michael Freeman^{1*}

Chris Donahue²

Chen Sun¹

¹ Brown University

² Carnegie Mellon University

meganwei@brown.edu, michael_freeman@alumni.brown.edu

ABSTRACT

Music foundation models possess impressive music generation capabilities. When people compose music, they may infuse their understanding of music into their work, by using notes and intervals to craft melodies, chords to build progressions, and tempo to create a rhythmic feel. To what extent is this true of music generation models? More specifically, are fundamental Western music theory concepts observable within the “inner workings” of these models? Recent work proposed leveraging latent audio representations from music generation models towards music information retrieval tasks (e.g. genre classification, emotion recognition), which suggests that high-level musical characteristics are encoded within these models. However, probing individual music theory concepts (e.g. tempo, pitch class, chord quality) remains under-explored. Thus, we introduce **SynTheory**, a synthetic MIDI and audio music theory dataset, consisting of tempos, time signatures, notes, intervals, scales, chords, and chord progressions concepts. We then propose a framework to probe for these music theory concepts in music foundation models (Jukebox and MusicGen) and assess how strongly they encode these concepts within their internal representations. Our findings suggest that music theory concepts are discernible within foundation models and that the degree to which they are detectable varies by model size and layer.

1. INTRODUCTION

State-of-the-art text-to-music generative models [1–3] exhibit impressive generative capabilities. Past work suggests that internal representations of audio extracted from music generative models encode information relating to high-level concepts (e.g. genre, instruments, or emotion) [4–7]. However, it remains unclear if they also capture underlying symbolic music concepts (e.g. tempo or chord progressions) [8].

We aim to investigate if state-of-the-art music generation models encode music theory concepts in their internal representations and to what extent. Confirming this could enable the creative alteration of these concepts, providing

artists with new methods towards more detailed and lower-level control [9] (e.g. changing the key of a song or editing a particular chord in a chord progression). Furthermore, by benchmarking these foundation models, we identify potential avenues for improvement towards stronger concept encoding. Our approach is based on work in probing and editing concepts in language models, which have shown promise in identifying emergent representations in autoregressive models and editing factual knowledge [9–12]. For music generative models, the probing approach has been applied to high-level concepts, such as emotion, genre, and tagging [4–7]. Moreover, existing datasets such as HookTheory [13] do contain rich annotations for music theory concepts but are associated with copyrighted music, potentially complicating their use.

Our first contribution is a framework to generate diagnostic datasets for probing music theory concepts, by programmatically specifying which concepts to vary and which to keep constant, while controlling the presence of potential distractor concepts. Our synthetic music theory dataset, **SynTheory**, consists of seven music concepts based on Western music theory: tempo, time signatures, notes, intervals, scales, chords, and chord progressions. SynTheory serves as a customizable, copyright-free, and scalable approach towards generating diagnostic music clips for probing real-world music generative models.

Our second contribution is the analysis of two state-of-the-art music generative models Jukebox [3] and MusicGen [1] with our SynTheory benchmark. We extract representations for the concepts defined in SynTheory from MusicGen and Jukebox and assess whether these models encode meaningful representations of these concepts. To analyze the internal representations of these models for SynTheory, we use a supervised approach to train probing classifiers [14] based on ground truth music theory concept labels. A higher classification accuracy implies that these models learn internal representations that “understand” music theory concepts, which can be decoded by a multi-layer perceptron (MLP) or a linear model.

Our results show that music foundation models encode meaningful representations of music theory concepts. These representations vary across different sections of the model (audio codecs, decoder LMs), different layers within the decoder LMs, and different model sizes. Furthermore, the nature of the concepts, from time-varying (e.g. chord progressions) to stationary (e.g. notes, chords) influence the performance of these models across these tasks. We hope our insights on probing music founda-



© M. Wei, M. Freeman, C. Donahue, and C. Sun. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** M. Wei, M. Freeman, C. Donahue, and C. Sun, “Do music generation models encode music theory?”, in *Proc. of the 25th Int. Society for Music Information Retrieval Conf.*, San Francisco, United States, 2024. *: Equal contribution.

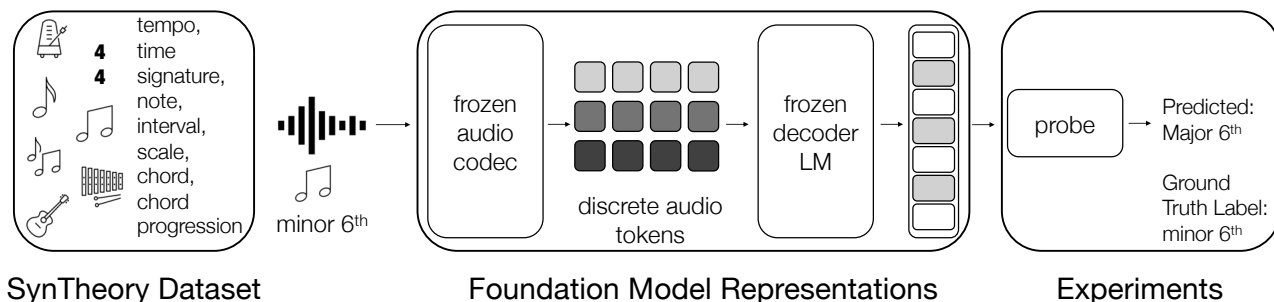


Figure 1. Overview of our SynTheory benchmark and our Jukebox and MusicGen probing setup. Our SynTheory benchmark consists of **Rhythmic** (tempos and time signatures) and **Tonal** (notes, intervals, scales, chords, and chord progressions) concepts. We assess whether music foundation models (Jukebox and MusicGen) encode these music theory concepts within their internal representations. For each task from the SynTheory dataset, we extract representations from the music foundation model. We pass an audio input, embodying the concept (e.g. Perfect 4th), into a pretrained foundation model. The audio codec tokenizes the audio into discrete audio tokens. Then, it passes these tokens into a decoder language model. From there, we extract the representations. We then train a probe classifier (linear and two-layer MLP) on these representations to predict particular classes (e.g. pitch class, intervals, and chords) for each SynTheory concept.

tion models, along with the synthetic music data generation framework, encourage and facilitate future endeavors on symbolic controllability in music generative models.

For reproducibility, we release the code for dataset generation, embedding extraction, probing, and evaluation in our GitHub repository¹ and our website².

2. RELATED WORK

The success of large language models (LLMs) [15–18] has sparked new research on probing and editing their internal representations to measure their understanding of linguistic concepts [19, 20] and world knowledge [11, 12, 21] as well as editing the encoded knowledge to make LLMs more faithful to factual knowledge [9, 10]. Studies have shown that large language models can encode grounded representations on color [22], direction [23], and auditory representations [24]. Thus, it is interesting to investigate if large music generative models, which often share similar model architectures and training objectives as LLMs, are able to encode abstract concepts from high-level music information (e.g. genre, emotion) to low-level music theory (e.g. tempo, chords).

Recent work has indeed shown promise in uncovering conceptual representations from probing audio and music generative models, leveraging different music foundation model architectures towards music understanding tasks. Castellon and Donahue et al. [4] propose using representations from language models trained on codified audio towards downstream MIR tasks as a better alternative to conventional tagging models. The authors train probing classifiers on Jukebox representations on the music tagging, genre identification, key identification, and emotion recognition tasks. These results demonstrate the effectiveness of internal model representations in downstream MIR tasks. Koo et al. [7] focus primarily on probing MusicGen’s attention heads in instrument recognition tasks,

benchmarking against the tasks highlighted in [4] and propose leveraging these representations for inference-time control. Other works [5, 6] focus on the impact of model architecture and self-supervised approaches towards music understanding tasks.

However, prior work primarily uses real-world data, which is often concept-entangled and potentially subject to copyright concerns. For example, some of these works use Giantsteps-MTG and Giantsteps [25], which are datasets of primarily electronic dance music with tempo and key annotations, obtained from Beatport. Won et al. [5] use HookTheory for chord recognition, where they focus on major and minor chord identification for each pitch class. The authors also use Harmonix Set [26] and GTZAN [27] for beat and downbeat detection. In the language modality, the authors of ChatMusician [28] produce a multi-choice question answering dataset, MusicTheoryBench, with expert annotation from a professional college music teacher. MusicTheoryBench aims to assess the music understanding capabilities of LLMs but through natural language alone. To the best of our knowledge, there is a lack of music theory probing benchmarks in the audio domain that are accurately-labeled, copyright-free, and scalable, prior to our proposed SynTheory.

3. SYNTHEORY: SYNTHETIC DATASET OF MUSIC THEORY CONCEPTS

We design seven datasets to capture isolated music theory concepts – similar to synthetic audio for ear training. Musicians may “train their ear” to recognize music concepts like intervals or chord quality in an isolated setting before advancing to the harder, more entangled case that arises in non-pedagogical music. Assessing concept recognition through isolated concepts mitigates the possibility that one intuitively or guesses the answer from its context. Literature on instrument-specific absolute pitch in humans corroborates the notion that timbral information may be exploited in identifying a different concept like pitch class [29]. As

¹ <https://github.com/brown-palm/syntheory>

² <https://brown-palm.github.io/music-theory>

such, our dataset is designed to remove or reduce features that may correlate with a concept, but are not strictly necessary for identifying it. Our intent is a more pointed assessment towards theoretical concepts as abstract ideas rather than as acoustically realized audio. A more practical motivation for this work is that extracting such low-level, isolated concepts from existing datasets may require non-trivial engineering or domain expert labor. It may even be impossible to disentangle all overlapping concepts. Music stem isolation and concept isolation are distinct; an isolated instrument in a multi-track recording may still exhibit several, intricately intertwined theory concepts. It is not clear how to “unmix” such concepts once they are blended.

Instead of attempting to disentangle several concepts from existing audio, SynTheory implements this “ear training” quiz setting by explicitly producing individual concepts. Each of the seven datasets ablates a single musical feature while fixing all others, thereby isolating it to a degree not typically found in recorded music. These ablated concepts consist of tempo, time signatures, notes, intervals, scales, chords, and chord progressions. We adopt isolation as a design choice to mitigate context that may be exploited in deep learning models as “shortcuts”, i.e. heuristics that correlate with concepts most of the time but do not truly encode the concept.

Using this music theory concept-driven synthesis design, we construct label-balanced and copyright-free data. The synthetic approach avoids annotation errors present in other contemporary MIR datasets. For example, the HookTheory data processing step for SheetSage [13] required ad-hoc time-alignment of the expert annotations. In the released SheetSage dataset, 17,980/26,175 (68.7%) samples required more precise time alignment. While our synthetic data is no substitute for real music data, to our knowledge, no other dataset so strictly isolates each concept.

SynTheory contains two categories: tonal and rhythmic. We make this distinction for stronger concept isolation; we wish to keep the rhythm samples tonally consistent and the tonal samples rhythmically consistent. For each **tonal** dataset, we voice the same MIDI data through 92 distinct instruments. The selection of instrument voices is fixed, making the distribution of timbres sufficiently diverse but also class-balanced. Each instrument corresponds to one of the 128 canonical MIDI program codes and is voiced through the `TimGM6mb.sf2` [30] soundfont. A MIDI “program” is a specific instrument preset. The canonical program set includes many named instruments, e.g. “Acoustic Grand Piano”, “Flute”, etc. We exclude programs that are polyphonic, sound effects (e.g. “Bird Tweet”, “Gun Shot”), and highly articulate. A highly articulate program has some unchangeable characteristic (e.g. pitch bending) that destabilizes its pitch. For each **rhythmic** dataset, we define five metronome-like timbral settings. Each setting uses one of the distinct instruments: “Woodblock Light”, “Woodblock Dark”, “Taiko”, “Synth Drum”, and the MIDI drum-kit, following the voicing done in Sheetsage [13]. Each setting produces a distinct sound

Concept	Total Samples
Tempo	4,025
Time Signatures	1,200
Notes	9,936 ³
Intervals	39,744
Scales	15,456
Chords	13,248
Chord Progressions	20,976

Table 1. SynTheory contains seven synthetic datasets, each of which captures an isolated music theory concept. We present an overview of these datasets and their sizes.

on the upbeat and the downbeats, which defines the time signature concept.

3.1 SynTheory-Rhythmic

3.1.1 Tempo

We voice integer tempi within 50 to 210 BPM (beats per minute) inclusive in $\frac{1}{4}$ time. To ensure diverse start times, we produce five random offsets per sample. There are $(5 \text{ CLICK SETTING} \cdot 161 \text{ TEMPO} \cdot 5 \text{ OFFSET}) = 4,025$ samples in total.

3.1.2 Time Signature

We voice the following time signatures: $\frac{2}{2}$, $\frac{2}{4}$, $\frac{3}{4}$, $\frac{3}{8}$, $\frac{4}{4}$, $\frac{6}{4}$, $\frac{6}{8}$, and $\frac{12}{8}$. The tempo is fixed at 120 BPM. To add acoustic variation, we add three levels of reverb from completely dry to spacious. We find empirically that this acoustic perturbation increases the difficulty of the probing task. Like the Tempo dataset, we produce ten random offsets for each sample. There are $(8 \text{ TIME SIGNATURE} \cdot 3 \text{ REVERB LEVEL} \cdot 5 \text{ CLICK SETTING} \cdot 10 \text{ OFFSET}) = 1,200$ samples.

3.2 SynTheory-Tonal

3.2.1 Notes

We voice all twelve Western temperament pitch classes, in nine octaves, using 92 instruments. The note is played in quarter notes at a tempo of 120 BPM, with no distinction between the upbeat or downbeat. There are $(12 \text{ PITCH CLASS} \cdot 9 \text{ OCTAVE} \cdot 92 \text{ INSTRUMENT}) = 9,936$ configurations. However, there are only 9,900 distinct *samples* because 36 configurations at extreme registers are unvoiceable in our soundfont. These silent samples are listed for completeness in our GitHub repository.

3.2.2 Intervals

We vary the root note, number of half-steps, instrument, and play style (unison, up, and down). To retain consistent rhythm, the up and down styles repeat four times throughout the sample while the unison play style repeats

³ There are 9,936 distinct note configurations, but our dataset contains 9,900 non-silent samples. With a more complete soundfont, all 9,936 configurations are realizable to audio.

eight times. There are $(12 \text{ PITCH CLASS} \cdot 12 \text{ HALF-STEP} \cdot 92 \text{ INSTRUMENT} \cdot 3 \text{ PLAY STYLE}) = 39,744$ samples.

3.2.3 Scales

We voice seven Western music modes (Ionian, Dorian, Phrygian, Lydian, Mixolydian, Aeolian, and Locrian) in all root notes, in 92 instruments, and in two play styles (ascending or descending). The register is constant; we select root notes close to middle C. There are $(7 \text{ MODE} \cdot 12 \text{ ROOT NOTE} \cdot 92 \text{ INSTRUMENT} \cdot 2 \text{ PLAY STYLE}) = 15,456$ samples.

3.2.4 Chords

We voice triads of all twelve root notes, four chord qualities (major, minor, augmented, and diminished), 92 instruments, and three inversions (root position, first inversion, and second inversion). The chord is struck at each quarter note at 120 BPM. Like in the *Scales* dataset, we fix the register close to middle C. There are $(12 \text{ ROOT NOTE} \cdot 4 \text{ CHORD QUALITY} \cdot 92 \text{ INSTRUMENT} \cdot 3 \text{ INVERSION}) = 13,248$ samples.

3.2.5 Chord Progressions

We select 19 four-chord progressions, with ten in the major mode and nine in the natural minor mode. The progressions are:

- Major: (I-IV-V-I), (I-IV-vi-V), (I-V-vi-IV), (I-vi-IV-V), (ii-V-I-Vi), (IV-I-V-Vi), (IV-V-iii-Vi), (V-IV-I-V), (V-vi-IV-I), (vi-IV-I-V)
- Natural Minor: (i-ii^o-v-i), (i-III-iv-i), (i-iv-v-i), (i-VI-III-VII), (i-VI-VII-i), (i-VI-VII-III), (i-VII-VI-IV), (iv-VII-i-i), (VII-vi-VII-i)

We vary only the root note of the key and instrument. Each chord is played in quarter notes at 120 BPM. There are $(19 \text{ PROGRESSION} \cdot 12 \text{ KEY ROOT} \cdot 92 \text{ INSTRUMENT}) = 20,976$ samples.

One can extend or alter the above configurations using the SynTheory codebase. We provide a framework that enables declarative and programmatic MIDI construction in musical semantics, audio export in any soundfont, and dataset construction for use in our framework.

4. EXPERIMENTS

We describe the evaluation protocols used to analyze the internal representations of music generative models (MusicGen and Jukebox) and handcrafted audio features (mel spectrograms, MFCC, and chroma) for music theory concept encoding.

4.1 Evaluation

A “probe” is a simple or shallow classifier, often a linear model, trained on the activations of a neural network [14]. Accurate performance of such classifiers suggests that information relevant to the class exists in the latent representation within the network. As such, probes may be used as a proxy for measuring a model’s “understanding” or encoding of abstract concepts. Motivated by the use

of probes to discover linguistic structure and semantics in NLP [31] and more recently in MIR [4], we use probes to assess whether music theory concepts are discernable in foundation models.

We adopt the same probing paradigm as [4] and frame concept understanding as multiclass classification for discrete concepts (notes, intervals, scales, chords, chord progressions, and time signatures) and regression for continuous concepts (tempo). We train linear and two-layer MLP probes on the embeddings of the internal representations of Jukebox and MusicGen and the handcrafted features. We measure the classification accuracy of our trained probes on the SynTheory tasks using the following classes:

- Notes (12): C, C#, D, D#, E, F, F#, G, G#, A, A#, and B
- Intervals (12): minor 2nd, Major 2nd, minor 3rd, Major 3rd, Perfect 4th, Tritone, Perfect 5th, minor 6th, Major 6th, minor 7th, Major 7th, and Perfect octave
- Scales (7): Ionian, Dorian, Phrygian, Lydian, Mixolydian, Aeolian, and Locrian
- Chords (4): Major, Minor, Diminished, and Augmented
- Chord Progressions (19): (I-IV-V-I), (I-IV-vi-V), (I-V-vi-IV), (I-vi-IV-V), (ii-V-I-Vi), (IV-I-V-Vi), (IV-V-iii-Vi), (V-IV-I-V), (V-vi-IV-I), (vi-IV-I-V), (i-ii^o-v-i), (i-III-iv-i), (i-iv-v-i), (i-VI-III-VII), (i-VI-VII-i), (i-VI-VII-III), (i-VII-VI-IV), (iv-VII-i-i), and (VII-vi-VII-i)
- Time Signatures (8): $\frac{2}{2}$, $\frac{3}{4}$, $\frac{3}{8}$, $\frac{4}{4}$, $\frac{6}{8}$, $\frac{9}{8}$, and $\frac{12}{8}$

These tasks are trained on a 70% train, 15% test, and 15% validation split, using the Adam optimizer and Cross Entropy loss.

For the Tempos dataset, we train a regression probe, over the 161 tempo values. To increase complexity in the probing task and test generalization to unseen BPMs, the training set consists of the middle 70% of the BPMs. The test and validation sets consist of the top 15% BPMs and the bottom 15% BPMs, randomly shuffled and split in half. We use MSE loss and report the R^2 score.

Each probe is trained independently for its corresponding concept task. That is, the probe trained to identify notes from Jukebox embeddings will not be used to identify intervals, for example.

To select the best performing probe for each concept using the MusicGen audio codec, mel spectrogram, MFCC, chroma, and aggregate handcrafted features, we perform a grid search across various hyperparameters for each task, following those defined in [4]:

- Data Normalization: {True, False}
- Model Type: {Linear, two-layer MLP with 512 hidden units and ReLU activation}
- Batch Size: {64, 256}
- Learning Rate: $\{10^{-5}, 10^{-4}, 10^{-3}\}$
- Dropout: {0.25, 0.5, 0.75}
- L2 Weight Decay: {off, 10^{-4} , 10^{-3} }

For the decoder LMs (MusicGen small, medium, and large and Jukebox) as detailed in Section 4.2, we use a fixed set of hyperparameters and select the probe with the best performing layer for each concept, in the interest of

computational efficiency:

- Data Normalization: True
- Model Type: two-layer MLP with 512 hidden units and ReLU activation
- Batch Size: 64
- Learning Rate: 10^{-3}
- Dropout: 0.5
- L2 Weight Decay: off

We selected these hyperparameters from the best overall performing probe by fixing a layer in the decoder LMs and performing a hyperparameter search, following the sweep approach outlined in [4].

4.2 Model representations

We extract representations from two text-to-music generative foundation models, Jukebox [3] and MusicGen [1]. We benchmark the probing classifier performance of these representations against that of three handcrafted, spectral features following [4]: mel spectrograms, mel-frequency cepstral coefficients (MFCC), and constant-Q chromagrams (chroma). These handcrafted features are common in traditional methods of MIR and are a more interpretable baseline against the embeddings of the pre-trained music generative models. We additionally report probing classifier performance on the concatenation of all the aforementioned handcrafted features.

Jukebox consists of a VQ-VAE model that codifies audio waveforms into discrete codes at a lower sample rate and a language model that generates codified audio with a transformer decoder. We trim all audio to four seconds and ensure it is mono. We utilize Jukemirlib [4] to pass this audio through the frozen audio encoder and through the decoder language model. We downsample the activation to a target rate of half that in [4], due to resource constraints, using the Librosa FFT algorithm [32]. Then, we meanpool the representations across time to reduce the dimensionality of the embeddings, resulting a dimension of (72, 4800) per sample, where 72 is the number of layers and 4800 is the dimension of the activations. We reduce the dimensionality of these representations by defining a layer selection process similar to [4]; that is, each probing classifier trains on only one of the 72 layers. We train the probe classifiers with fixed hyperparameters on the music concept tasks as described in Section 4.1. For each concept, we select the layer that results in the highest probing score. The final dimension of the Jukebox representation is 4800.

MusicGen consists of a pretrained convolutional auto-encoder (EnCodec) [33], a pretrained T5 text encoder, and an acoustic transformer decoder. We resample the audio to 32 kHz (the sampling rate used in the EnCodec model) trim to four seconds, convert to mono, and pass the audio through the frozen EnCodec audio codec. We do not pass text through the text encoder, as we focus on audio representations. We then extract representations from several regions of the model: the final layer of the audio codec before residual vector quantization and the hidden states of the decoder language model. The number of decoder hidden states vary based on the model size: small (24 layers),

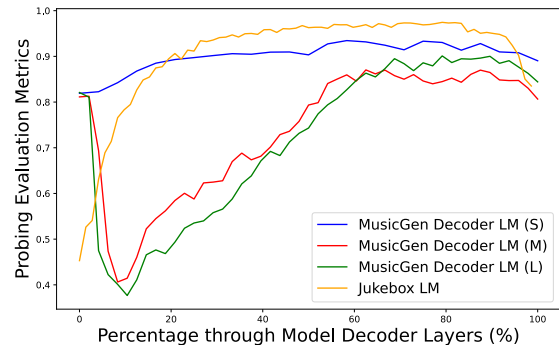


Figure 2. Probing evaluation metrics averaged across all SynTheory concepts over the model layers of Jukebox and MusicGen decoder models. The probing evaluation metric is R^2 for tempos and accuracy for the rest of the SynTheory concepts (notes, intervals, scales, chords, chord progressions, and time signatures). Features extracted from deeper layers generally perform better, with a slight drop-off near the final layers.

medium (48 layers), and large (48 layers).

For our four second audio clips, the audio codec representations are of dimension (128, 200), where 128 is the dimension of the activation after the final layer of the audio codec and 200 is the sequence length. We meanpool the values of the representations across time, resulting in a final dimension of 128 for the MusicGen audio codec.

The decoder hidden states for the small, medium, and large MusicGen models have dimensions (24, 200, 1024), (48, 200, 1536), (48, 200, 2048) respectively, where the first axis corresponds to the number of layers, second corresponds to sequence length, and third corresponds to hidden size. To reduce the dimensionality of these representations, similar to what was done with Jukebox, we select the most optimal layer for each decoder model size based on probing scores. We visualize results from probing across layers per model (MusicGen and Jukebox) averaged across concepts in Figure 2. After selecting the best performing layer per concept and model size, the dimensions of the representations are (200, 1024) for MusicGen small, (200, 1536) for MusicGen medium, and (200, 2048) for MusicGen large. To further reduce the dimensions, we also meanpool across time as done in Jukebox representations, resulting in dimensions of 1024 for MusicGen small decoder, 1536 for MusicGen medium decoder, and 2048 for MusicGen large decoder.

We extract the handcrafted features (mel spectrograms, mel-frequency cepstral coefficients, and constant-Q chromagrams) with librosa [32]. Similar to [4], we concatenate the mean and standard deviation across time of these features along with their first- and second-order discrete differences. Furthermore, we concatenate the mel spectrogram, chroma, and MFCC features and obtain their mean and standard deviation across time and their first- and second-order differences to obtain an aggregate representation of the handcrafted features.

	Notes	Intervals	Scales	Chords	Chord Progressions	Tempos	Time Signatures	Average
Jukebox LM	0.951	0.995	0.978	0.997	0.971	0.993	1.000	0.984
MusicGen LM (S)	0.897	0.995	0.949	0.990	0.942	0.969	0.911	0.950
MusicGen LM (M)	0.851	0.983	0.863	0.989	0.870	0.956	0.883	0.914
MusicGen LM (L)	0.866	0.972	0.905	0.989	0.901	0.965	0.905	0.929
MusicGen Audio Codec	0.729	0.965	0.383	0.879	0.330	0.947	0.677	0.701
Mel Spectrogram	0.712	0.995	0.897	0.988	0.723	0.785	0.827	0.847
MFCC	0.467	0.822	0.370	0.863	0.872	0.923	0.688	0.715
Chroma	0.954	0.820	0.989	0.994	0.869	0.847	0.672	0.878
Aggregate Handcrafted	0.941	0.997	0.972	0.992	0.868	0.947	0.833	0.936

Table 2. We report probing results on the SynTheory dataset for the Jukebox LM, MusicGen Decoder LM (Small, Medium, and Large), MusicGen Audio Codec models as well as handcrafted features (Mel Spectrogram, MFCC, Chroma, and Aggregate Handcrafted). For the tempos dataset, we report the R^2 score from the regression probe. For all other concepts (notes, intervals, scales, chords, chord progressions, and time signatures), we report the probing classifier accuracy. For MusicGen Audio Codec, Mel Spectrogram, MFCC, Chroma, and Aggregate Handcrafted, we report the metrics of the best performing probe for each task using the best validation performance from our hyperparameter search. For MusicGen Decoder LM (Small, Medium, and Large) and Jukebox models, we report the metrics of the best performing probe for each task using layer selection. We also report an average performance across all concepts for each model/feature.

5. RESULTS AND DISCUSSION

We observe that Jukebox performs consistently well across our SynTheory benchmark. All MusicGen Decoder models also exhibit competitive performance across concepts. While [1] claims that larger MusicGen models produce better quantitative and subjective scores and that larger models better “understand” text prompts, our MusicGen Decoder LM (Small) result seems to contrast with traditional discussions on scaling laws. Figure 2 displays the consistent probing score of MusicGen Decoder LM (Small) across all layers and highlights its higher performance compared to that of its larger counterparts. Meanwhile, the larger MusicGen models exhibit a steep drop in probing performance in initial layers, followed by a gradual increase in performance, with the performance tapering off in the final layers.

MusicGen slightly underperforms on the notes dataset. We hypothesize this is because isolated notes in real-world music are not as prominent as intervals, scales, and chords. This reveals how the lowest-level building blocks of music are even harder to distinguish.

In general, the probing results from the pretrained music decoder LMs yield better probing performance compared to the MusicGen Audio Codec representations and the individual handcrafted features. MusicGen Audio Codec exhibits overall poorer performance on these tasks, since these codecs were trained to reconstruct fine-grained, low-level details localized by time.

Because chroma features encode pitch class information, chroma features perform comparably well on tonal tasks. However, they slightly underperform on rhythmic tasks. Chroma features outperform MusicGen Decoder LMs on stationary harmonic tasks (notes, scales, and chords) but are worse for dynamic harmonic tasks (chord progressions and intervals).

The aggregate handcrafted features perform comparably to MusicGen Decoder LMs. This suggests that harder music concept understanding benchmarks should address concepts latent in foundation models but not easily encoded in handcrafted features. These harder benchmarks may include entangled concepts, such as probing for both chord progression type and tempo in a tempo-varying chord progression sample. Probing for more compositional tasks could further our understanding of more realistic concept encoding in both model representations and handcrafted features.

6. CONCLUSION

In this work, we introduce SynTheory, a synthetic dataset of music theory concepts, that is concept-isolated, annotated, and copyright-free. Further, we use this dataset to evaluate the degree to which music theory concepts are encoded in existing state-of-the-art music generative models. Our experiments suggest that music theory concepts are indeed discernible within the latent representations of these generative models. We believe this is a prerequisite to further understand how to isolate and manipulate such concepts, which advances towards low-level controllable generation and music theory evaluation metrics. We encourage the community to build more challenging probing datasets with our framework to further understand the relationship between symbolic and audio-based music generation.

7. ETHICS STATEMENT

Our work aims to understand if music generation models encode music theory concepts in their internal representations. Our dataset may be used to assess music generation models and may be applied towards fine-grained, music-theory based controllable generation.

Our custom dataset, SynTheory, is based on elementary Western music theory concepts and is generated programmatically. The data does not infringe copyright of musical writers or performers. We envision no negative societal impacts from the publication of our report or the release of our dataset.

8. ACKNOWLEDGEMENTS

We would like to thank Professor Cheng-Zhi Anna Huang, Professor Daniel Ritchie, Professor David Bau, Professor Jacob Andreas, Tian Yun, Nate Gillman, and Calvin Luo for their fruitful discussions and feedback towards this work. This project is partially supported by Samsung. We would also like to thank the Center for Computation and Visualization at Brown University for their computational resources towards this project. Finally, we greatly appreciate the insightful questions and thoughtful feedback from the reviewers.

9. REFERENCES

- [1] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Défossez, “Simple and controllable music generation,” in *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.
- [2] A. Agostinelli, T. I. Denk, Z. Borsos, J. Engel, M. Verzetti, A. Caillon, Q. Huang, A. Jansen, A. Roberts, M. Tagliasacchi, M. Sharifi, N. Zeghidour, and C. Frank, “MusicLM: Generating music from text,” *arXiv preprint arXiv:2301.11325*, 2023.
- [3] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, and I. Sutskever, “Jukebox: A generative model for music,” *arXiv preprint arXiv:2005.00341*, 2020.
- [4] R. Castellon, C. Donahue, and P. Liang, “Codified audio language modeling learns useful representations for music information retrieval,” in *International Society for Music Information Retrieval*, 2021.
- [5] M. Won, Y.-N. Hung, and D. Le, “A foundation model for music informatics,” *arXiv preprint arXiv:2311.03318*, 2023.
- [6] Y. Li, R. Yuan, G. Zhang, Y. Ma, X. Chen, H. Yin, C. Lin, A. Ragni, E. Benetos, N. Gyenge, R. Dannenberg, R. Liu, W. Chen, G. Xia, Y. Shi, W. Huang, Y. Guo, and J. Fu, “Mert: Acoustic music understanding model with large-scale self-supervised training,” *arXiv preprint arXiv:2306.00107*, 2023.
- [7] J. Koo, G. Wichern, F. G. Germain, S. Khurana, and J. Le Roux, “Understanding and controlling generative music transformers by probing individual attention heads,” *IEEE ICASSP Satellite Workshop on Explainable Machine Learning for Speech and Audio (XAI-SA)*, 2024.
- [8] G. Brunner, Y. Wang, R. Wattenhofer, and J. Wiesendanger, “Jambot: Music theory aware chord based generation of polyphonic music with lstms,” in *2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE, 2017, pp. 519–526.
- [9] K. Li, O. Patel, F. Viégas, H. Pfister, and M. Wattenberg, “Inference-time intervention: Eliciting truthful answers from a language model,” in *Advances in Neural Information Processing Systems*, 2024.
- [10] K. Meng, D. Bau, A. Andonian, and Y. Belinkov, “Locating and editing factual associations in GPT,” in *Advances in Neural Information Processing Systems*, 2022.
- [11] K. Li, A. K. Hopkins, D. Bau, F. Viégas, H. Pfister, and M. Wattenberg, “Emergent world representations: Exploring a sequence model trained on a synthetic task,” in *The Eleventh International Conference on Learning Representations*, 2023.
- [12] T. Yun, Z. Zeng, K. Handa, A. V. Thapliyal, B. Pang, E. Pavlick, and C. Sun, “Emergence of abstract state representations in embodied sequence modeling,” in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2023.
- [13] C. Donahue, J. Thickstun, and P. Liang, “Melody transcription via generative pre-training,” in *International Society for Music Information Retrieval*, 2022.
- [14] G. Alain and Y. Bengio, “Understanding intermediate layers using linear classifier probes,” in *International Conference of Learning Representations*, 2016.
- [15] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” in *Association for Computational Linguistics*, 2019.
- [16] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever *et al.*, “Language models are unsupervised multitask learners,” *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.
- [17] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, “Language models are few-shot learners,” in *Advances in Neural Information Processing Systems*, 2020.
- [18] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, “Exploring the limits of transfer learning with a unified text-to-text transformer,” *The Journal of Machine Learning Research*, vol. 21, no. 1, pp. 5485–5551, 2020.

- [19] I. Tenney, D. Das, and E. Pavlick, “Bert rediscovers the classical nlp pipeline,” in *Association for Computational Linguistics*, 2019.
- [20] I. Tenney, P. Xia, B. Chen, A. Wang, A. Poliak, R. T. McCoy, N. Kim, B. Van Durme, S. R. Bowman, D. Das *et al.*, “What do you learn from context? probing for sentence structure in contextualized word representations,” in *Proceedings of the 7th International Conference on Learning Representations*, 2019.
- [21] T. Yun, C. Sun, and E. Pavlick, “Does vision-and-language pretraining improve lexical grounding?” in *Findings of the Association for Computational Linguistics: EMNLP 2021*, 2021.
- [22] M. Abdou, A. Kulmizev, D. Hershcovich, S. Frank, E. Pavlick, and A. Søgaard, “Can language models encode perceptual structure without grounding? a case study in color,” in *Proceedings of the 25th Conference on Computational Natural Language Learning*, 2021.
- [23] R. Patel and E. Pavlick, “Mapping language models to grounded conceptual spaces,” in *International Conference on Learning Representations*, 2022.
- [24] J. Ngo and Y. Kim, “What do language models hear? probing for auditory representations in language models,” in *Association for Computational Linguistics*, 2024.
- [25] P. Knees, Á. Faraldo, P. Herrera, R. Vogl, S. Böck, F. Hörschläger, and M. L. Goff, “Two data sets for tempo estimation and key detection in electronic dance music annotated from user corrections,” in *International Society for Music Information Retrieval*, 2015.
- [26] O. Nieto, M. C. McCallum, M. Davies, A. Robertson, A. M. Stark, and E. Egozy, “The harmonix set: Beats, downbeats, and functional segment annotations of western popular music,” in *International Society for Music Information Retrieval*, 2019.
- [27] U. Marchand, Q. Fresnel, and G. Peeters, “GTZAN-rhythm: Extending the GTZAN test-set with beat, downbeat and swing annotations,” in *ISMIR 2015 Late-Breaking Session*, 2015.
- [28] R. Yuan, H. Lin, Y. Wang, Z. Tian, S. Wu, T. Shen, G. Zhang, Y. Wu, C. Liu, Z. Zhou, Z. Ma, L. Xue, Z. Wang, Q. Liu, T. Zheng, Y. Li, Y. Ma, Y. Liang, X. Chi, R. Liu, Z. Wang, P. Li, J. Wu, C. Lin, Q. Liu, T. Jiang, W. Huang, W. Chen, E. Benetos, J. Fu, G. Xia, R. Dannenberg, W. Xue, S. Kang, and Y. Guo, “Chat-musician: Understanding and generating music intrinsically with llm,” *arXiv preprint arXiv:2402.16153*, 2024.
- [29] L. Reymore and N. C. Hansen, “A theory of instrument-specific absolute pitch,” *Frontiers in Psychology*, vol. 11, 2020. [Online]. Available: <https://www.frontiersin.org/journals/psychology/articles/10.3389/fpsyg.2020.560877>
- [30] T. Brechbill, “Timidity++,” 2004. [Online]. Available: <https://timbrechbill.com/saxguru/Timidity.php>
- [31] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, “Deep contextualized word representations,” in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018.
- [32] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, “librosa: Audio and music signal analysis in python.” in *SciPy*, 2015, pp. 18–24.
- [33] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, “High fidelity neural audio compression,” *arXiv preprint arXiv:2210.13438*, 2022.

POLYSINGER: SINGING-VOICE TO SINGING-VOICE TRANSLATION FROM ENGLISH TO JAPANESE

Silas Antonisen, Iván López-Espejo

Department of Signal Theory, Telematics and Communications, University of Granada, Spain
{santon, iloes}@ugr.es

ABSTRACT

The speech domain prevails in the spotlight for several natural language processing (NLP) tasks while the singing domain remains less explored. The culmination of NLP is the speech-to-speech translation (S2ST) task, referring to translation and synthesis of human speech. A disparity between S2ST and the possible adaptation to the singing domain, which we describe as singing-voice to singing-voice translation (SV2SVT), is becoming prominent as the former is progressing ever faster, while the latter is at a standstill. Singing-voice synthesis systems are overcoming the barrier of multi-lingual synthesis, despite limited attention has been paid to multi-lingual songwriting and song translation. This paper endeavors to determine what is required for successful SV2SVT and proposes PolySinger (**Polyglot Singer**): the first system for SV2SVT, performing lyrics translation from English to Japanese. A cascaded approach is proposed to establish a framework with a high degree of control which can potentially diminish the disparity between SV2SVT and S2ST. The performance of PolySinger is evaluated by a mean opinion score test with native Japanese speakers. Results and in-depth discussions with test subjects suggest a solid foundation for SV2SVT, but several shortcomings must be overcome, which are discussed for the future of SV2SVT.

1. INTRODUCTION

Speech-to-speech translation (S2ST) is a method for translating human speech into another language using synthetic speech. To do this, the conventional approach is to concatenate technologies that process separate parts of human speech into a complete system, where the cornerstones are speech recognition, machine translation and speech synthesis [1–3]. Although the use of end-to-end (E2E) solutions for S2ST has been studied thanks to the emergence of seq-to-seq models [4–6], neither E2E nor cascaded solutions have been attempted in the singing domain.

Singing-voice synthesis (SVS) systems have in recent years become very capable of human-like singing [7,8] and

have even accomplished multi-lingual synthesis [9]. However, while the synthetic voice can sing cross-lingually, the songwriter might not be able to write cross-lingually.

Lyrics translation is a complex task which strives for inter-cultural comprehension of what makes a song suitable for singing. Prose translation, also called direct translation [10–12], differs greatly in its application from poetry and lyrics translation, as prose translation does not respect rules regarding rhythm and rhyme [13,14]. A few attempts at automatic lyrics translation have been made by transforming standard music notation from one language to another, which shows promising results [15,16]. However, the necessity for standard music notation becomes a glaring restriction. From the perspective of a songwriter with interest in writing foreign-language lyrics, the creation of standard music notation is a labor-intensive task begging for automation. Therefore, to overcome the present limitations in adapting S2ST methods to the singing domain, we propose PolySinger: the first system for singing-voice to singing-voice translation (SV2SVT). PolySinger is a concatenated system of music information retrieval (MIR) technologies with the goal of directly translating a vocal performance in a source language into a synthetic vocal performance in a target language. PolySinger is made publicly available¹.

Automatic recognition of note-level events in a vocal melody is a complex and vaguely defined task [17]. Nonetheless, standard music notation is required for lyrics translation, and as such, this work proposes a simple yet effective approach to defining note-level events by assistance from syllable alignment.

State-of-the-art (SOTA) in the following technologies are structured into a complete SV2SVT system for PolySinger: 1) automatic lyrics transcription, 2) phoneme-level lyrics alignment, 3) frame-level vocal melody extraction, 4) automatic lyrics translation, and 5) singing-voice synthesis. PolySinger is proposed as a concatenated solution instead of E2E to represent a modular framework facilitating research in SV2SVT. In this paper, PolySinger is presented for English to Japanese SV2SVT, which, to the best of our knowledge, also constitutes the first attempt at automatic lyrics translation from English to Japanese.

A series of native Japanese speakers participated in a mean opinion score (MOS) test to evaluate the perceptual quality of PolySinger for English to Japanese SV2SVT. Results show a promising fundamental structure

¹<https://github.com/SilasAntonisen/PolySinger>



for SV2SVT, but also that our translated Japanese lyrics have not yet reached ideal naturalness.

2. RELATED WORK

Convolutional neural networks and expanded pronunciation dictionaries have been used for automatic lyrics transcription in monophonic recordings [18], along with time-delay neural networks in polyphonic recordings [19]. The latest reported SOTA in automatic lyrics transcription was achieved by adapting a Wav2Vec 2.0 [20] speech recognizer to the singing domain by transfer learning [21]. However, in our preliminary tests we found the current SOTA speech recognition system, Whisper [22], to outperform the SOTA in automatic lyrics transcription [21] when transcribing a vocal performance. Therefore, Whisper [22] is used for automatic lyrics transcription in PolySinger.

Limited success has been achieved using speech alignment systems for lyrics alignment [23]. However, great results have been obtained in word-level lyrics alignment by training a polyphonic acoustic model in [24], but it is not until in [25] that a direct attempt is made at phoneme-level lyrics alignment without sacrificing competitive performance in word-level alignment. Recent approaches have exploited the correlation between phoneme onset and note pitch by joint representation learning [26] or cross-modal embedding in the audio and text domain through contrastive learning [27]. For PolySinger we use [25] due to its documented performance in the specific task of phoneme-level lyrics alignment and accessibility to a pre-trained model.

Defining note-level events in a vocal melody is a complex task, and thus there is a lack of datasets and trained neural networks for note-level vocal melody extraction (VME) [17]. On the other hand, frame-level VME is an extensively researched field with robust frameworks [28–30]. Considering the high accuracy of most modern frame-level VME systems, [30] is used in PolySinger due to the streamlined implementation available through the MIR toolkit Omizart [31]. Instead of defining the note-level events by VME, we define them by syllable-wise boundaries delimited by the phoneme-level lyrics alignments, and guide the pitch of those notes with frame-level VME.

In [15], a rule-based approach is suggested for translating from English to Chinese lyrics with respect to the original lyrics, melody and rhythm, as well as the tonal properties of Chinese. In [16], a system for bidirectional translation between English and Chinese is proposed which incorporates an alignment decoder for determining the amount of syllables to write in the translation and how they should align to the melody. Additionally in [16], the evaluation process of the system is assisted by synthesizing the translation via SVS. For PolySinger, we take inspiration from [15] by going for a simple rule-based approach for English to Japanese lyrics translation due to data scarcity and an interest in unraveling the implications of processing Japanese lyrics. To do so, we exploit the pre-trained SOTA model for multi-lingual translation nllb-200 [12] by transferring it to the singing domain.

Early work on SVS created concatenated singing libraries of sampled vocal sounds in a wide range of pitches, from which a synthesizer chose the samples for synthesis based on a musical score [32, 33]. More recent approaches use acoustic models trained on vocal performances from a singer to replicate the way he/she would perform a song given a musical score [7]. Furthermore, cross-lingual synthesis has become possible even when only training on mono-lingual singers [9]. The open-source scene has entered the SVS consumer market, e.g., by use of the ENUNU² plugin to enable usage of the NNSVS toolkit [8] in the OpenUTAU editor. Synthesizer V³ is gathering a common consensus of being one of the best consumer products for SVS with a wide range of high-quality neural singing libraries capable of cross-lingual synthesis in a user-friendly environment with scripting capabilities. Therefore, Synthesizer V is used for SVS in PolySinger. Similarly to [16], we synthesize the translated lyrics, but we want to emphasize that, *differently from [16], PolySinger automates the intermediate link between automatic lyrics translation and SVS.*

3. PROPOSED SINGING-VOICE TO SINGING-VOICE TRANSLATION SYSTEM

Figure 1 illustrates a flowchart of the proposed SV2SVT system. This section will systematically break down the technology, implementation and functionality of each block presented in this figure.

3.1 Automatic Lyrics Transcription

Whisper [22] is a Transformer-based model [10] originally pre-trained on 680k hours of weakly-labeled audio for multi-task learning; there among the main task being multi-lingual automatic speech recognition. The most recent checkpoint, *Whisper-large-v3*, is trained on 1M hours of weakly-labeled audio and 4M hours of audio which was pseudo-labeled by *Whisper-large-v2*. We collect *Whisper-large-v3* from HuggingFace⁴ for automatic lyrics transcription. The model has 1,550M parameters and was trained for 2 epochs on the dataset. We have not fine-tuned Whisper on singing data due to Whisper’s great ability to generalize across several domains. To keep the memory usage of Whisper within ~8 GB, a chunking algorithm segments the vocal performance into 30-second segments which are processed individually with a batch size of 4. Block 1 in Figure 1 is facilitated by Whisper to transcribe an English string of text from an English vocal performance.

3.2 Phoneme-Level Lyrics Alignment and Syllable-Level Lyrics Alignment

In Western languages, poetry and lyrics are very reminiscent of each other. Poetry has a rhythmic structure called

² <https://github.com/oatsu-gh/ENUNU>

³ <https://dreamtonics.com/synthesizerv/>

⁴ <https://huggingface.co/openai/whisper-large-v3>

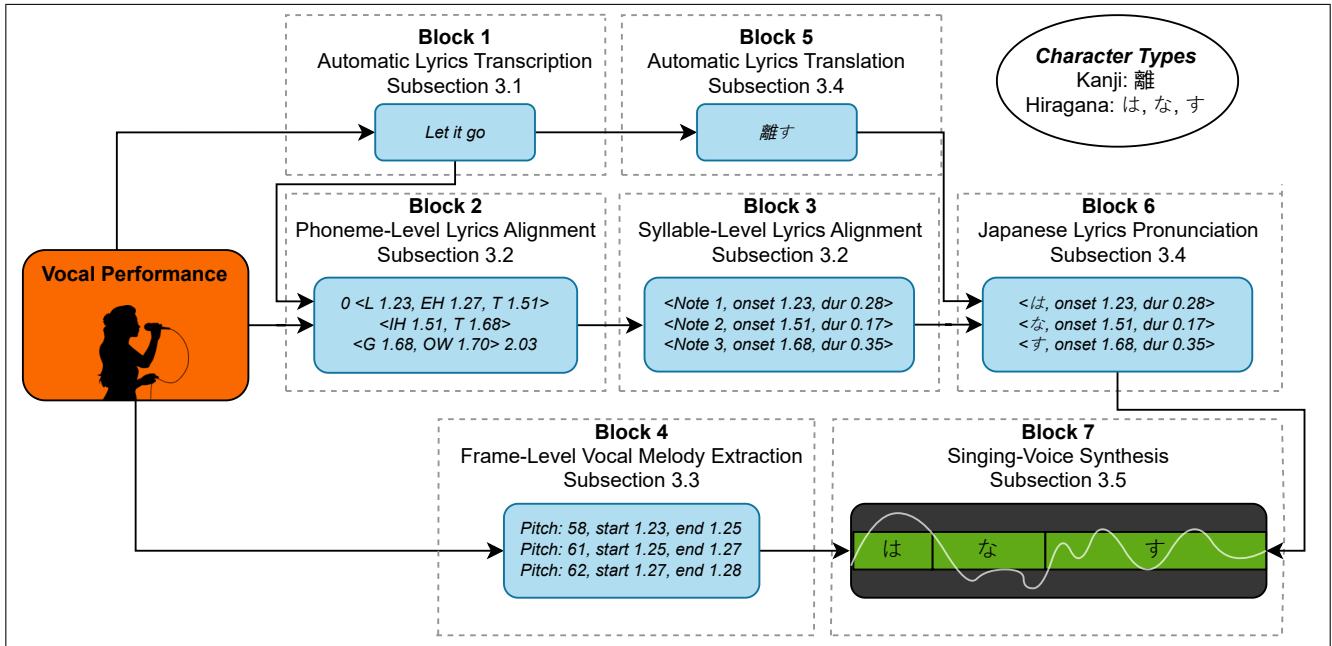


Figure 1. Overview of our proposed SV2SVT system, PolySinger. Provided an English vocal performance, a synthetic vocal performance is created in Block 7, defined by notes with onsets, durations and Japanese lyrics, guided by a frame-level melody. Every numeric value is in seconds and “<>” illustrates the boundaries of notes. The process of segmenting words into syllables is illustrated in Table 1. Fundamentals of Japanese writing are explained in Subsection 3.4, and the process of converting kanji to hiragana is illustrated in Table 2.

meter. This structure can be dissected into a syllabic pattern [34]. Therefore, in this work, we define the onset and duration of notes by aligning the sung syllables to the vocal performance. To obtain syllable-level lyrics alignments, we first align the sequence of phonemes present in the vocal performance. The phoneme sequence is extracted with the pre-trained phoneme-level lyrics aligner informed in [25]. This model is a deep neural network trained for joint phoneme-level lyrics alignment and singing-voice separation. Text and audio are encoded separately. Text features and audio features are aligned by dynamic time warping-attention to minimize the total distance between audio frames and phonemes. The list of possible phonemes is provided by CMUdict⁵. Block 2 in Figure 1 uses the phoneme-level lyrics aligner to align the transcribed lyrics provided by Block 1 to the vocal performance. The phoneme alignments are informed in seconds. The phoneme sequence is concatenated into syllables by simple rules: 1) it is assumed that each phoneme corresponding to a vowel makes an individual syllable, and 2) consonants are merged with their closest neighboring vowel, gravitating towards the rightmost vowel in case of both neighboring phonemes corresponding to vowels. The process of breaking a word into phonemes according to CMUdict and concatenating them into syllables is illustrated in Table 1. In Block 3 of Figure 1, in order to perform syllable-level lyrics alignment, we define the onset of a note as the start of the first phoneme in a syllable, and we define the duration as the time difference between the onset and the end of the last phoneme in the syllable.

⁵ <http://www.speech.cs.cmu.edu/cgi-bin/cmudict>

Word	BLUEBERRY
Phonemes	B L UW1 B EH2 R IY0
Syllables	BLUW BEH RIY

Table 1. Example of the word “blueberry” being deconstructed into phonemes with respect to CMUdict and reconstructed into syllables. An integer ranging 0-2 is associated with each vowel to indicate the type of vowel stress.

3.3 Vocal Melody Extraction

The notes in Block 3 of Figure 1 with timings defined by phoneme boundaries are not individually associated with a unique pitch. Instead, to preserve the melody from the vocal performance as much as possible, the melody is extracted at a frame level. This frame-level pitch contour is used to automate the pitch over time for the note sequence. The notes are set to a standard pitch of 60, and the contour is used to describe the deviation from 60 at each frame. The frame-level VME system presented in [30] deploys a deep convolutional neural network for semantic segmentation across a time-frequency image. Additionally, a progressive neural network is used for cross-domain transfer learning between the audio domain (frequencies) and the symbolic domain (pitch). Block 4 in Figure 1 utilizes this model for extracting the frame-level melody contour as pitches from the vocal performance. The start and end of pitches are in seconds. The model is used through the Omnizart toolkit [31].

3.4 Automatic Lyrics Translation and Japanese Lyrics Pronunciation

nllb-200 [12] is a Transformer [10]-based mixture-of-experts (MoE) multi-lingual translation model that achieves SOTA results in many languages. This success is largely owed to the parallel development of datasets: 1) the expertly-annotated Flores-200 dataset (which consists of 3,001 English sentences translated into 204 languages), and 2) automatically-generated datasets by web-scraping for either mono-lingual sentences with high probability of being each other’s translation, or mono-lingual sentences for back-translation. The biggest model, nllb-200-MoE, has 54B parameters, which is infeasible to run locally. Therefore, we collect the smaller checkpoint nllb-200-distilled-600M from HuggingFace⁶ and fine-tune the model for English to Japanese lyrics translation. Since neither high-quality nor high-quantity dataset of paired English and Japanese lyrics exists, we take inspiration from [15] and scrape the web for lyrics translations that are not necessarily singable. Our dataset consists of ~213k paired lines, thereof ~80% (~20%) being Japanese⇒English (English⇒Japanese)⁷. With PolySinger, we perform English⇒Japanese translation, so we invert the Japanese⇒English lyrics pairs into English⇒Japanese lyrics pairs. We hypothesize that this inversion does not raise an issue, but might in fact incentivize the model to produce high-quality Japanese lyrics even when provided with low-quality English lyrics. Besides, unlike in [15], we attempt fine-tuning with no prior self-supervised training on mono-lingual lyrics due to our larger paired dataset.

Ideally, the output of our fine-tuned model should have the same amount of syllables as established in Block 3 of Figure 1. However, counting syllables is not as simple in Japanese as in English. Japanese has moraic syllabaries in the form of kana. Kana characters have specific pronunciations that take up one mora (Japanese syllable) each. Japanese also uses kanji as logograms, that is, characters that convey a certain meaning. Kanji characters have multiple readings depending on the context, and, as such, it becomes a challenging task to decide the pronunciation of a Japanese sentence. To get the correct pronunciation of kanji characters, pyKAKASI⁸ is used to decode kanji into their hiragana (a type of kana) readings. An illustration of the relation between kanji and Japanese pronunciation can be seen in Table 2. pyKAKASI is dictionary-based, and it can therefore be difficult to convert sentence-wise instead of word-wise. Japanese does not use blank space to separate words, therefore, we use Nagisa⁹, a recurrent neural network trained for Japanese word segmentation.

During inference, a beam search is applied to the output of our fine-tuned nllb-200-distilled-600M with

<i>Kanji character</i>	離
<i>Hiragana readings</i>	り はな
<i>Roman readings</i>	RI HA NA
<i>Mora count</i>	1 2

Table 2. Example of two possible hiragana readings for a kanji character.

as many beams as memory will allow (~50 beams in our tests). The beams are biased towards a token count lower than the number of syllables in Block 3 of Figure 1 due to a kanji always corresponding to at least one syllable. Each generated sentence becomes word-separated with Nagisa and the kanji are converted into hiragana readings with pyKAKASI. The sentence with the lowest non-negative difference between mora count and syllable count gets selected and assigned to the notes in Block 6 of Figure 1.

3.5 Singing-Voice Synthesis

Synthesizer V is a SVS system with growing popularity among musicians. The technology behind Synthesizer V is kept proprietary. Based on related literature [7, 8], it is assumed that AI singing-voices in Synthesizer V are acoustic models trained on phoneme-level annotated vocal performances. With this training scheme, the model recognizes patterns in a singer’s vocal performances, e.g., articulation of phoneme sequences, transitions between pitches and tendencies to use vibrato. Additionally, Synthesizer V AI voices have parameters for vocal modes, which can be included in training by annotating vocal samples with a singing style, e.g., nasal, powerful, soft, and whisper. AI singing-voices are usually only trained on vocal performances by a mono-lingual or bilingual singer, however AI voices in Synthesizer V are capable of cross-lingual synthesis in English, Japanese, Mandarin Chinese, Cantonese, and, recently, Spanish. It is assumed, based on related literature [9], that cross-lingual synthesis is achieved by unifying phoneme representations across languages with the international phonetic alphabet and training on data labeled with language identification such that the acoustic model can learn language-specific features. As illustrated in Figure 1, the notes with Japanese lyrics provided by Block 6 are plotted into Synthesizer V at a standard pitch of 60. The vocal contour provided by Block 4 is used to automate the deviation from pitch 60 over time. We use the AI singing-voice Mai in Synthesizer V to generate the Japanese vocal performance.

4. EXPERIMENTS

Objective measures in machine translation such as BLEU [35] are typically used for word-wise similarity with respect to a ground truth. Such a method does not suit lyrics translation as there should rather be a focus on semantic interpretation rather than precise word choice. Moreover, PolySinger has to be evaluated on the overall per-

⁶ <https://huggingface.co/facebook/nllb-200-distilled-600M>

⁷ Both English⇒Japanese and Japanese⇒English are collected from <https://lyricstranslate.com/>. Extra Japanese⇒English is also collected from <https://www.animelyrics.com/>.

⁸ <https://codeberg.org/miurahr/pykakasi>

⁹ <https://github.com/taishi-i/nagisa>

Score	1	2	3	4	5
Meaning	Very poor	Poor	Neutral	Good	Very good

Table 3. Five-point scale for MOS test.

ID	Question
Q1	How much sense do the lyrics make?
Q2	How natural is the Japanese used in the lyrics?
Q3	How well is the meaning of the original lyrics preserved?
Q4	How singable are the generated lyrics?
Q5	How well are the lyrics and melody aligned?
Q6	What is the overall quality of the generated Japanese singing?

Table 4. Questions asked to the test subjects in the MOS test of Section 4.

formance achieved for English⇒Japanese SV2SVT rather than solely on the translation quality. Therefore, we evaluate PolySinger subjectively by means of a MOS test.

4.1 Methodology

Six native Japanese speakers participated in a MOS test to evaluate the perceptual quality of English⇒Japanese SV2SVT using PolySinger on 5 different vocal performances. All test subjects were females ranging from 24 to 39 years old with no hearing impairment. The test subjects were asked to self-report their English speaking level. Two participants reported complete fluency (5/5), one reported near fluency (4/5), two more indicated advanced comprehension (3/5), and the final one reported intermediate comprehension (2/5). Using the inference procedure described in Subsection 3.4, PolySinger was alternately tested with the original `nllb-200-distilled-600M` (*Baseline*) and our fine-tuned `nllb-200-distilled-600M` (*Fine-tuned*) on every vocal performance. The test subjects were asked to first listen to an English vocal performance, followed by the synthetic performances generated by the two PolySinger versions (i.e., Baseline and Fine-tuned). Participants were not informed which synthetic vocal performance was generated by which system variant. Using the 5-point scale shown in Table 3, the test subjects were asked to assess each generated performance by the 6 MOS questions displayed in Table 4. The average time a participant spent on the evaluation was 53 min. The audio samples used for evaluation can be accessed here¹⁰.

After the participants submitted their MOS scores, we additionally had a brief discussion with them individually about their general opinions and observations.

4.2 Results

Table 5 shows the MOS test results along with 95% confidence intervals from the Student’s *t*-distribution [36]. Both system variants (i.e., Baseline and Fine-tuned) lie somewhere between poor and neutral in all 6 MOS questions Q1–Q6. The relatively large confidence intervals in Table

5 suggest a high variance in opinion scores. We investigate this variance in Figure 2 by representing per-question score’s relative frequency. While it is true that the majority of opinion scores lies in the mid-to-low end of the spectrum, several evaluations have also resulted in good or very good opinion scores. This emphasizes the very subjective nature of the SV2SVT problem.

Given a MOS question Q1–Q6, we determine if there is a statistically significant difference between the opinion scores for Baseline and Fine-tuned. A Kolmogorov-Smirnov test [37] generally rejects, at a standard significance level of 5%, the null hypothesis that our opinion score sample populations follow Gaussian distributions. Therefore, we use a Wilcoxon rank-sum test [38] to determine whether there are statistically significant differences in MOS between the two system variants. The *p*-values shown in Table 6 demonstrate that the performance of the two systems is rather equivalent. Specifically, these *p*-values indicate that there are no statistically significant differences between Baseline and Fine-tuned at a standard significance level of 5% given any of the 6 MOS questions.

During discussions conducted after the MOS test, the test subjects generally conveyed a positive reaction towards SV2SVT being possible with PolySinger. However, as anticipated, the participants mainly assessed PolySinger by the naturalness of the Japanese language used in the context of singing and the pronunciation of words. The most recurring observations from the participants, that were suggested as crucial improvements needed for the pursuit of natural Japanese singing, are summarized in Table 7. In the next section, we will discuss the comments in Table 7 as to why our processing and synthesis of Japanese might not have been of ideal quality, along with our plan for improving them in future. Moreover, the statistically insignificant difference between Baseline and Fine-tuned is also discussed along with techniques and technologies that may assist in improving PolySinger.

5. DISCUSSION

To produce natural Japanese speech synthesis, the front-end of a text-to-speech system requires phonetic and prosodic features [39]. Phonetic features, i.e., pronunciations, are typically acquired by grapheme-to-phoneme (G2P) conversion, and prosodic features, i.e., rhythm and intonation, are in Japanese typically acquired by phrase break prediction and accent estimation [40–42]. G2P conversion is particularly difficult in Japanese, since kanji characters can have multiple pronunciations. As indicated by our test subjects (C1 in Table 7) and discussed in [40], the accuracy obtained by dictionary-based G2P conversion in Japanese is not satisfactory. Japanese has no word separators, which also makes it difficult to determine phrase breaks. In our work, we performed word segmentation with `Nagisa` to avoid intra-word breaking, and attempted to define phrase breaks as the pauses transcribed by lyrics alignment on an English vocal performance. However, according to our test subjects (C5 in Table 7), these methods yielded limited success. As future work, we will inves-

¹⁰ <https://antonisen.dev/polysinger/>

System / Question	Q1	Q2	Q3	Q4	Q5	Q6
Baseline	2.53 ± 0.49	2.57 ± 0.48	2.47 ± 0.44	2.40 ± 0.41	2.50 ± 0.52	2.33 ± 0.45
Fine-tuned	2.17 ± 0.46	2.30 ± 0.48	2.10 ± 0.44	2.23 ± 0.44	2.10 ± 0.40	2.13 ± 0.41

Table 5. MOS quality test results, broken down by question, with 95% confidence intervals.

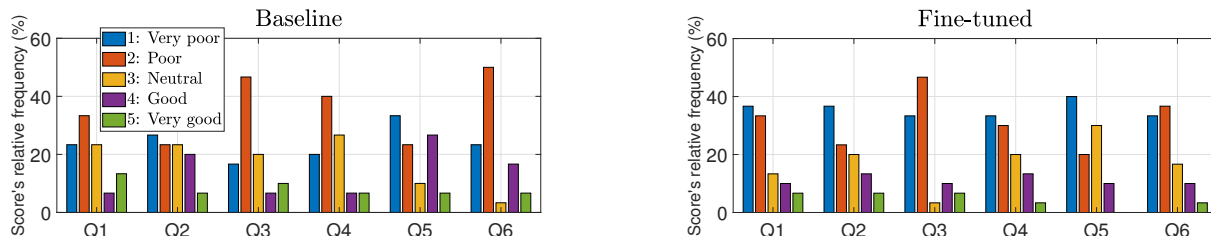


Figure 2. Bar plots representing per-question score’s relative frequency from the MOS quality test for Baseline (left) and Fine-tuned (right).

Question	Q1	Q2	Q3	Q4	Q5	Q6
<i>p</i> -value	0.228	0.389	0.115	0.509	0.279	0.557

Table 6. *p*-values, broken down by question, from a Wilcoxon rank-sum test comparing MOS scores from Baseline and Fine-tuned.

ID	Comment
C1	Incorrect readings of kanji
C2	Usage of keigo in casual language
C3	Direct translations where interpretations are needed
C4	Occasionally, lyrics are not entirely translated
C5	Both intra- and inter-word separation at unnatural places
C6	Missing keywords important to the song
C7	Wrong word order
C8	Improper mixture of feminine and masculine language

Table 7. Comments from discussions with the test subjects on essential improvements that could lead to more natural synthetic Japanese singing.

tigate the adaptation of SOTA methodologies in Japanese text-to-speech to SV2SVT such as phrase break prediction with large language models (LLMs) [41] and G2P conversion via machine translation [40].

In [15], they demonstrated an improvement in automatic lyrics translation by fine-tuning on paired lyrics that were not necessarily singable, but also by pre-training on mono-lingual lyrics. In this work, we avoided pre-training on mono-lingual lyrics and only fine-tuned on paired lyrics that were not necessarily singable, which resulted in no statistically significant improvement with respect to the baseline model (see Table 6). We applied a beam search to find translated lyrics that fit well into the syllable count of the original lyrics. The selected lyrics were occasionally not a full translation of the original lyrics (C4 in Table 7). Apart from the use of keigo (honorific language) being inappropriate for the inherent casual nature of song lyrics (C2 in Table 7), we conjecture that keigo could also be a major

cause of incomplete lyrics translations. This is because keigo will usually incorporate more characters than casual language, which means that it will be harder to fit the lyrics into the fixed syllable count.

In [16], they achieve SOTA results by training on a dataset created by back-translating mono-lingual lyrics and automatically aligning automatically-generated melodies that fit both the source and target lyrics. As future work, creating such a dataset and training an alignment decoder similarly to [16] could very well be adapted to Japanese. However, we hypothesize that translation systems have an inherent limitation towards cross-lingual songwriting that hinders them from rivaling professional human translators due to a lack of abstract interpretation and “imagination”. Hence, as future work, we will also investigate the usage of LLMs for sentiment analysis and feature extraction to exploit poetry/lyrics generation models. By lyrics generation, guided by keyword spotting, we can also address the issue of missing keywords (C6 in Table 7).

6. CONCLUSION

The goal of this paper has been to adapt conventional S2ST to the singing domain. To do so, we have built the first SV2SVT system, PolySinger, by cascading SOTA MIR technologies facilitating a modular tool for extended research in SV2SVT. We have conducted a MOS test with native Japanese speakers to evaluate PolySinger’s performance for English to Japanese SV2SVT. Results indicate that we have created a fundamentally-coherent structure for SV2SVT, but the translation of English lyrics into Japanese and the automatic synthesis of it is not yet natural enough. To further develop SV2SVT, our future work will investigate—to facilitate creative lyrics generation—the usage of sentiment analysis and feature extraction for abstract meaning representation of lyrics as opposed to translation. Finally, we will also investigate the necessities for autonomous generation of natural Japanese lyrics.

7. ACKNOWLEDGMENTS

This work has been funded by the Spanish Ministry of Science and Innovation under the “Ramón y Cajal” programme (RYC2022-036755-I). In addition, we want to express our heartfelt appreciation to the test subjects who voluntarily contributed to this study by participating in the perceptual quality test.

8. REFERENCES

- [1] A. Lavie, A. Waibel, L. Levin, M. Finke, D. Gates, M. Gavalda, T. Zeppenfeld, and P. Zhan, “Janus-III: speech-to-speech translation in multiple languages,” in *Proc. of the 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 1997, pp. 99–102.
- [2] S. Nakamura, K. Markov, H. Nakaiwa, G. Kikui, H. Kawai, T. Jitsuhiro, J.-S. Zhang, H. Yamamoto, E. Sumita, and S. Yamamoto, “The ATR multilingual speech-to-speech translation system,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, pp. 365–376, 2006.
- [3] International Telecommunication Union, “Functional requirements for network-based speech-to-speech translation services,” *ITU-T F.745*, 2016.
- [4] Y. Jia, R. J. Weiss, F. Biadys, W. Macherey, M. Johnson, Z. Chen, and Y. Wu, “Direct speech-to-speech translation with a sequence-to-sequence model,” in *Proc. of Interspeech 2019*, 2019.
- [5] X. Li, Y. Jia, and C.-C. Chiu, “Textless direct speech-to-speech translation with discrete speech representation,” in *Proc. of the 2023 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2023.
- [6] Seamless Communication, L. Barrault, Y.-A. Chung, M. C. Meglioli, D. Dale, N. Dong, P.-A. Duquenne, H. Elsahar, H. Gong, K. Heffernan, J. Hoffman, C. Klaiber, P. Li, D. Licht, J. Maillard, A. Rakotoarison, K. R. Sadagopan, G. Wenzek, E. Ye, B. Akula, P.-J. Chen, N. E. Hachem, B. Ellis, G. M. Gonzalez, J. Haaheim, P. Hansanti, R. Howes, B. Huang, M.-J. Hwang, H. Inaguma, S. Jain, E. Kalbassi, A. Kallet, I. Kulikov, J. Lam, D. Li, X. Ma, R. Mavlyutov, B. Peloquin, M. Ramadan, A. Ramakrishnan, A. Sun, K. Tran, T. Tran, I. Tufanov, V. Vogeti, C. Wood, Y. Yang, B. Yu, P. Andrews, C. Balioglu, M. R. Costa-jussà, O. Celebi, M. Elbayad, C. Gao, F. Guzmán, J. Kao, A. Lee, A. Mourachko, J. Pino, S. Popuri, C. Ropers, S. Saleem, H. Schwenk, P. Tomasello, C. Wang, J. Wang, and S. Wang, “SeamlessM4T: Massively multilingual & multimodal machine translation,” *arXiv*, vol. abs/2308.11596, 2023.
- [7] J. Liu, C. Li, Y. Ren, F. Chen, and Z. Zhao, “Diff-Singer: Singing voice synthesis via shallow diffusion mechanism,” *Proc. of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 10, pp. 11 020–11 028, 2022.
- [8] R. Yamamoto, R. Yoneyama, and T. Toda, “NNSVS: A neural network-based singing voice synthesis toolkit,” in *Proc. of the 2023 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2023.
- [9] X. Wang, C. Zeng, J. Chen, and C. Wang, “CrossSinger: A cross-lingual multi-singer high-fidelity singing voice synthesizer trained on monolingual singers,” in *Proc. of the 2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2023.
- [10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in Neural Information Processing Systems*, vol. 30, 2017.
- [11] Y. Tang, C. Tran, X. Li, P.-J. Chen, N. Goyal, V. Chaudhary, J. Gu, and A. Fan, “Multilingual translation with extensible multilingual pretraining and finetuning,” *arXiv*, vol. abs/2008.00401, 2020.
- [12] NLLB Team, M. R. Costa-jussà, J. Cross, O. Çelebi, M. Elbayad, K. Heffernan, K. Heffernan, E. Kalbassi, J. Lam, D. Licht, J. Maillard, A. Sun, S. Wang, G. Wenzek, A. Youngblood, B. Akula, L. Barrault, G. M. Gonzalez, P. Hansanti, J. Hoffman, S. Jarrett, K. R. Sadagopan, D. Rowe, S. L. Spruit, C. Tran, P. Y. Andrews, N. F. Ayan, S. Bhosale, S. Edunov, A. Fan, C. Gao, V. Goswami, F. Guzmán, P. Koehn, A. Mourachko, C. Ropers, S. Saleem, H. Schwenk, and J. Wang, “No language left behind: Scaling human-centered machine translation,” *arXiv*, vol. abs/2207.04672, 2022.
- [13] J. Franzon, “Three dimensions of singability. An approach to subtitled and sung translations,” in *Text and Tune. On the Association of Music and Lyrics in Sung Verse*. Peter Lang, 2015, pp. 333–346.
- [14] M. Ghazvininejad, Y. Choi, and K. Knight, “Neural poetry translation,” in *Proc. of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 2018, pp. 67–71.
- [15] F. Guo, C. Zhang, Z. Zhang, Q. He, K. Zhang, J. Xie, and J. Boyd-Graber, “Automatic song translation for tonal languages,” in *Findings of the Association for Computational Linguistics: ACL 2022*, 2022, pp. 729–743.
- [16] C. Li, K. Fan, J. Bu, B. Chen, Z. Huang, and Z. Yu, “Translate the beauty in songs: Jointly learning to align melody and translate lyrics,” in *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023, pp. 27–39.

- [17] J.-Y. Hsu and L. Su, “VOCANO: A note transcription framework for singing voice in polyphonic music,” in *ISMIR 2021: Proceedings of the 22nd International Society for Music Information Retrieval Conference*, 2021, pp. 293–300.
- [18] E. Demirel, S. Ahlbäck, and S. Dixon, “Automatic lyrics transcription using dilated convolutional neural networks with self-attention,” in *Proc. of the 2020 International Joint Conference on Neural Networks (IJCNN)*, 2020.
- [19] E. Demirel, S. Ahlbäck, and S. Dixon, “MSTRE-Net: Multistreaming acoustic modeling for automatic lyrics transcription,” in *Proc. of ISMIR 2021, International Society for Music Information Retrieval*, 2021.
- [20] A. Baeovski, Y. Zhou, A. Mohamed, and M. Auli, “wav2vec 2.0: A framework for self-supervised learning of speech representations,” in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 12 449–12 460.
- [21] L. Ou, X. Gu, and Y. Wang, “Transfer learning of wav2vec 2.0 for automatic lyric transcription,” in *Proc. of ISMIR 2022, International Society for Music Information Retrieval*, 2022.
- [22] A. Radford, J. Wook Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, “Robust speech recognition via large-scale weak supervision,” in *Proc. of the 40th International Conference on Machine Learning: ICML 2023, Hawaii (USA)*, 2023, pp. 28 492–28 518.
- [23] A. M. Kruspe, “Bootstrapping a system for phoneme recognition and keyword spotting in unaccompanied singing,” in *Proc. of ISMIR 2016, International Society for Music Information Retrieval*, 2016.
- [24] C. Gupta, E. Yilmaz, and H. Li, “Automatic lyrics alignment and transcription in polyphonic music: Does background music help?” in *Proc. of ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2020, pp. 496–500.
- [25] K. Schulze-Forster, C. S. J. Doire, G. Richard, and R. Badeau, “Phoneme level lyrics alignment and text-informed singing voice separation,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 2382–2395, 2021.
- [26] J. Huang, E. Benetos, and S. Ewert, “Improving lyrics alignment through joint pitch detection,” in *Proc. of ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2022, pp. 451–455.
- [27] S. Durand, D. Stoller, and S. Ewert, “Contrastive learning-based audio to lyrics alignment for multiple languages,” in *Proc. of ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2023.
- [28] R. M. Bittner, J. Salamon, S. Essid, and J. P. Bello, “Melody extraction by contour classification,” in *Proc. of the International Conference on Music Information Retrieval (ISMIR)*, 2015.
- [29] S. Kum, C. Oh, and J. Nam, “Melody extraction on vocal segments using multi-column deep neural networks,” in *ISMIR 2016: Proceedings of the 17th International Society for Music Information Retrieval Conference*, 2016, pp. 819–825.
- [30] W.-T. Lu and L. Su, “Vocal melody extraction with semantic segmentation and audio-symbolic domain transfer learning,” in *ISMIR 2018: Proceedings of the 19th International Society for Music Information Retrieval Conference*, 2018, pp. 521–528.
- [31] Y.-T. Wu, Y.-J. Luo, T.-P. Chen, I.-C. Wei, J.-Y. Hsu, Y.-C. Chuang, and L. Su, “Omnizart: A general toolbox for automatic music transcription,” *Journal of Open Source Software*, vol. 6, no. 68, p. 3391, 2021.
- [32] M. Macon, L. Jensen-Link, E. B. George, J. Olive-rio, and M. Clements, “Concatenation-based MIDI-to-singing voice synthesis,” in *Proc. of the Audio Engineering Society Convention 103*, 1997.
- [33] H. Kenmochi and H. Ohshita, “VOCALOID – commercial singing synthesizer based on sample concatenation,” in *Proc. of Interspeech 2007*, 2007.
- [34] E. Greene, T. Bodrumlu, and K. Knight, “Automatic analysis of rhythmic poetry with applications to generation and translation,” in *Proc. of the 2010 Conference on Empirical Methods in Natural Language Processing*, 2010, pp. 524–533.
- [35] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “BLEU: a method for automatic evaluation of machine translation,” in *Proc. of the 40th Annual Meeting of the Association for Computational Linguistics*, 2002, pp. 311–318.
- [36] N. Blachman and R. Machol, “Confidence intervals based on one or more observations,” *IEEE Transactions on Information Theory*, vol. 33, no. 3, pp. 373–382, 1987.
- [37] F. J. Massey Jr, “The Kolmogorov-Smirnov test for goodness of fit,” *Journal of the American Statistical Association*, vol. 46, no. 253, pp. 68–78, 1951.
- [38] F. Wilcoxon, “Individual comparisons by ranking methods,” *Biometrics Bulletin*, vol. 1, no. 6, pp. 80–83, 1945.
- [39] T. Fujimoto, K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda, “Impacts of input linguistic feature representation on Japanese end-to-end speech synthesis,”

in *Proc. of the 10th ISCA Speech Synthesis Workshop. ISCA, Vienna, Austria*, 2019.

- [40] N. Kakegawa, S. Hara, M. Abe, and Y. Ijima, “Phonetic and prosodic information estimation from texts for genuine Japanese end-to-end text-to-speech,” in *Proc. of Interspeech*, 2021, pp. 126–130.
- [41] K. Futamata, B. Park, R. Yamamoto, and K. Tachibana, “Phrase break prediction with bidirectional encoder representations in Japanese text-to-speech synthesis,” in *Proc. of Interspeech 2021*, 2021.
- [42] K. Kurihara, N. Seiyama, and T. Kumano, “Prosodic features control by symbols as input of sequence-to-sequence acoustic modeling for neural TTS,” *IEICE Transactions on Information and Systems*, vol. 104, no. 2, pp. 302–311, 2021.

ON THE VALIDITY OF EMPLOYING CHATGPT FOR DISTANT READING OF MUSIC SIMILARITY

Arthur Flexer

Institute of Computational Perception
Johannes Kepler University Linz, Austria
arthur.flexer@jku.at

ABSTRACT

In this work we explore whether large language models (LLM) can be a useful and valid tool for music knowledge discovery. LLMs offer an interface to enormous quantities of text and hence can be seen as a new tool for 'distant reading', i.e. the computational analysis of text including sources about music. More specifically we investigated whether ratings of music similarity, as measured via human listening tests, can be recovered from textual data by using ChatGPT. We examined the inferences that can be drawn from these experiments through the formal lens of validity. We showed that correlation of ChatGPT with human raters is of moderate positive size but also lower than the average human inter-rater agreement. By evaluating a number of threats to validity and conducting additional experiments with ChatGPT, we were able to show that especially construct validity of such an approach is seriously compromised. The opaque black box nature of ChatGPT makes it close to impossible to judge the experiment's construct validity, i.e. the relationship between what is meant to be inferred from the experiment, which are estimates of music similarity, and what is actually being measured. As a consequence the use of LLMs for music knowledge discovery cannot be recommended.

1. INTRODUCTION

When developing and validating hypotheses in musicology, relevant information very often is obtained from written documents. This information from collections, anthologies, compilations, biographies, reviews, journals, etc is today often available in digitized formats, enabling usage of methods from natural language processing (NLP) for music knowledge discovery [1]. In the humanities such an approach is also known as 'distant reading' [2, 3], i.e. the computational analysis of large quantities of books and texts which cannot be handled by individual scholars in what is known as traditional 'close reading', i.e. very careful and detailed expert reading of only comparably few

texts. Large language models (LLM) [4–6] offer a convenient interface to enormous quantities of text and hence can be seen as a new tool for distant reading. In our previous work [7] we have evaluated the use of LLMs for distant reading of music similarity. Our results showed that music similarity, as measured via human listening tests, can to a certain degree be recovered from textual data by using ChatGPT as a distant reading tool. However, it also already became clear that the black box nature of LLMs, and especially of ChatGPT, presents a problem for the validity of such an approach.

In this article we therefore critically appraise our own previous work by utilizing an established framework of validity by Shadish et al. [8]. Validity is the truth of an inference made from evidence gathered through an experiment and as such an integral pillar of working scientifically. We will question our approach to music knowledge discovery concerning its statistical conclusion, internal, construct and external validity. All four types of validity have recently been discussed by applying Shadish et al. [8] to the context of music information research [9], which we will use as a guideline in this article. Reformulating our work in the general framework of validity will allow us to draw conclusions going beyond our particular music similarity setting to the general problem of using LLMs for music knowledge discovery.

We present related work in section 2 and explain the experimental setting (including preceding work we build on) in section 3. In sections 4 to 7 we critically appraise a primary study on human perception of music similarity [10], our own previous work with ChatGPT [7], as well as a number of additional ChatGPT experiments conducted for this work, all concerning four types of validity. We discuss our main findings and conclude in section 8.

2. RELATED WORK

Large language models (LLM) are deep neural models that learn representations of text by trying to predict the next word given a textual context. State of the art approaches are based on transformer architectures [4, 5], implementing an attention mechanism which learns to reweight parts of the textual input in relation to its importance for the task under consideration. ChatGPT (<https://openai.com/blog/chatgpt>) is a chatbot imitating a human conversational partner and is also based on



© A. Flexer. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** A. Flexer, "On the validity of employing ChatGPT for distant reading of music similarity", in *Proc. of the 25th Int. Society for Music Information Retrieval Conf.*, San Francisco, United States, 2024.

a ‘Generative Pre-trained Transformer’ (GPT). We have used GPT-3.5, which itself is a fine-tuned version of GPT-3 [5], for our experiments in this paper and in our previous work [7]. A problem common to all members of the GPT family (including ChatGPT) is that exact details of models, training sets, parameters, etc are not known. A non peer reviewed report [6] by the developing team about the latest version (GPT-4) even states that "[...] no further details about the architecture (including model size), hardware, training compute, dataset construction, training method, or similar" can be given due to "safety implications" and the "competitive landscape" of LLM research.

ChatGPT has already been used in a music context rating instrument sounds on a set of 20 semantic scales [11]. It was found that ChatGPT’s answers are only partially correlated with human ratings, with Pearson correlations above 0.80 only achieved for clearly defined dimensions of musical sounds such as brightness (bright–dark) and pitch height (deep–high). This is closely related to another approach trying to extract psychophysical information from text by aligning GPT-4 results with human auditory experience [12]. Further applications of LLMs to music include lyrics summarization [13] and usage as ranking models for music recommendation [14]. Applying LLMs to music data is also reminiscent of preceding approaches computing music similarity from textual sources, including web-based data [15], semantic music tags [16] or lyrics [17]. There also exists related work in the text domain, e.g. on using LLMs for evaluation of jokes [18, 19].

Previous work on distant reading in music information research (MIR) includes automatic band member detection and automatic recognition of all their released records from internet text sources [20], sentiment analysis of a large corpus of Pop music reviews [1], discovery of social and professional networks from Wikipedia articles on Renaissance musicians [21], extraction of semantic information from an online discussion forum on Carnatic music [22], or very detailed cross-linking of references to musical passages in musicological texts [23].

3. EXPERIMENTAL SETUP

In our previous research [7] we explored whether ChatGPT can be used to ‘distant read’ the similarity between songs and compared the results to a study employing human listening tests on the same pairs of songs [10]. ChatGPT therefore has to recover music similarity, as judged by humans listening to audio, solely from textual data. Textual sources could also provide complementary information like cultural connotations, or other forms of so-called music context [24]. Such information is of course not present in music audio alone, but the mere knowledge of such contextual facts may nevertheless influence human listeners in their judgement. Our major hypothesis therefore was:

"Music similarity estimated with ChatGPT correlates positively with human perception of music similarity"

In order to being able to properly discuss the validity of conclusions drawn from the respective experiment, we must identify its components. *Treatments* are the things applied to *units* in order to cause an effect. In our case the participants and ChatGPT are the treatments, while the set of questions (pairs of songs to be evaluated) are the units. The effect we want to cause is to gain an estimate of music similarity. *Observations* are what is measured on a unit, in our case the music similarity ratings ranging from 0 to 100.

3.1 Human evaluation of music similarity

The primary study conducted a series of listening tests with human participants [10], with the age of participants ranging from 26 to 34 years with an average of 28.2 (three females and three males, called graders S1 to S6 from here on). The 5×18 songs belonged to five genres (for a full list see section A of the appendix of the original article (<https://doi.org/10.5334/tismir.107.s1>)): (i) **American Soul** from the 1960s and 1970s with only male singers singing; (ii) **Bebop**, the main jazz style of the 1940s and 1950s, with excerpts containing trumpet, saxophone and piano parts; (iii) **High Energy** (Hi-NRG) dance music from the 1980s, typically with continuous eighth note bass lines, aggressive synthesizer sounds and staccato rhythms; (iv) **Power Pop**, a Rock style from the 1970s and 1980s, with chosen songs being guitar-heavy and with male singers; (v) **Rocksteady**, which is a precursor of Reggae with a somewhat soulful basis. All songs had limited popularity with under 50.000 accesses on Spotify at the time of the study. The authors validated genres via respective Wikipedia artist pages as well as by listening to all songs. Fifteen seconds of a representative part of every song (usually the refrain) were presented in the listening tests and participants were asked to:

"assess the similarity between the query song and each of the five candidate songs by adjusting the slider" (ranging from 0 to 100 %) and "to answer intuitively since there are no wrong answers"

Based on randomly chosen 15 query songs, comparisons of five pairs had to be made for every query group yielding a total of $15 \times 5 = 75$ pairs, with every song appearing exactly once in the whole questionnaire (15 as query songs, 75 as candidate songs).

3.2 ChatGPT evaluation of music similarity

For our initial experiments [7] conducted on the 5th and 6th of April 2023 we used the "Free Research Preview" of the ChatGPT Mar 23 Version (<https://openai.com/blog/chatgpt>). The service came with a warning that "ChatGPT may produce inaccurate information about people, places, or facts" and the information that "ChatGPT is fine-tuned from GPT-3.5, a language model trained to produce text. ChatGPT was optimized for dialogue by using Reinforcement Learning with Human Feedback (RLHF) – a method that uses human demonstrations

and preference comparisons to guide the model toward desired behavior".

We asked ChatGPT the following question for the exact same $15 \times 5 = 75$ song pairs as used in the human listening test:

"On a scale of 1 to 100, how similar is the song [s_i] by [artist_A] to the song [s_j] by [artist_B]?"

Interestingly, ChatGPT sometimes needed persuasion to provide an answer at all, stating e.g. that "As an AI language model, I do not have the ability to directly listen to music or interpret subjective qualities such as similarity between songs", or that any answer would be "merely speculation". The following additional input sentences (in that order) provided by us in ensuing dialogues always resulted in ChatGPT providing a similarity score:

1. "Please just make a guess based on the information you have already"
2. "Please try anyway"
3. "Then please just speculate"

Such additional persuasion was necessary for 8 out of 75 questions, mostly at the beginning of ChatGPT sessions. Experiments had to be split over three separate sessions due to restriction of the free ChatGPT version.

4. STATISTICAL CONCLUSION VALIDITY

Statistical conclusion validity is "the validity of inferences about covariation between two variables" [8]. Here the main concern is with *statistical significance*, i.e., that an observed covariation between treatment and effect is not likely to arise by chance.

In accordance with the initial study [10], we recorded the music similarity ratings and then, to gain an estimate of the level of agreement between human participants and ChatGPT, we analysed degrees of inter-rater agreement. Specifically, we computed the Pearson correlations ρ_{listen} between graders S1 to S6 as well as ρ_{gpt} between graders S1 to S6 and ChatGPT for the 75 pairs of query/candidate songs (see table 1 for an overview of all results). The human listening test had been conducted twice at time points t1 and t2 with a two week time lag [10]. The 15 plus 15 correlations ρ_{listen} (t1 and t2) between the six graders range from 0.59 to 0.86, with an average of 0.74. The 6 plus 6 correlations ρ_{gpt} (t1 and t2) between the six graders and ChatGPT are considerably lower, with a range from 0.39 to 0.72 and an average of 0.58. The correlation ρ_{gpt} is statistically significant, i.e. the probability that we observe such a positive correlation by chance is basically zero ($t(898) = -17.83$, $p=0.00$). Hence, a valid statistical conclusion is that we observe a significant covariation between the human participants and ChatGPT in the observed estimates of music similarity. This result therefore seems to corroborate our hypothesis that music similarity estimated with ChatGPT correlates positively with human perception

	five genres		one genre
agreement	inter	intra	inter
ρ_{listen}	0.74	0.80	0.24
ρ_{gpt}	0.58	0.68	0.06

Table 1. Overview of results for five and one genre experiments. Shown are levels of inter- and intra-rater agreement between human participants (ρ_{listen}) and between human participants and ChatGPT (ρ_{gpt}).

of music similarity. In addition, the differences in correlation between ρ_{listen} and ρ_{gpt} are also statistically significant ($t(40)=6.05$, $p=0.00$).

5. INTERNAL VALIDITY

Internal validity is "the validity of inferences about whether the observed covariation between two variables is causal" [8]. It is therefore focused on the *cause* of a particular response to the treatment, going beyond statements concerning only the strength of covariation. A typical threat to internal validity is *confounding*, which is the confusion of the treatment with other factors, often arising from poor operationalisation in an experiment.

For our specific experiment we are interested in estimates of music similarity, either via listening tests with humans or from text sources via ChatGPT. One explanation for the observed level of rater agreement ρ_{gpt} is that indeed human perception of audio music similarity is positively correlated with ChatGPT estimates of music similarity. This is certainly one explanation consistent with our observations, but is it the only one? The internal validity of this conclusion relies on the key assumption that the observed positive correlation can *only* be explained in terms of music similarity and that there is no other way to arrive at the observations.

However, already in the initial study [10], participants commented that the genre of the songs was an important factor when evaluating the similarity of songs. When both query and candidate songs belonged to the same genre, similarity ratings were on average higher (within genres: 43.09) when compared to song pairs from different genres (between genres: 30.10). For our initial ChatGPT experiments [7] we have observed something related in the explanations provided by ChatGPT together with the similarity ratings. We provide some exemplary ChatGPT explanations with different levels of detail in table 2. As is typical for the answers ChatGPT provided, genre, instrumentation or era of recording are being discussed. A standard definition [25] of music genre states that it is "a set of musical events ... governed by a definite set of socially accepted rules", with musical events being "any type of activity performed around any type of event involving sound". Sound events are of course also linked to the concept of music similarity, making it clear that music genre and similarity are related but not synonymous concepts.

We therefore repeat the analysis of similarity ratings re-

"These two songs are from very different genres and have distinct musical styles."
"[...] I can attempt to speculate based on the artist's genre and the era of the music"
"They are from different musical genres, different eras, and have different rhythms, melodies, instrumentation, and lyrics."
"Both songs share some similarities in terms of their musical genres, but they are likely to have different arrangements, melodies, and lyrics."
"While both songs are in the broad category of popular music, they come from different genres (soul/R&B for Major Harris and reggae for The Heptones) and have different rhythms, melodies, instrumentation, and lyrics."
"[...] given that both artists were active in the same time period and were part of the Jamaican music scene, it is possible that there may be some similarities in terms of instrumentation, rhythm, or vocal style"

Table 2. Typical explanations provided by ChatGPT.

garding genres of query/candidate songs and get the following results: within genres: 43.13, and between genres: 13.42. Just as for human ratings, ChatGPT ratings seem to rely at least in part on genre information and not music similarity alone. This then calls into question how the design of our experiment relates to what we actually want to measure, which is music similarity. This is where construct validity becomes relevant.

6. CONSTRUCT VALIDITY

Construct validity is “the validity of inferences about the higher order constructs that represent sampling particulars” [8]. This concerns the operationalisation of the experimentalist’s intention, i.e. the relationship between what is meant to be inferred from an experiment and what is actually measured. In our case the higher order construct is music similarity.

An important part of the operationalisation of our experiment is the exact form of questions the participants (“assess the similarity between the query song”) and ChatGPT (“how similar is the song”) are being asked. Both questions clearly aim at the similarity between songs but do not specify what exact aspect of similarity is meant. Many possibilities come to mind, e.g. similar in terms of melody, tempo, instrumentation, time of publishing, or maybe genre? In-

deed, as has already been explained above, human participants commented that the genre of the songs was an important factor when evaluating the similarity of songs. Many of the explanations provided by ChatGPT were also about music genre or instrumentation, with the latter being an indirect indication of genre. A decisive difference is however that we of course trust in the honesty of human participants when answering post-experiment questions concerning their strategies, while with ChatGPT such trust seems unwarranted. ChatGPT has been criticized for sometimes ‘hallucinating’ [6] facts that sound plausible but are actually incorrect. We verified that ChatGPT’s argumentation seems to be correct basically all the time by searching and reading respective online sources (e.g. Wikipedia or Discogs), or by listening to the audio. Nevertheless the black box nature of LLMs and especially ChatGPT is a problem for judging construct validity. Since the exact training data and modeling approach are unknown [6], we have no way to judge whether ChatGPT really used genre clues for providing music similarity scores. One possibility is that respective webpages about artists and songs, often including genre information, have been part of ChatGPT’s training data, allowing ChatGPT to reproduce this content when being queried accordingly. Indeed recent results indicate that LLMs seem to memorize large parts of their training data [26].

One way to test the hypothesis that ChatGPT uses genre information when judging music similarity is to repeat the experiment with music from a single genre. The initial study [10] repeated the listening tests with 90 songs all belonging to the genre **Power Pop** (for a full list see section B of the appendix of the original article (<https://doi.org/10.5334/tismir.107.s1>)) with 28 participants of an average age of 25.6. The average inter-rater agreement between human participants ρ_{listen} dropped from 0.74 to 0.24 when the song material was restricted to a single genre (see table 1). We now repeat the ChatGPT experiments with the restriction to **Power Pop** songs only. The average inter-rater agreement between human participants and ChatGPT ρ_{gpt} drops from 0.58 to 0.06 (see table 1). It seems that without the possibility to resort to genre information, ChatGPT has severe problems to rate music similarity. In the explanations provided by ChatGPT, it is often correctly stated that both songs “were part of the power pop genre during the same era”, but sometimes also subgenres are being named when justifying certain scores, e.g. “... leans towards a pop-rock sound ... while ... tends to blend progressive and art-rock elements” or “... was associated with power pop and new wave music, while ... was known for its indie rock and power pop sound”. Nevertheless the scores provided by ChatGPT remain very restricted, essentially consisting of three values (30, 40, 50) around the middle of the possible range.

From these results it seems evident that the poor operationalisation of the experiment, essentially not asking a clear enough question, has led to a lack of construct validity: we were aiming for music similarity as a higher order construct but music genre seems to also have been a

relevant aspect for both human participants and ChatGPT when answering questions during the experiments. For the human participants this problem became already evident during post-experiment questioning and was then only corroborated with the restricted single genre experiment. The lack of trust due to ChatGPT's black box nature however made the same experiment inevitable to clarify construct validity of the ChatGPT experiment.

Another way to question construct validity is to assess the outcomes of different experiments which are supposed to measure the same higher order constructs. We could for instance study correlations of results from different LLMs being queried with identical prompts. Low correlations between LLM outputs could point to problems of construct validity. This kind of testing already points to the concept of external validity.

7. EXTERNAL VALIDITY

External validity is "the validity of inferences about the extent to which a causal relationship holds over variations in experimental units, settings, treatment variables and measurement variables" [8]. Therefore, external validity is the truth of a generalised causal inference made from an experiment. It is clear that if a causal inference we draw from an experiment already lacks internal validity, then generalising that inference to variations not even tested will not have external validity. In addition, a major threat is that variation of components of an experiment might dismantle the causal inference that holds in the experiment.

One component that could be varied are the annotators, i.e. the human participants or the type of LLM employed. Already in the initial experiment [10] it became clear that human annotators only agree to a certain extent in their evaluation of music similarity (average ρ_{listen} of 0.74). This is because human perception of music is highly subjective with personal taste, listening history, familiarity with the music, current mood, etc, being important influencing factors [24, 27]. Such a lack of inter-rater agreement presents a problem of external validity because inferences from the experiment do not generalize from users or annotators in the experiment to the intended target population of arbitrary users/annotators. It would be interesting to test the level of agreement between ChatGPT-3.5, which has been used for the experiments in this paper, and newer versions like ChatGPT-4 [6] or even alternative LLMs like Google's Gemini (<https://gemini.google.com/>), LLaMA [28] or Alpaca [29]. There already is evidence that ChatGPT's responses differ between different versions [30]. In case we want the conclusions drawn from our experiment to have external validity beyond one specific type of LLM, such additional experiments would of course be necessary.

One could even ask the question what the level of agreement within one person is when faced with identical annotation tasks at different points in time. Results from the initial experiment already showed that such an intra-rater agreement, tested two weeks apart, is only slightly higher than inter-rater agreement [10] at 0.80 versus 0.74.

We therefore also repeated our ChatGPT five genre experiment on January 5, 2024, nine month after the first experiment. Although the LLM used was supposedly still a ChatGPT-3.5 version, the intra-rater agreement was only at 0.68. This is actually not much higher than the inter-rater agreement between human participants and ChatGPT ρ_{gpt} at 0.58. It therefore seems that there is a lack of external validity when generalizing ChatGPT results to different points in time.

Another problem of external validity is the influence of prompt engineering on LLM results. It is known that slight variations in prompt formulation can lead to quite different results, which brought about a whole new 'science' of so-called 'prompt engineering' [31]. One example is 'chain-of-thought prompting', where a few chain of thought demonstrations provided to an LLM as prompts lead to improved results [32]. 'Positive thinking' prompts like "You are an expert mathematician" also improve LLM performance and automatic prompt optimization sometimes produces quite bizarre results [33]: answer prefixes with an affinity to the science fiction show Star Trek (e.g.: "Captain's Log, Stardate [insert date here]: We have successfully plotted a course through the turbulence and are now approaching the source of the anomaly.") are able to boost some LLM's proficiency in mathematical reasoning. The reproducibility of LLM experiments seems doubtful given that seemingly irrelevant variations in prompting can have such big influence on results.

8. DISCUSSION AND CONCLUSION

In this work we applied the formal framework of validity [8] to music knowledge discovery, thereby enabling a critical appraisal of using Large Language Models (LLMs) for 'distant reading' of music knowledge. This was demonstrated for the extraction of psychophysical information from text by comparing GPT-3.5 results to human auditory experience. Specifically we re-evaluated our own previous results [7] of using ChatGPT to gain 'distant reading' estimates of music similarity. By evaluating a number of threats to validity and conducting additional experiments with ChatGPT, we were able to show that internal, construct and external validity of our approach are seriously compromised.

A re-analysis of music similarity ratings separate for different or similar pairs of music genres showed that in our experiment music similarity is confounded with genre. Both human participants and ChatGPT at least partly rely on the confounding factor of genre when judging music similarity, which is a clear breach of internal validity. This lead us to scrutinize the operationalisation of the experiment and its construct validity. A closer assessment of the exact questions being asked during the experiment made it evident that they are not precise enough concerning what is actually meant with music similarity. Post-experiment interviews with human participants made clear that they indeed used genre as indication of music similarity. Because of the blackbox nature of ChatGPT, and its doubtful relation to factuality, we had to conduct an additional ex-

periment to corroborate that ChatGPT also relies on genre information. This additional experiment with music from a single genre lead to a complete breakdown of the correlation between human and ChatGPT estimates of music similarity. We also appraised external validity by asking whether our results would generalize to variations in the experimental setting like employing different LLMs or versions thereof or making slight changes to prompts. We conducted a repetition of the ChatGPT experiment with a nine month time lag and showed that correlation of results is moderate at best, although the LLM is supposedly still based on the same version of GPT-3.5.

The overarching question we wanted to answer with this work is whether LLMs can be used as a distant reading tool of music knowledge. The main obstacle seems to be the opaqueness of systems like ChatGPT which make it very hard to judge their construct validity. This opaqueness is evident from the developing team’s own statements concerning their unwillingness to share details about their algorithm [6]. This has lead researchers to state that "it is particularly hard to perform scientific experiments, especially since human feedback causes their behaviours to change at a rapid pace" [34]. The latter statement points to the additional problem of constant re-training of models, which might explain the lack of external validity we observed when repeating our experiment with a nine month time lag. It has also lead to speculations as to how ChatGPT actually works, e.g. showing that it performs better when the correct output is a high-probability word sequence, indicating that one should be careful in low-probability situations [35]. This might be connected to the fact LLMs seem to memorize large parts of their training data [26]. It has also been pointed out that the “reasoning process” of LLMs is fundamentally different from humans, as LLMs basically just sample from a probability distribution [34]. As they are not embodied agents in the physical world, their understanding and knowledge lacks symbol grounding [36]. LLMs do not experience the world directly but model the world of text, which of course is a very indirect representation of the real world.

As a concluding comment we want to state that ChatGPT is not a suitable tool for distant reading of music knowledge because of its essentially black box nature which entails severe problems of judging its construct validity. Future work should explore whether open source alternatives like LLaMA [28], Alpaca [29] or OpenAssistant (<https://github.com/LAION-AI/Open-Assistant>) will be able to change assessment of the usefulness of large language models for distant reading.

9. ACKNOWLEDGMENTS

This research was funded in whole by the Austrian Science Fund (FWF) [10.55776/P36653]. For open access purposes, the authors have applied a CC BY public copyright license to any author accepted manuscript version arising from this submission.

10. ETHICS STATEMENT

For all experiments with human involvement, informed consent to participate in the respective studies was obtained from participants in accordance with university and international regulations.

As a potential societal implication we like to mention the fact that it is not known what precise data any of the ChatGPT versions have been trained on. There is however a reasonable suspicion that OpenAI, the company behind ChatGPT, did not obtain legal consent from all creators of text it used during training procedures. As a consequence, anyone using ChatGPT for their own purposes, including distant reading of music knowledge, would be implicated with the corresponding ethical issues.

11. REFERENCES

- [1] S. Oramas, L. Espinosa-Anke, F. Gómez, and X. Serra, “Natural language processing for music knowledge discovery,” *Journal of New Music Research*, vol. 47, no. 4, pp. 365–382, 2018.
- [2] F. Moretti, “Conjectures on world literature,” *New left review*, vol. 2, no. 1, pp. 54–68, 2000.
- [3] —, *Graphs, maps, trees: abstract models for a literary history*. Verso, 2005.
- [4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2019.
- [5] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, “Language models are few-shot learners,” *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.
- [6] OpenAI, “GPT-4 technical report,” *arXiv preprint arXiv:2303.08774*, 2023.
- [7] A. Flexer, “Can ChatGPT be useful for distant reading of music similarity?” in *HCMIR23: 2nd Workshop on Human-Centric Music Information Research, Milan, Italy*, 2023.
- [8] W. R. Shadish, T. D. Cook, and D. T. Campbell, *Experimental and quasi-experimental designs for generalized causal inference*. Boston: Houghton Mifflin, 2002.
- [9] B. L. T. Sturm and A. Flexer, “A review of validity and its relationship to music information research,” in *Proceedings of the 24th International Society for Music Information Retrieval Conference*, 2023, pp. 47–55.
- [10] A. Flexer, T. Lallai, and K. Rašl, “On evaluation of inter- and intra-rater agreement in music recommendation,” *Transactions of the International Society for Music Information Retrieval*, vol. 4(1), pp. 182–194, Nov 2021.

- [11] K. Siedenburg and C. Saitis, “The language of sounds unheard: Exploring musical timbre semantics of large language models,” *arXiv preprint arXiv:2304.07830*, 2023.
- [12] R. Marjeh, I. Sucholutsky, P. van Rijn, N. Jacoby, and T. L. Griffiths, “Large language models predict human sensory judgments across six modalities,” *arXiv preprint arXiv:2302.01308*, 2023.
- [13] Y. Zhang, J. Jiang, G. Xia, and S. Dixon, “Interpreting song lyrics with an audio-informed pre-trained language model,” in *Proceedings of the 23rd International Society for Music Information Retrieval Conference*, 2022, pp. 19–26.
- [14] Y. Hou, J. Zhang, Z. Lin, H. Lu, R. Xie, J. McAuley, and W. X. Zhao, “Large language models are zero-shot rankers for recommender systems,” *arXiv preprint arXiv:2305.08845*, 2023.
- [15] P. Knees, E. Pampalk, and G. Widmer, “Artist classification with web-based data,” in *Proceedings of the 5th International Conference on Music Information Retrieval*, 2004.
- [16] D. Turnbull, L. Barrington, D. Torres, and G. Lanckriet, “Towards musical query-by-semantic-description using the cal500 data set,” in *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, 2007, pp. 439–446.
- [17] B. Logan, A. Kositsky, and P. Moreno, “Semantic analysis of song lyrics,” in *IEEE International Conference on Multimedia and Expo (ICME)*, vol. 2. IEEE, 2004, pp. 827–830.
- [18] F. Góes, Z. Zhou, P. Sawicki, M. Grzes, and D. G. Brown, “Crowd score: A method for the evaluation of jokes using large language model ai voters as judges,” *arXiv preprint arXiv:2212.11214*, 2022.
- [19] L. F. Góes, P. Sawicki, M. Grzes, D. Brown, and M. Volpe, “Is GPT-4 good enough to evaluate jokes?” in *Proceedings of the 14th International Conference on Computational Creativity*, 2023.
- [20] P. Knees and M. Schedl, “Towards semantic music information extraction from the web using rule patterns and supervised learning,” in *Workshop on music recommendation and discovery*, 2011, pp. 18–25.
- [21] I. Fujinaga and S. F. Weiss, *Digital prosopography for renaissance musicians: Discovery of social and professional networks*. NEH White Paper, 2016.
- [22] M. Sordo, J. Serrà Julià, G. K. Koduri, and X. Serra, “Extracting semantic information from an online carnic music forum,” in *Proceedings of the 13th International Society for Music Information Retrieval Conference*, 2012.
- [23] R. F. E. Sutcliffe, T. Crawford, C. Fox, D. L. Root, E. H. Hovy, and R. Lewis, “Relating natural language text to musical passages,” in *Proceedings of the 16th International Society for Music Information Retrieval Conference*, 2015, pp. 524–530.
- [24] M. Schedl, A. Flexer, and J. Urbano, “The neglected user in music information retrieval research,” *Journal of Intelligent Information Systems*, vol. 41, no. 3, pp. 523–539, 2013.
- [25] F. Fabbri, “A theory of musical genres. two applications,” in *Popular music perspectives*, D. Horn and P. Tagg, Eds. Göteborg and Exeter, International Association for the Study of Popular Music, 1982, vol. 1, pp. 52–81.
- [26] M. Nasr, N. Carlini, J. Hayase, M. Jagielski, A. F. Cooper, D. Ippolito, C. A. Choquette-Choo, E. Wallace, F. Tramèr, and K. Lee, “Scalable extraction of training data from (production) language models,” *arXiv preprint arXiv:2311.17035*, 2023.
- [27] A. Flexer and T. Grill, “The problem of limited inter-rater agreement in modelling music similarity,” *Journal of New Music Research*, vol. 45, no. 3, pp. 239–251, 2016.
- [28] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, “Llama: Open and efficient foundation language models,” *arXiv preprint arXiv:2302.13971*, 2023.
- [29] R. Taori, I. Gulrajani, T. Zhang, Y. Dubois, X. Li, C. Guestrin, P. Liang, and T. B. Hashimoto, “Stanford alpaca: An instruction-following llama model,” https://github.com/tatsu-lab/stanford_alpaca, 2023.
- [30] L. Chen, M. Zaharia, and J. Zou, “How is chatgpt’s behavior changing over time?” *arXiv preprint arXiv:2307.09009*, 2023.
- [31] B. Chen, Z. Zhang, N. Langrené, and S. Zhu, “Unleashing the potential of prompt engineering in large language models: a comprehensive review,” *arXiv preprint arXiv:2310.14735*, 2023.
- [32] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou *et al.*, “Chain-of-thought prompting elicits reasoning in large language models,” *Advances in Neural Information Processing Systems*, vol. 35, pp. 24 824–24 837, 2022.
- [33] R. Battle and T. Gollapudi, “The unreasonable effectiveness of eccentric automatic prompts,” *arXiv preprint arXiv:2402.10949*, 2024.
- [34] M. Peeperkorn, D. Brown, and A. Jordanous, “On characterizations of large language models and creativity evaluation,” in *Proceedings of the 14th International Conference on Computational Creativity*, 2023.

- [35] R. T. McCoy, S. Yao, D. Friedman, M. Hardy, and T. L. Griffiths, “Embers of autoregression: Understanding large language models through the problem they are trained to solve,” *arXiv preprint arXiv:2309.13638*, 2023.
- [36] S. Harnad, “The symbol grounding problem,” *Physica D: Nonlinear Phenomena*, vol. 42, no. 1-3, pp. 335–346, 1990.

SANIDHA: A STUDIO QUALITY MULTI-MODAL DATASET FOR CARNATIC MUSIC

Venkatakrishnan Vaidyanathapuram Krishnan¹ Noel Alben¹ Anish Nair¹
Nathaniel Condit-Schultz¹

¹ School of Music, Georgia Institute of Technology, USA
{vkrishnan65, noelalben3, anair323, natcs@gatech.edu}

ABSTRACT

Music source separation demixes a piece of music into its individual sound sources (vocals, percussion, melodic instruments, etc.), a task with no simple mathematical solution. It requires deep learning methods involving training on large datasets of isolated music stems. The most commonly available datasets are made from commercial Western music, limiting the models' applications to non-Western genres like Carnatic music. Carnatic music is a live tradition, with the available multi-track recordings containing overlapping sounds and bleeds between the sources. This poses a challenge to commercially available source separation models like Spleeter and Hybrid Demucs. In this work, we introduce *Sanidha*, the first open-source novel dataset¹ for Carnatic music, offering studio-quality, multi-track recordings with minimal to no overlap or bleed. Along with the audio files, we provide high-definition videos of the artists' performances. Additionally, we fine-tuned Spleeter, one of the most commonly used source separation models, on our dataset and observed improved SDR performance compared to fine-tuning on a pre-existing Carnatic multi-track dataset. The outputs of the fine-tuned model with *Sanidha* are evaluated through a listening study.

1. INTRODUCTION

Carnatic music is a traditional "art music" genre from the Southern part of India. Carnatic Music is largely improvised, requiring all musicians to utilize a complex understanding of the melodic and rhythmic structures of the music to improvise coherently. Carnatic performances generally feature four to five musicians centered around a vocalist in the lead role. The core instruments are the violin, in both supportive and lead roles; the *mridangam*, a tonal two-sided drum that provides rhythmic support; and

¹ *Sanidha* dataset (Licensed under CC-BY-4.0) is hosted in the server: <https://ccml.gtcmt.gatech.edu/data/Sanidha>

the *ghatam*, a clay pot instrument that contributes rhythmic patterns to complement the *mridangam* in a higher frequency range. Carnatic Music performances are also accompanied by a *tanpura*, which constantly oscillates the *sa*, the tonic, and either the *pa*, the fifth or sometimes *ma*, the fourth. All the instruments are tuned to these frequencies, including the percussion instruments, which, too, have tonal qualities [1]. This leads to a significant overlap of frequency content, making Carnatic Music source separation almost impossible with simple dictionary learning methods [2].

Like most traditional music genres, Carnatic Music is performed live [1]. Thus, recordings of Carnatic Music lack multi-track isolation, as microphones inevitably capture signals from multiple instruments as well as the audience—these unwanted signals are known in music production as leakage or “bleed.” This contrasts with Western pop music, where completely isolated multi-tracks are commonplace, and many source separation datasets are available [3–6]. The most extensive open-source Music Information Retrieval (MIR) dataset of Indian art music—the *Saraga* dataset [7]—exhibits significant leakage between different audio tracks: For example, the sound of the violin is audible in the vocal track - The bleeding of other sources into other microphones is significant [8–10].

1.1 Leakage Problem

Consider a signal \mathbf{s} , noise \mathbf{n} , and a mix \mathbf{x} , at 0 dB Signal-to-Noise Ratio (SNR): $\mathbf{x} = \mathbf{s} + \mathbf{n}$, where $\mathbf{x}, \mathbf{s}, \mathbf{n} \in \mathbb{R}^d$. Let $\mathbf{s}_t, \mathbf{n}_t \in \mathbb{R}^d$ such that they represent ground truth signal and noise with bleed. Assume no microphone sensor noise and no Room Impulse Response (RIR). Then

$$\mathbf{x} = \mathbf{s}_t + \mathbf{n}_t \quad (1)$$

$$\mathbf{s}_t = f(\mathbf{s}, \mathbf{n}) = \alpha \mathbf{s} + \beta \mathbf{n} \quad (2)$$


where $\alpha \in [0, 1]$ and $\beta \in [0, 1]$, using Eq. 1, it follows that

$$\mathbf{n}_t = g(\mathbf{s}, \mathbf{n}) = (1 - \alpha)\mathbf{s} + (1 - \beta)\mathbf{n} \quad (3)$$

Assume that functions f and g are linear time-invariant functions for all audios. However, the α and β values will vary for different signals in a general unclean dataset.

Let the source separation function trained with $(\mathbf{s}_t, \mathbf{n}_t)$ as the ground truth be \mathbf{F} , such that

$$\mathbf{F}(\mathbf{x}) = (\hat{\mathbf{s}}, \hat{\mathbf{n}})$$

 © V. Vaidyanathapuram Krishnan, N. Alben, A. Nair, and N. Condit-Schultz. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** V. Vaidyanathapuram Krishnan, N. Alben, A. Nair, and N. Condit-Schultz, “Sanidha: A Studio Quality Multi-Modal Dataset for Carnatic Music”, in *Proc. of the 25th Int. Society for Music Information Retrieval Conf.*, San Francisco, United States, 2024.

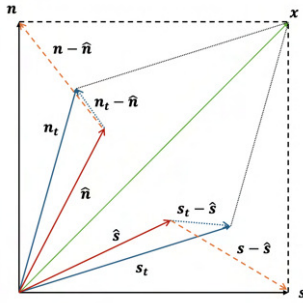


Figure 1. Problem of Poor Ground Truth

For simplicity, let us assume \hat{s} , s_t and \mathbf{n}_t lie in the same subspace as s and \mathbf{n} ; s and \mathbf{n} are orthogonal to each other i.e. $s^T \mathbf{n} = 0$ as seen in Figure 1.

The most common metric used for evaluation and loss in the source separation community is the Signal-to-Distortion Ratio (SDR) and, more recently, the Scale-Invariant version of SDR called SI-SDR [11]. For simplicity, let's consider using SDR for evaluation since the idea can easily be extended to SI-SDR. SDR is defined by [12] for the `BSS_eval` toolbox (which is the same as classical SNR) as:

$$\text{SDR}_s = 10 \log_{10} \left(\frac{\|s\|}{\|s - \hat{s}\|} \right)$$

Given that f and g functions vary for each audio, the SDR formula above is modified for data with bleed as:

$$\text{SDR}_{s,mod} = 10 \log_{10} \left(\frac{\|s_t\|}{\|s_t - \hat{s}\|} \right)$$

These objective results from SDR, however good, will never truly represent what the original source must sound like. Training on data with a significant bleed will never push the predicted \hat{s} towards the actual source s , since the loss function will be trained on the modified function dependent on sources with bleeding.

Furthermore, the result will be subpar after incorporating scale invariance [11]. If we calculate the norm of s_t and \mathbf{n}_t , using Eq. 2, 3, and the triangle inequality, we can prove:

$$\|s_t\| + \|\mathbf{n}_t\| \leq \|s\| + \|\mathbf{n}\| \quad (4)$$

This means that if we had to calculate the average SI-SDR of the signal and the noise with respect to the sources with bleed, the error would be significant. This error will be large when compared to calculating it with respect to "true" sources, which are inaccessible. It is also important to note that this was based on the assumption that all were in the same subspace, but that is never true in real scenarios, resulting in increased error.

Hence, the *Saraga* dataset cannot be used as accurate ground truth data for supervised source separation models for both training and especially evaluation, hindering the development of such models for Carnatic Music. As a workaround, some have attempted using source separation models like Spleeter [13], presumably trained on

a few or no Carnatic Music examples [9], directly on the vocal multi-track with bleeding for certain MIR tasks [8, 10]. However, attempts toward source separation for Carnatic using the currently available datasets have been made [2, 9, 14].

The stems obtained for Western Music datasets [3–6] are all from studio recordings, recorded separately and mixed, resulting in zero bleeds of other instruments in the multi-tracks. This allows for evaluation metrics such as SDR, Scale-Invariant SDR [11], Signal-to-Artifacts Ratio, Signal-to-Interference Ratio (SIR), etc., to be used without problems. However, there is no such available dataset for Carnatic Music [2, 9], and since it is a live tradition, it is impossible to record the artists at separate times.

There have been a lot of datasets for Carnatic Music and Hindustani Music, which provide clean studio-quality data for individual instruments [15, 16]. However, there have been no completely isolated full live concert recordings of studio quality. To directly address this requirement, we present a new dataset of well-isolated multi-track recordings of Carnatic Music: *Sanidha*. The *Sanidha* dataset features audio and video recordings of Carnatic musicians playing together in real-time but in total isolation within a modern studio environment.

2. METHODOLOGY

Serra [17] proposed five essential considerations when creating new corpora: purpose, coverage, completeness, quality, and reusability. These considerations guided the creation of the *Saraga* dataset of Indian art music, [18], and we have worked to apply the same principles to the construction of *Sanidha*.

The isolated tracks for the commercial Western music source separation datasets are often created by the process of overdubbing in the studio. Carnatic Music must be improvised collectively in real-time, so parts cannot be "overdubbed" one at a time, thereby posing a significant challenge. Carnatic musicians listen closely to each others' playing and communicate extensively using visual cues. In particular, the vocalist often indicates the *taalam* (metric structure) with their hands. Visual cues are critical during fully improvisational sections like the *kalpana swaram* and *tani avartanam*. Consequently, the only way to record the music with audio isolation is for each musician to play in separate rooms while maintaining communication through audio and video.

2.1 Recording Sessions

We organized five Carnatic music concerts within our recording facility in March of 2024. Concert sessions lasted 2–3 hours, garnering an average of 1.6 hours of music per concert once silence between pieces was edited out. To perform these concerts, we recruited fifteen professional Carnatic musicians from Atlanta's thriving Carnatic music scene. All the artists voluntarily agreed to contribute to the dataset for research purposes, with no compensa-

tion².

Our musicians included three male vocalists, two female vocalists, four violinists, and six percussionists. Two out of five concerts featured a vocalist accompanied by the full set of core Carnatic instruments (violin, *mridangam*, and *ghatam*). The other three concerts proceeded without a *ghatam* player—which is not unusual for the style. Through the efforts of multiple talented musicians, we were able to capture gender diversity in the vocal timbre and a wide array of stylistic and improvisational approaches, which enhances the value of our data to the research community.

2.2 Recording Facility and Setup

The dataset was recorded in four rooms of the West Village Music Annex, in the Georgia Institute of Technology’s campus in Atlanta, Georgia, USA. These rooms are multi-purpose spaces with large acoustic curtains, which enhanced our ability to control reverberation and maintain adequate isolation. The four isolated rooms have connection points wired to a single recording control room, including low-impedance, balanced analog audio, and digital video (SDI) connections. The control room uses a 32-channel digital mixing console to control audio routing and doubles as a multi-channel audio interface for digital audio recording into our Digital Audio Workstation (DAW). A *tanpura* drone, generated by a *shruti* box or a video from the internet, was also routed to each artist’s headphones from the control room. We used the board’s on-board reverb, compression, and equalization effects to create custom monitoring mixes sent to their headphones/in-ears, catering to individual artist needs and simulate the live traditional performing scenario of Carnatic Music.

Each artist’s performance video was captured using a professional 4K video camcorder. The recorded video feed was then delivered through SDI cables from each room to the control room to generate a multi-source mixed feed, allowing us to transmit all four video feeds within a 2x2 grid (Figure 2). Musicians could see the 2x2 feed projected onto a screen in the performance room, allowing them to observe each other at all times.

Our musicians had little to no experience performing in a studio setting, isolated from each other, with headphones/in-ears on. Our efforts were focused on ensuring that the recording sessions were comfortable for the musicians and maintained the “natural” performance feeling as much as possible. Despite our best efforts, our musicians noted specific challenges performing within the constraints of the setup and sometimes felt that it slightly affected the quality of their performance.

Though our audio-monitoring setup achieved close to zero latency, we found that our video-monitoring setup lagged by about 50 ms, possibly due to the converters used to transmit the video feed to the projectors. This made it extremely difficult for the artists to coordinate with each

other, since they could not follow the *taalam* or beat given by the vocalist. To overcome this problem, we used a *proxy-taalam* setup. One of our team members would sit in front of each artist (except the vocalist) and provide the visual *taalam* cue by focusing on just the audio feed from the vocalist. This setup was most helpful for our percussion artists; even the violinists appreciated it during the improvised *kalpana swaram* sections. The *proxy-taalam* setup allowed the musicians to play in time with each other, react to the cues from the vocalist similar to a live concert setup, and make the improvisational sections of Carnatic Music - *tani avartanam* and *kalpana swaram* sections possible.

We also identified a potential issue much later when we observed that some artists partially removed one side of their headphones in the middle of their performance. In some cases, artists required loud headphone output. This resulted in slight bleeding of the headphone output to the performer’s microphone. To combat this, we shifted the monitoring system to in-ear monitors exclusively for all further concerts, which nullifies possible bleeding from headphones.

2.3 Audio Data

For each concert, we recorded six (excluding *ghatam*) or eight (full group) separate unprocessed audio tracks. Vocalists were recorded using a single microphone; the other instruments were recorded using two microphones each. We captured the violin and *mridangam* in a standard stereo (left-right) image. The *ghatam* recording setup used two microphones as well. A line-in track was used to record the *tanpura* drone.

In total, we have nearly eight hours of recorded music, across the five concert sessions. The recorded audio is in WAV format, with CD-standard sampling rate of 44.1 kHz and a bit depth of 16 bits. Table 1 displays all the individual concert durations.

2.3.1 Microphones

For each concert, we used different combinations of microphones, maximizing the sonic variety of the data. The choices of microphones were professional, studio-grade condenser microphones with cardioid polar pickup patterns, with each instrument requiring matched pairs of identical microphones. The use of high-fidelity condenser microphones contrasts with the dynamic microphones used commonly in traditional Carnatic music concerts. However, capturing the highest fidelity audio will produce the most broadly usable data. A series of non-linear operations can be performed at the post-processing stage to alter high-fidelity signals to sound more like dynamic microphones. The details of the microphones used for each instrument are stored in a JSON file located within respective concert folders.

For our first concert, the vocal microphone was placed close to the vocalist’s mouth. We realized that this position obstructed the video of the performer’s face. For all subsequent concerts, we corrected this by placing microphones

² The concerts were conducted with the approval of the Georgia Tech Institution Review Board (IRB) (ethics board), including two minors who were accompanied by their parents.

Concert	Instruments	Multi-tracks	Front-View Video	Side-View Video	Duration (hr)	Vocals Gender
1	Vocal	1	✓	-	1.08	Female
	Violin	2	✓	-		
	Mridangam	2	✓	-		
	Ghatam	2	✓	-		
2	Vocal	1	✓	✓	1.63	Male
	Violin	2	✓	-		
	Mridangam	2	✓	-		
3	Vocal	1	✓	✓	1.37	Male
	Violin	2	✓	-		
	Mridangam	2	✓	-		
	Ghatam	2	✓	-		
4	Vocal	1	✓	✓	1.97	Female
	Violin	2	✓	-		
	Mridangam	2	✓	-		
5	Vocal	1	✓	✓	1.92	Male
	Violin	2	✓	-		
	Mridangam	2	✓	-		

Table 1. Dataset Details

closer to chest level, pointing upwards towards the mouth, ensuring an obstruction-free video.

We placed microphones for the violin and *mridangam* on either side of the artist, at a distance of approximately 50 cm. This positioning ensured microphone stability, kept the video feed unobstructed, and highlighted each instrumentalist’s gestures and hand movements. As the *ghatam* is a relatively quiet instrument, we placed the first microphone as close as possible to the playing surface. The second was pointed toward the opening of the *ghatam* at a distance of ≈ 30 cm. This can be seen in Figure 2.

2.4 Video Data

Performance video data for Carnatic is significantly limited compared to Hindustani music. The access to video data has given rise to a significant interest in the multi-modal analysis of Hindustani music among the MIR community [16, 19–21]. Our motivation to include video recordings with our dataset is to promote multi-modal research endeavors for Carnatic music.

All of our videos are recorded at 29.97 FPS in 1080p. The snapshot of the front view videos of each instrumentalist can be seen in Figure 2. The lighting for all the videos takes advantage of the many light sources available in the multi-purpose recording rooms.

For each concert, we successfully captured the front-view videos of every musician and included an additional side view of the vocalist. This combination is a first for a dataset of this kind.

The framing of the front-view videos is similar to the stills used in [20]. To ensure a solid background, we placed solid black sound panels behind the vocalists and solid yellow curtains behind the other artists, as seen in Figure 2.


Figure 2. Snapshot of the Front-view videos of Concert 3

2.5 Supplementary Information

2.5.1 Metadata

To fulfill Serra’s completeness criteria, we collected annotations and metadata similar to *Saraga* [7]. This metadata is stored in separate JSON files for each song performed during the concerts. The metadata includes the composition name, original composer, and the performers’ names and roles. We also include relevant music-theory information regarding the compositions, mentioning the *rāgam*, *tālam*, and song form.

2.5.2 Section Annotations

The song form is encoded as audio timestamps indicating the start and end of each major musical section for every song: the key sections are the *aalapana*, *pallavi*, *anupallavi*, *muktayi swaram*, *charanam*, *cittai swaram*, *kalpana swaram* and *neraval*. The performing musicians were consulted to review all of the metadata.

2.5.3 Pitch Annotations

Carnatic music contains two melody sources: the lead vocals and the violin, which complements the vocals. Since

we have clean vocals and violin data, the Melodia algorithm proposed by Salamon and Gómez [22] was used to extract pitch (F0) contours for these two parts. The pitch tracks are stored in a two-column format, with the time stamps in the first column and the pitch values in the second.

2.5.4 Tonic Annotations

Obtaining the tonic frequency is relatively easy since we have a clean *tanpura* source within our multi-track data. We followed a similar approach used by Gulati et al. [23] and used Melodia [22] on the *tanpura* multi-track directly for the tonal feature extraction. The tonic does not change within a concert; hence, we included a single tonic file, which stores the tonic value in Hertz, inside each concert folder instead of having one for every song.

3. EXPERIMENTS

The experiments aim to cover the Coverage and Quality principles [17] introduced in Section 2 and demonstrate the value and usability of our new dataset with a simple source separation experiment.

It is important to note that our aim in this work is to demonstrate our data’s potential through these preliminary experiments and not benchmark performance against the state-of-the-art results for source separation of Carnatic Music.

3.1 Experiment Setup

We ran a simple two-stem source separation fine-tuning experiment on *Sanidha* and *Saraga* datasets using the Spleeter model [13]. Two-stem Spleeter training requires the vocals, accompaniment, and mix audios. We fine-tuned the pre-trained model using three different approaches: (1) using the *Sanidha* dataset, (2) using the *Saraga* dataset, (3) using curriculum training [24, 25] by partly fine-tuning the model with *Saraga*, and then fine-tuning it further with the *Sanidha* dataset. The curriculum training strategy presents the data to the model in a meaningful order to learn better. Using these three models will help us evaluate the potential of our data and its performance when combined with other Carnatic Datasets, in this case, *Saraga*.

Since *Sanidha* has fewer concerts than *Saraga*, the major problem which could arise, is the possibility of overfitting. To potentially avoid this, the third model is fine-tuned on *Saraga* for 225K steps (90% of the total steps), while the rest 10% is finetuned on *Sanidha* for 25K steps.

3.2 Sanidha Data Preparation

Sanidha’s audio data is of high quality as it was recorded in isolated spaces using condenser microphones with almost no bleed. Therefore, just linearly adding the signals to prepare mixes for training [26] will not be representative of the traditional Carnatic Concerts. To prevent this, we chose ten concerts from the *Saraga* dataset and used them as reference tracks to create two unique mixes for each of the five *Sanidha* concerts. Eight out of these ten *Saraga*

tracks are used as references for processing the training set and the remaining two are used for validation. The multiple mixes allow us to obtain twice the original amount of data. This can be considered as data augmentation since we have limited clean data. Our goal was to match the number of hours of training data used on the models individually trained on *Sanidha* and *Saraga* respectively, to make a fair comparison. The *Sanidha* training set makes up a total of 13.21 hours of audio data, and the validation set is 2.14 hours.

The critical mixing strategies for vocals and accompaniment include a combination of multiple non-linear and some time-varying operations - (1) Adding distortion, (2) Adding white noise, (3) Processing the stems through a digital amplifier plus cabinet models, (4) Heavy compression, (5) Adding reverb, (6) Attenuating the body of the instruments and vocals, and (7) High-cut filtering. Each of these operations is performed in varied amounts to match the sonic features of the reference tracks. The aim is to mix the tracks to emulate a real live concert while maintaining the isolated ground-truth audio.

The processed vocals (\mathbf{v}) and the processed accompaniment (\mathbf{a}) audios are linearly added at 0 dB SNR to create the mixture file ($\mathbf{m} = \mathbf{v} + \mathbf{a}$) for training. For the SNR computation, we consider the signal to be the vocals and the noise to be the accompaniment.

The third *Sanidha* concert was chosen for the validation set, as it has all of the typical instruments, including the *ghatam*. The rest of the concerts used in the training set maintain a good distribution of vocalist’s gender and vocal timbre, as seen in Table 1. We made two unique mixes for each song in the validation set, which totaled to 2.14 hours of mixture audio data.

3.3 Saraga Data Preparation

Seven out of the eight references *Saraga* concerts described in Section 3.2 make up the training set for the *Saraga*-trained model. As *Saraga* consists of live multi-track recordings from Carnatic concerts, the accompaniment audio is created by linearly adding all the multi-track audios except the vocals. For the validation set of the *Saraga*-trained model, we selected the same reference concerts from *Saraga* that were used to create the mixes for the validation set in Section 3.2.

The ground truth multi-tracks used have an inherent bleed in them [8–10], as described in Section 1. The purpose of using a noisy validation set from *Saraga* is to evaluate the model purely trained on *Saraga*, assuming the *Sanidha* dataset never existed. However, the metrics obtained in Table 2 are on the validation set used for *Sanidha* training. The total training duration comes to 12.37 hours. The remaining unused concerts in *Saraga* are used for the perceptual tests.

4. EVALUATION

4.1 Objective evaluation

We compute the SDR, SIR, SAR, and also the SI-SDR of each of the models for the *Sanidha* validation set. Table 2

			<i>Sanidha</i> - Objective Evaluation				<i>Saraga</i> - Perceptual Evaluation	
Models	Hours	Source	SDR	SIR	SAR	SI-SDR	Isolation	Audio Quality
<i>Saraga</i>	12.37	Vocals	7.66	17.05	8.02	6.65	0.596	0.627
		Accomp.	7.68	13.65	8.84	7.29	0.546	0.532
		Average	7.67	15.35	8.43	6.97	0.564	0.580
<i>Sanidha</i>	13.21	Vocals	7.86	17.38	8.26	6.93	0.598	0.635
		Accomp.	7.87	13.96	8.99	7.52	0.541	0.507
		Average	7.87	15.67	8.63	7.22	0.570	0.572
Mix	12.37 + 13.21	Vocals	7.63	16.88	8.00	6.62	0.605	0.621
		Accomp.	7.65	13.99	8.73	7.25	0.561	0.525
		Average	7.64	15.44	8.36	6.93	0.583	0.573

Table 2. Results

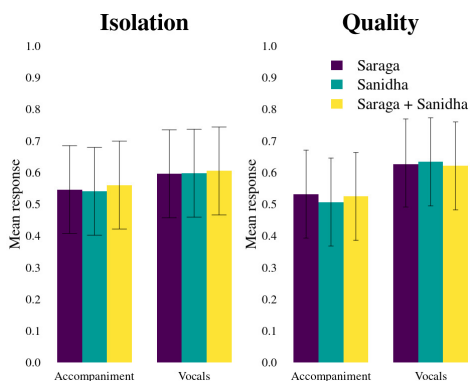


Figure 3. Mean participant responses across twelve conditions, with 95% confidence limits.

displays all the results.

We can see that the *Sanidha*-trained model has outperformed, however marginally, in all the objective metrics for vocals and the accompaniment separation. The improvement is only slight, perhaps because *Sanidha* was only trained on four concerts, while *Saraga* is trained on seven (which would mean seven unique vocalists as compared to four), even if the training hours are comparable. Also, the curriculum training technique performs almost similar to the *Saraga*-trained model.

4.2 Subjective evaluation

We conducted a listening study to evaluate the three source separation models and assess their perceptual effectiveness in isolating vocals and accompaniments in Carnatic Music.

The audio stimuli were selected after we randomly sampled four ten-second excerpts from four different *Saraga* recordings; If the randomly selected excerpt did not contain the three key instruments in Carnatic Music (vocals, violin, and the *mridangam*), we sampled again until an appropriate excerpt was identified. This iterative process ensured that our evaluation remains focused on relevant audio features while maintaining the unbiased nature of the sample selection.

In the listening study—approved by the Georgia Tech ethics board—fourteen participants listened to processed

versions of our selected excerpts. The survey was conducted in a manner similar to the MUSHRA framework [27]. All the participants responded to twelve questions for each excerpt, which focused on vocal isolation, vocal audio quality, accompaniment isolation, and accompaniment audio quality for the three models. This resulted in 48 questions per participant. These terms have been commonly used in subjective testing of source separation models [9, 28]. We used a slider-based metric for the evaluation, ranging from zero to one. Isolation and quality were explained with examples before the start of the survey and also presented as a reference for each question.

Average slider responses for the twelve conditions are shown in Figure 3 and in Table 2. We conducted a mixed-effects ANOVA on the data, with the participant and excerpt as random intercepts and the three variables (response type, target source, and model) as fixed effects. No effect was statistically significant, except for the target source (voice vs accompaniment), where participants tended to rate vocals higher in general ($\chi^2(8) = 45.97, p < .05$). This behavior is very similar to the objective results as well.

5. CONCLUSION AND FUTURE WORK

Although fine-tuning spleeter using *Sanidha* did not result in a significant source separation improvement, we cannot discount the importance of the availability of clean target sources for source separation. This is a clear distinction and advantage that our dataset collection methodology has over the existing *Saraga*. We can now use common metrics for source separation evaluation with a good degree of accuracy using our dataset, which was not possible with the existing *Saraga* dataset. Given the inherent challenges, our introduction of the *Sanidha* dataset marks a significant advancement in this domain. This novel dataset also presents an avenue for solving a multitude of other MIR and multi-modal tasks in Carnatic Music.

We will soon expand our dataset and invite more musicians to conduct concerts using our methodology. With the resources at hand, we aim to promote computational analysis for Indian Art music and pave the path towards more accessible research resources within the community.

6. ACKNOWLEDGEMENTS

We extend our gratitude to all the artists whose contributions have been pivotal to the creation of this dataset:

1. **Concert 1:** Amita Krishnan (Vocals), Sudharshan Prasanna (Violin), Vajraang Kamat (Mridangam), Tejas Veedhulur (Ghatam)
2. **Concert 2:** Salem Shriram (Vocals), Vishal Sowmyan (Violin), Anirudhah Narayanan (Mridangam)
3. **Concert 3:** Prashant Krishnamoorthy (Vocals), Nivik Sanjay Bharadwaj (Violin), Arvind Narayan (Mridangam), Vajraang Kamat (Ghatam)
4. **Concert 4:** Anjana Nagaraja (Vocals), Pranavi Srinivasa (Violin), Arvind Narayan (Mridangam)
5. **Concert 5:** Prasanna Soundararajan (Vocals), Vishal Sowmyan (Violin), Santosh Chandru (Mridangam)

7. REFERENCES

- [1] A. Srinivasamurthy, S. Gulati, R. Caro Repetto, and X. Serra, "Getting started on computational musicology and music information research: an indian art music perspective," *Rao P, Murthy HA, Prasann SRM, editors. Indian art music: a computational perspective.[New Delhi]: Scheme for Promotion Academic and Research Collaboration, 2023. p. 3-38., 2023.*
- [2] J. Sebastian and H. A. Murthy, "Group delay based music source separation using deep recurrent neural networks," in *2016 International Conference on Signal Processing and Communications (SPCOM)*. IEEE, 2016, pp. 1–5.
- [3] E. Manilow, G. Wichern, P. Seetharaman, and J. Le Roux, "Cutting music source separation some Slakh: A dataset to study the impact of training data quality and quantity," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2019.
- [4] Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimilakis, and R. Bittner, "Musdb18-a corpus for music separation," 2017.
- [5] R. M. Bittner, J. Salamon, M. Tierney, M. Mauch, C. Cannam, and J. P. Bello, "Medleydb: A multitrack dataset for annotation-intensive mir research." in *ISMIR*, vol. 14, 2014, pp. 155–160.
- [6] A. Liutkus, F.-R. Stöter, Z. Rafii, D. Kitamura, B. Rivet, N. Ito, N. Ono, and J. Fontecave, "The 2016 signal separation evaluation campaign," in *Latent Variable Analysis and Signal Separation - 12th International Conference, LVA/ICA 2015, Liberec, Czech Republic, August 25-28, 2015, Proceedings*, P. Tichavský, M. Babaie-Zadeh, O. J. Michel, and N. Thirion-Moreau, Eds. Cham: Springer International Publishing, 2017, pp. 323–332.
- [7] A. Srinivasamurthy, S. Gulati, R. C. Repetto, and X. Serra, "Saraga: Open datasets for research on indian art music," *Empirical Musicology Review*, vol. 16, no. 1, pp. 85–98, 2021.
- [8] T. Nuttall, G. Plaja-Roglans, L. Pearson, and X. Serra, "The matrix profile for motif discovery in audio-an example application in carnatic music," in *International Symposium on Computer Music Multidisciplinary Research*. Springer, 2021, pp. 228–237.
- [9] G. Plaja-Roglans, M. Miron, A. Shankar, and X. Serra, "Carnatic singing voice separation using cold diffusion on training data with bleeding," 2023.
- [10] G. Plaja-Roglans, T. Nuttall, L. Pearson, X. Serra, and M. Miron, "Repertoire-specific vocal pitch data generation for improved melodic analysis of carnatic music," *Transactions of the International Society for Music Information Retrieval*, Jun 2023.
- [11] J. L. Roux, S. Wisdom, H. Erdogan, and J. R. Hershey, "Sdr – half-baked or well done?" in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 626–630.
- [12] E. Vincent, R. Gribonval, and C. Fevotte, "Performance measurement in blind audio source separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 4, pp. 1462–1469, 2006.
- [13] R. Hennequin, A. Khelif, F. Voituret, and M. Mousallam, "Spleeter: a fast and efficient music source separation tool with pre-trained models," *Journal of Open Source Software*, vol. 5, no. 50, p. 2154, 2020, deezer Research. [Online]. Available: <https://doi.org/10.21105/joss.02154>
- [14] N. Dawalatabad, J. Sebastian, J. Kuriakose, C. C. Sekhar, S. Narayanan, and H. A. Murthy, "Front-end diarization for percussion separation in taniavartanam of carnatic music concerts," *arXiv preprint arXiv:2103.03215*, 2021.
- [15] K. Subramani and P. Rao, "Carnatic violin dataset," 2020. [Online]. Available: <https://doi.org/10.5281/zenodo.3940330>
- [16] M. Clayton, J. Li, A. R. Clarke, M. Weinzierl, L. Leante, and S. Tarsitani, "Hindustani raga and singer classification using pose estimation," 2021.
- [17] X. Serra, "Creating research corpora for the computational study of music: the case of the compmusic project," in *AES 53rd International Conference: Semantic Audio; 2014 Jan 27-29; London, UK. New York: Audio Engineering Society; 2014. Article number 1-1 [9 p.]*. Audio Engineering Society, 2014.
- [18] A. Srinivasamurthy, G. K. Koduri, S. Gulati, V. Ishwar, and X. Serra, "Corpora for music information research in indian art music," in *Georgaki A, Kouroupetroglou*

G, eds. *Proceedings of the 2014 International Computer Music Conference, ICMC/SMC; 2014 Sept 14-20; Athens, Greece.*[Michigan]: Michigan Publishing; 2014. Michigan Publishing, 2014.

- [19] S. Paschalidou and I. Miliaresi, “Multimodal deep learning architecture for hindustani raga classification,” *Sensors & Transducers*, vol. 260, no. 2, pp. 77–86, 2023.
- [20] M. Clayton, P. Rao, N. N. Shikarpur, S. Roychowdhury, and J. Li, “Raga classification from vocal performances using multimodal analysis.” in *ISMIR*, 2022, pp. 283–290.
- [21] T. Kelkar, “Applications of gesture and spatial cognition in hindustani vocal music,” Ph.D. dissertation, International Institute of Information Technology Hyderabad, India, 2015.
- [22] J. Salamon and E. Gómez, “Melody extraction from polyphonic music signals using pitch contour characteristics,” *IEEE transactions on audio, speech, and language processing*, vol. 20, no. 6, pp. 1759–1770, 2012.
- [23] S. Gulati, “A tonic identification approach for indian art music,” *Unpublished master’s thesis*. Universitat Pompeu Fabra, Barcelona, 2012.
- [24] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, “Curriculum learning,” in *Proceedings of the 26th annual international conference on machine learning*, 2009, pp. 41–48.
- [25] X. Wang, Y. Chen, and W. Zhu, “A survey on curriculum learning,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 9, pp. 4555–4576, 2021.
- [26] E. Manilow, P. Seetharman, and J. Salamon, *Open Source Tools & Data for Music Source Separation*. <https://source-separation.github.io/tutorial>, 2020. [Online]. Available: <https://source-separation.github.io/tutorial>
- [27] M. Schoeffler, S. Bartoschek, F.-R. Stöter, M. Roess, S. Westphal, B. Edler, and J. Herre, “webmushra—a comprehensive framework for web-based listening tests,” *Journal of Open Research Software*, vol. 6, no. 1, p. 8, 2018.
- [28] A. Défossez, “Hybrid spectrogram and waveform source separation,” *arXiv preprint arXiv:2111.03600*, 2021.

BETWEEN THE AI AND ME: ANALYSING LISTENERS' PERSPECTIVES ON AI- AND HUMAN-COMPOSED PROGRESSIVE METAL MUSIC

Pedro Sarmiento Jackson Loth Mathieu Barthet

Centre for Digital Music
Queen Mary University of London

{p.p.sarmiento, j.j.loth, m.barthet}@qmul.ac.uk

ABSTRACT

Generative AI models have recently blossomed, significantly impacting artistic and musical traditions. Research investigating how humans interact with and deem these models is therefore crucial. Through a listening and reflection study, we explore participants' perspectives on AI- vs human-generated progressive metal, in symbolic format, using rock music as a control group. AI-generated examples were produced by ProgGP [1], a Transformer-based model. We propose a mixed methods approach to assess the effects of generation type (human vs. AI), genre (progressive metal vs. rock), and curation process (random vs. cherry-picked). This combines quantitative feedback on genre congruence, preference, creativity, consistency, playability, humanness, and repeatability, and qualitative feedback to provide insights into listeners' experiences. A total of 32 progressive metal fans completed the study. Our findings validate the use of fine-tuning to achieve genre-specific specialization in AI music generation, as listeners could distinguish between AI-generated rock and progressive metal. Despite some AI-generated excerpts receiving similar ratings to human music, listeners exhibited a preference for human compositions. Thematic analysis identified key features for genre and AI vs. human distinctions. Finally, we consider the ethical implications of our work in promoting musical data diversity within MIR research by focusing on an under-explored genre.

1. INTRODUCTION

Recently, advancements in AI have resulted in generative models capable of creating remarkable musical pieces. This has been particularly evident in the audio domain, with models such as Jukebox (OpenAI) [2], MusicLM (Google) [3], AudioCraft/MusicGen (Meta) [4] and Stable Audio (Stability AI) [5], to name a few. In contrast to audio generative models, which can produce complete, directly perceivable music with limited user input beyond

specifying a prompt, symbolic music generation methods yield outputs that necessitate subsequent decoding and interpretation by performers and mixing by audio engineers before transforming them into music suitable for listening experiences. Unless automated using synthesis and machine mixing, by requiring human interpretation through performance and mixing, symbolic music rendering opens the door for the infusion of cultural and social elements. These elements become integral aspects of the final musical experience for listeners.

One major controversy surrounding AI music generation models is their training on copyrighted data, often without consent nor royalty mechanisms. This raises concerns that AI-generated music could threaten artists' and musicians' income streams, amongst others [6]. These issues apply to both symbolic and audio AI music generation. However, symbolic approaches may pose less risk to artists' revenue streams since human musicians are still essential to the final product.

In this work, we focus on symbolic generative AI applied to progressive metal, which is considered a sub-genre of metal. Building on progressive rock's complex phrasing and odd time signatures, it incorporates a heavier focus on guitars and metal influences. The genre encompasses prominent bands such as Dream Theater, Between The Buried And Me¹, and Meshuggah [7]. It is, however, relatively unexplored in academic literature, particularly in the context of AI music generation and MIR research [8] [9].

Guitar tablature (see Figure 1) is a symbolic musical notation that translates guitar notes into fret and string numbers. Due to the genre's emphasis on guitar, progressive metal bands commonly use tablature to notate their compositions. Given that technical complexity is a large appeal of the genre, artists often sell their music in the form of tablatures for learning purposes through tablature publishing companies². Musicians from within the genre often use digital representations of tablatures and software like Guitar Pro³ for dissemination of musical ideas and computer-assisted music making.

We conducted a listening and reflection study to explore participants' perceptions of AI-generated progressive metal music. We used examples generated by ProgGP [1],



© P. Sarmiento, J. Loth, and M. Barthet. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).
Attribution: P. Sarmiento, J. Loth, and M. Barthet, "Between the AI and Me: Analysing Listeners' Perspectives on AI- and Human-Composed Progressive Metal Music", in *Proc. of the 25th Int. Society for Music Information Retrieval Conf.*, San Francisco, United States, 2024.

¹ Used as inspiration for the title of this paper.

² As an example, Sheet Happens Publishing: <https://www.sheethappenspublishing.com/>

³ <https://www.guitar-pro.com/>



Figure 1. A screenshot from Guitar Pro of two measures from an AI-generated progressive metal song.

an AI model for multi-instrument guitar tablature creation (an example is shown in Figure 1). Overall, the contributions of this paper are: (1) a listening and reflective study methodology and questionnaire; (2) a subjective assessment of the capabilities of ProgGP for symbolic music generation, particularly in tablature format; (3) identification of compositional features of the progressive metal genre; (4) a critical analysis of AI-generated music through the lens of the progressive metal community; (5) an ethical reflection on musical data diversity in MIR, propelled by this study focusing on the underexplored progressive metal genre.

2. BACKGROUND

2.1 Symbolic Music Generation

Music generation has seen an increase in popularity due to recent advances in deep learning [10], with many researchers utilizing techniques such as Recurrent Neural Networks (RNNs) [11] [12], Variational Autoencoders (VAEs) [13], Generative Adversarial Networks (GANs) [14], and Transformers [15]. The Transformer model [16], known for its performance in natural language processing (NLP) tasks, has been adapted for generating symbolic piano music in Huang et al.’s Music Transformer [15], with Musenet [17] and Pop Music Transformer [18] further improving the approach.

The field of guitar tablature generation gained significant momentum with the release of the DadaGP dataset [19]. This dataset provides songs in two formats: GuitarPro, a popular tablature editing software, and a dedicated textual token format. This allows researchers to develop AI models that can both represent and generate music in tablature format. GTR-CTRL [20] implements a Transformer-based model [18] for generating tablature that incorporates multiple instruments. It offers control over instrumentation (inst-CTRL) and musical genre (genre-CTRL). ProgGP [1], the model used in this study, focuses specifically on the progressive metal genre (see Figure 1 and description in Section 3). LooperGP [21] adapts the method to generate loopable music excerpts, making it

applicable e.g. for live coding performances. By fine-tuning the model on the music of four iconic guitar players, ShredGP [22] demonstrates its ability to replicate specific styles.

2.2 Subjective Evaluation of AI-Generated Music

Objective computational measures can provide an initial assessment of AI-generated music quality [23]. However, often they struggle to capture the subtleties needed to judge their aesthetic merit. The combination of objective computational measures with subjective human evaluations provides a more holistic understanding of AI-generated music. Listening tests typically involve ranking or scoring AI- and human-generated stimuli according to several metrics to gain a more comprehensive understanding of perceived quality. This often involves comparing outputs from different models with the established reference (the known ideal or benchmark). Metrics used to assess AI music vary from general attributes such as *musicality* [15], *liking* [24] [25] [18] [26], *pleasantness* [27], *richness* [28], to more specific qualities such as *consistency* [29], or *structural/stability* properties [29]. Whereas ranking involves sorting the different stimuli along a given dimension, scoring tasks commonly rely on 5- or 7-point Likert items [30] [31]. A musical Turing test, similar to the original Turing test, is designed to assess a machine’s ability to exhibit human-level musical creation features. In such tests, participants attempt to distinguish between human- and AI-generated music [32]. To assess AI-generated music, we employ a mixed methods approach, combining quantitative and qualitative data from a listening and reflection study. This approach, common in music perception and HCI research (e.g. [33]), allows for a deeper understanding of the problem. While listening tests enable us to better understand human perception of AI-generated music, they are not without limitations. These limitations include listener fatigue, potential biases due to stimuli or participant selection. Additionally, they may lack sufficient statistical power to generalize the findings to a broader population.

3. METHODOLOGY

We used a mixed methods listening and reflective study to assess AI music, with an ethical approval from the Queen Mary Ethics of Research Committee. All data was collected anonymously. The study took around 1h to complete and participants were compensated with a £10 Amazon voucher.

We evaluated AI-generated progressive metal music from the ProgGP model [1], a Transformer-XL model trained on the DadaGP dataset [19] and fine-tuned on a progressive metal corpus, by comparing it to human-composed progressive metal pieces. We compare two ways of choosing AI music: picking songs at random and subjectively choosing the “best” ones (cherry-picking) through active listening. Additionally, we compare the ProgGP model’s outputs with rock music generated by the genre-CTRL model [20], a similar model conditioned on the rock genre. Human-composed rock music serves as another control group in this comparison.

3.1 Hypotheses

Our study tests the following hypotheses:

- **H_1 : Human-composed music obtains better scores than AI-generated music.** We compare AI- and human-generated music along the following dimensions: preference, creativity, consistency, playability and repeatability.
- **H_2 : AI-generated and human-composed music can be distinguished.** This hypothesis is linked to the musical Turing test.
- **H_3 : AI-generated music matches the genre used for model conditioning.** The ability of the model to specialize in a specific genre (progressive metal).
- **H_4 : Cherry-picked AI-generated music is preferred to randomly chosen AI-generated music.** We hypothesize that picking examples by hand leads to better performance than random selection.

3.2 Stimuli

The stimuli were rendered using Guitar Pro 7, a software for playing/editing digital guitar tablatures. The human-composed music was obtained using publicly available transcriptions of progressive metal and rock songs hosted on Songsterr⁴, a website hosting Guitar Pro tablatures, as well as from the DadaGP dataset [19]. All the examples were trimmed to 15 seconds, and rendered as WAV files using the default virtual instruments in Guitar Pro 7. They were further loudness-normalized [34]. The study comprised 60 stimuli⁵ broken down into the following six groups with 10 examples per group: **progcp** (progressive metal examples generated using ProgGP cherry-picked), **progrand** (progressive metal examples generated with ProgGP, randomly selected), **proghum** (progressive metal examples from the dataset used to fine-tune ProgGP, human-generated, randomly selected), **rockcp** (rock examples generated using genre-CTRL prompted for rock, cherry-picked), **rockrand** (rock examples generated using genre-CTRL prompted for rock, randomly selected), and **rockhum** (from rock examples in the dataset used for genre-CTRL, human-generated, randomly selected). The AI-generated stimuli were selected out of a corpus of 200 compositions from each genre.

3.3 Participants

We recruited participants familiar with progressive metal as we wanted to involve domain experts capable of identifying differences between rock and progressive metal. To this end, we advertised the call for participants on the *r/progmetal* sub-forum from the Reddit platform. This community comprises progressive metal aficionados⁶. 26 participants were gathered from this forum. We recruited six additional progressive metal fans within our department, for a total of 32 participants. Their age distribution

was 29 ± 5 years old, with 27 males and 5 females. The participants had an average Gold-MSI score of 81.41, indicating average level of musical sophistication compared to results of previous studies [35].

3.4 Procedure

Participants first went through a familiarization stage containing two excerpts, followed by the main task during which musical excerpts were presented in random order to minimise potential order effects. Participants were instructed to focus on the quality of the composition and not on the quality of the virtual instruments or the music production mix. For each excerpt, participants had to listen to the stimulus, report their familiarity, and answer the following questions using 7-point Likert items: Q_1 (“This composition features the qualities of the progressive metal genre.”), Q_2 (“This composition features the qualities of the rock genre.”), Q_3 (“I like this composition.”), Q_4 (“This composition is creative.”), Q_5 (“This composition is consistent.”), Q_6 (“This composition is playable.”), Q_7 (“This composition was generated using AI.”) and Q_8 (“This composition is repetitive.”). Once the participants finished rating all excerpts, they were presented with a post-task questionnaire to assess their reasoning when distinguishing between genres as well as between AI- and human-composed excerpts (see Section 4.2).

4. RESULTS

4.1 Listening Test

We visualize Likert item answers using violin plots in Figure 2. We conducted statistical analyses investigating the effects of the music creation process (six levels: **progcp**, **progrand**, **proghum**, **rockcp**, **rockrand**, **rockhum**) on several dependent variables (preference, creativity, consistency, playability, repeatability, humanness, genre congruency, and AI curation method, where relevant). Because we employed a within-participant design with repeated measures, and the collected data are ordinal, we used the non-parametric Friedman test. We use a Type I error α of 0.05; results are presented in Table 1. The Friedman test was followed by post-hoc pairwise Wilcoxon tests, using a Bonferroni-adjusted α level of .0033 (.05/15). This enables us to compare two generation types (AI vs. human), two genres (progressive metal vs. rock), and two AI selection methods (random vs. cherry-picked). Results are presented in Table 2. For question about AI-generated music (Q_7), we excluded responses (345 out of 1,920) where participants indicated prior song familiarity.

4.2 Thematic Analysis

We performed a thematic analysis [36] of answers to post-task questions to better understand the thought process of participants’ decisions during the study. Multiple themes were obtained from the responses, and results are ordered by number of codes within each theme (in parentheses next to each theme, indicating number of occurrences).

⁴ <https://www.songsterr.com/>

⁵ <https://drive.google.com/drive/folders/1-PVPXNCMu73ICfNf0qlwxzdVpNxrWIVL?usp=sharing>

⁶ Available at: <https://www.reddit.com/r/progmetal/>

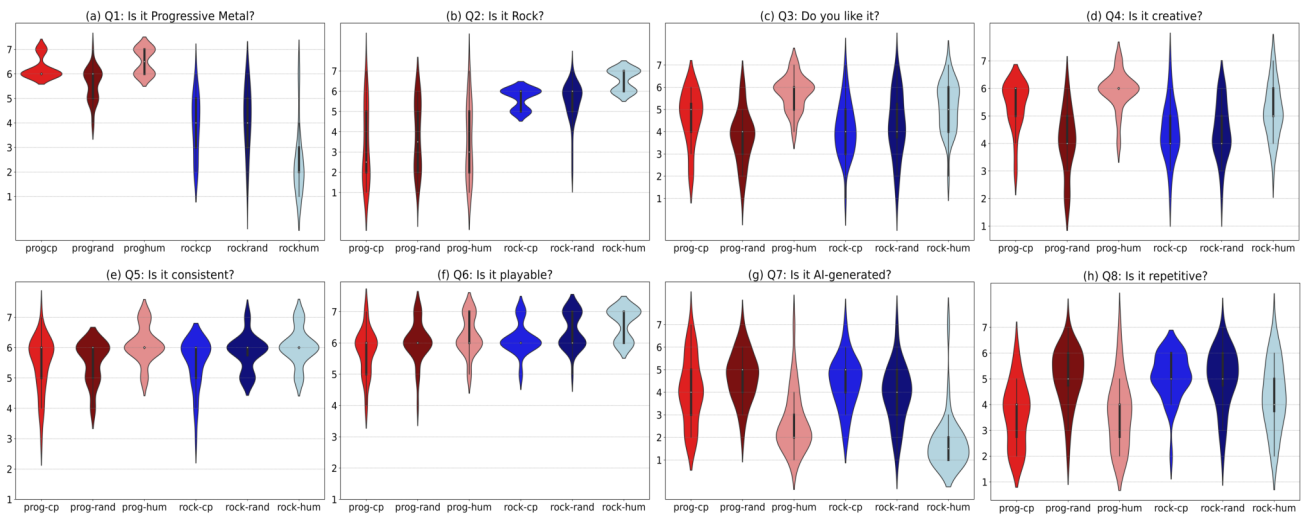


Figure 2. Violin plots of answers to Likert items for Q_1 to Q_8 , in plots (a) to (h), respectively, providing an estimation of the probability density function of the data. Horizontal axis represents the different groups of stimuli. Vertical axis reports the 7-point Likert ratings from 1 (Strongly Disagree) to 7 (Strongly Agree).

Question	Friedman Test Statistic	p-value	Significance
Q_1	$\chi^2(5) = 136.90$	8.14×10^{-28}	***
Q_2	$\chi^2(5) = 110.09$	3.90×10^{-22}	***
Q_3	$\chi^2(5) = 77.56$	2.72×10^{-15}	***
Q_4	$\chi^2(5) = 88.54$	1.36×10^{-17}	***
Q_5	$\chi^2(5) = 42.50$	4.67×10^{-8}	***
Q_6	$\chi^2(5) = 55.47$	1.04×10^{-10}	***
Q_7	$\chi^2(5) = 51.59$	6.53×10^{-10}	***
Q_8	$\chi^2(5) = 79.20$	1.23×10^{-15}	***

Table 1. Friedman test results investigating the effect of the creation method for each question (Q_1 to Q_8).

4.2.1 What features made you identify excerpts as progressive metal?

Complexity (40): A huge emphasis was put on the complexity of a composition, particularly the rhythmic but also the harmonic and melodic complexity. Uncommon and changing time signatures were mentioned by roughly half of the participants. The difficulty of playing a composition was also a very common answer.

Composition/style (38): Many compositional and stylistic elements were seen as particularly relevant to the genre, such as aggressiveness, speed and atmosphere. A sense of cohesion is important, with “clear and distinct ideas glued together”. The composition should be experimental, with creative rhythms, unique segments and interesting harmonic choices. Dissonant melodies, arpeggios, metal drum patterns and guitar specific techniques such as “chugs” are also deemed as important.

Instrumentation (7): Participants mention unique instruments and extended range guitars being particularly indicative of the genre.

4.2.2 What features made you identify excerpts as Rock?

Musical structure/composition (24): These excerpts were repetitive and had slower tempos, utilizing a question

and answer structure and accents on beats two and four. They were also generally soft and not particularly aggressive.

Simple/straightforward (23): The excerpts identified as rock were seen as simplistic, using simple drums, melodies, chord progressions and particularly 4/4 time signatures. These songs had straightforward grooves and generic solos.

Guitar techniques (14): Many techniques were seen as specific to the rock genre such as the use of the pentatonic scale, open chords, power chords and double stops. Participants noted a clear blues inspiration in the guitar playing.

Instrumentation (11): The rock genre was seen as guitar driven, with guitars and bass parts being separated. The drums were generally synchronized with the guitar and emphasized the hi-hat cymbals.

4.2.3 What made you identify excerpts as being composed using AI?

Something “off” about the composition (40): A major theme involved participants having some feeling of unease about the composition. Preference for human-composed music might be attributed to a perceived lack of qualities often associated with human creation, such as “soul” and creativity, or the inability to emulate human-like musical performance (playability). Many participants noted that some compositions lacked a sense of cohesiveness and consistency, or even sounded random, with odd note choices and bass lines which did not make sense. Participants also felt there was too much complexity, but also not enough of particular types of complexity (e.g. harmonic complexity).

Repetition (14): This theme specifically refers to negatively perceived repetition. Many described excerpts being overly repetitive or repeating “musically uninteresting or unsatisfying phrases”.

Uninteresting/simple (8): Participants describe boring

Group 1	Group 2	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Q_7	Q_8
rockrand	rockhum	1.10×10^{-4}	2.63×10^{-5}	0.26	0.15	4.29	1.70	0.44	0.55
rockcp	rockrand	6.42	1.15×10^1	8.25	1.32×10^1	3.00	1.70	1.30	1.36×10^1
rockcp	rockhum	3.96×10^{-4}	1.43×10^{-5}	1.78×10^{-2}	7.52×10^{-2}	0.35	2.30×10^{-2}	7.22×10^{-3}	0.19
progrand	proghum	5.79×10^{-5}	7.09	2.19×10^{-7}	2.46×10^{-7}	9.24×10^{-2}	5.16	2.94×10^{-6}	8.43×10^{-4}
progrand	rockcp	3.78×10^{-6}	2.17×10^{-4}	4.24	4.42	1.39×10^1	1.08×10^1	9.08	1.40×10^1
progrand	rockrand	1.86×10^{-4}	1.36×10^{-3}	1.79	3.60	2.32	0.97	0.48	1.26×10^1
progrand	rockhum	3.45×10^{-8}	7.61×10^{-9}	5.47×10^{-4}	4.80×10^{-3}	0.24	1.30×10^{-2}	1.40×10^{-3}	0.34
progcp	progrand	7.78×10^{-3}	3.56	3.69×10^{-2}	1.69×10^{-4}	1.44×10^1	1.65	2.10	3.62×10^{-4}
progcp	proghum	0.80	9.36	3.16×10^{-3}	0.25	0.14	0.22	5.60×10^{-3}	1.44×10^1
progcp	rockcp	9.29×10^{-10}	3.79×10^{-5}	0.67	3.16×10^{-3}	1.36×10^1	0.64	4.71	7.25×10^{-5}
progcp	rockrand	2.38×10^{-8}	1.98×10^{-4}	3.56	8.38×10^{-3}	2.56	1.78×10^{-2}	1.07×10^1	7.31×10^{-4}
progcp	rockhum	2.81×10^{-9}	1.63×10^{-8}	2.59	3.64	0.33	1.24×10^{-4}	0.34	0.60
proghum	rockcp	4.11×10^{-10}	2.72×10^{-5}	8.83×10^{-6}	3.92×10^{-6}	0.14	7.92	3.00×10^{-5}	1.98×10^{-4}
proghum	rockrand	3.50×10^{-9}	2.17×10^{-4}	5.31×10^{-4}	9.47×10^{-6}	2.32	5.69	5.06×10^{-3}	1.61×10^{-3}
proghum	rockhum	8.49×10^{-10}	1.93×10^{-8}	0.37	2.40×10^{-2}	1.08×10^1	0.25	3.00	0.65

Table 2. Pairwise post-hoc wilcoxon test results for each question. Each cell indicates p -value, while green cells indicates statistical significance (i.e. with Bonferroni correction $p < 0.0033$).

and generic riffs as well as simplistic and bland drum patterns.

Melody (7): A lack of interest or satisfaction with melodies was mentioned, specifically melodies that “run too long and miss their resolution” and “do not seem go anywhere”.

4.2.4 What made you identify excerpts as being composed by humans?

Well-composed (36): A sense of cohesion and consistency throughout the instrumentation and musical ideas was a popular reason for identifying an excerpt as human. Many also mentioned musical choices which feel deliberate and intentional. In general, compositions which felt natural, predictable, and emotionally satisfying were seen as more human.

Human-qualities (10): Certain qualities were perceived as more human, such as creativity, “soul”, and playability. The use of music theory, as well as breaking the rules of music theory were also mentioned.

5. DISCUSSION

5.1 H_1 : Human-composed music obtains better scores than AI-generated music

Human-composed progressive metal (**proghum**) was significantly preferred to all the other AI-generated groups (see Q_3 in Table 2). However, this could be due to participants all being progressive metal fans. Our findings suggest that a Turing test style approach may have limitations in evaluating generative models. While participants struggled to distinguish AI-generated from human-composed music, they still preferred the human compositions.

Participants’ evaluations used more negative language (e.g., ‘repetitive’) to describe AI compositions and more positive language for human compositions (see Sections 4.2.3 and 4.2.4). One might naturally expect significant differences in responses to the listening experiment between the AI-generated and human-composed music stim-

uli groups. While this is true for the randomly selected AI-generated progressive metal group (**progrand**), it does not hold for the rock groups as well as the cherry-picked AI-generated progressive metal (**progcp**) group. However, the violin plots (see Figure 2) do show the human-composed groups to generally have a better mean and smaller variance. Q_5 (“This composition is consistent”) and Q_6 (“This composition is playable”) saw no or few significant differences between stimuli groups. One of the negatively framed indicators of AI compositions was repetition. The cherry-picked and human-composed progressive metal groups were both significantly different to every group other than each other and the human-composed rock (**rockhum**) group in the responses to Q_8 (“This composition is repetitive”). Figure 2 also shows the responses in these groups trending toward not repetitive, while the others trend closer to repetitive. The **rockhum** group was not significantly different to any of the other groups, both AI and human-composed. While the level of repetition in AI-generated excerpts may be roughly similar to human-composed excerpts in their respective genre (with the exception of **progrand**), it is possible that repetition quality is different between AI and human-composed excerpts. Overall, we can conclude that the test shows strong evidence for H_1 in terms of preference, but not necessarily for the other dimensions.

5.2 H_2 : AI-generated and human-composed music can be distinguished

Figure 2 shows a large variance in the responses for Q_7 (“This composition was generated using AI”) for the AI generated stimuli groups, as well as numerous classification errors. The human stimuli groups also show classification errors, though less when compared to the AI stimuli groups. The **proghum** stimuli group was significantly different from both the cherry-picked rock (**rockcp**) and **progrand** groups. The **rockhum** group was only significantly different to the **progrand** group. The **progcp** and randomly selected AI-generated rock (**rockrand**) groups were not found to be significantly different to either of

the human-composed groups. Ultimately, this is evidence against H_2 , though there seems to be some dependence on the model used and the samples selected from that model.

5.3 H_3 : AI-generated music matches the genre used for model conditioning

The responses to Q_1 and Q_2 (see Figure 2 (a) and (b)) show a clear ability of participants to distinguish between the genres of progressive metal and rock, supporting H_3 . This is expected given that participants described the genres very differently to each other (see Sections 4.2.1 and 4.2.2). Of the progressive metal stimuli groups, only **progrand** and **proghum**, differed significantly, suggesting that at least the human curated AI-generated progressive metal stimuli (**progcp**) have features of similar quality to those of the human-generated group. The same cannot be said of the rock samples, though Figure 2 (b) shows the mean ratings in the rock groups are clearly higher than neutral (rated as 4), indicating that they identified the samples as rock. This may suggest that ProgGP excels in comparison to genre-CTRL in generating musical examples in its target genre. However, genre-CTRL, being trained on a wider range of styles (rock, punk, metal, classical, folk), could theoretically generate music in various styles, unlike ProgGP which is limited to its training genre.

5.4 H_4 : Cherry-picked AI-generated music is preferred over randomly chosen AI-generated music

At the surface level, the cherry-picked and randomly selected stimuli groups do not seem to have many differences. The groups in the rock genre have no significant differences between them in any of the questions, and the progressive metal groups only yield significant differences for Q_4 (“This composition is creative”). However, we observe that there are several questions with significant differences between the **progrand** and **proghum** groups, while the **progcp** and **proghum** groups only differ for Q_3 . This seems to indicate that the **progcp** stimuli have more in common with the human-composed excerpts than the randomly selected ones do in the tested features. Additionally, the difference seen in Q_4 between the AI-generated progressive metal groups concerns creativity, shown to be an indicator of human composition in Section 4.2.4. It is difficult to make any definite conclusions about H_4 given these results, but there seems to be some weak evidence for it in the progressive metal genre.

5.5 Study Limitations

The study is bounded by the number of participants (32) and an unbalanced gender distribution. Moreover, the stimuli were only 15 seconds long each, meaning that participants could not judge any long-term compositional features. Finally, the responses discussed in the study focus on compositional features and discard expressive and timbre-related aspects.

6. ETHICS OF MUSICAL DATA DIVERSITY

The broader topic of diversity within MIR is debated by Born in [37], in which the author highlights points such

as (1) the demographics within the field, (2) the nature of the music that is commonly researched, questions (3) the applicability of scientific work to a broader, more diverse, corpus of music, and (4) how to better stir MIR research towards a more encompassing music economy. Of particular relevance to this reflection is (2), closely linked to the concerns on musical data diversity. Despite efforts towards research concerning traditional, folk or ethnic music, MIR is still predominately based on the mainstream popular music that follows a “western” tradition [38]. Moreover, by referring to an ISMIR keynote by Seeger [39], Sturm et al. [40] point out that even within “western” music, there seems to be an emphasis on US pop music and European classical music. For automatic symbolic music generation, a closer look at the most commonly used datasets in the MIR community [41] suggests that styles such as western classical, pop and jazz music, often modelled using piano when dealing with single instrument systems, are a recurrent practice within the field. Following from these premises, it is important to first clarify that ProgGP is still grounded in the western music tradition, and to acknowledge this as a limitation given the musical data diversity concerns explained before. However, its musical style can be said to emphasize content that diverges from the *mainstream popular music* landscape. This begged the question: can the stylistic biases in the outputs from ProgGP contribute to a wider context of data diversity within MIR research? We argue that releasing training data for specific genres, exemplified by the release of data for fine-tuning ProgGP [1], is a step towards a more musically diverse MIR. This, along with publishing studies to understand underexplored genres and their challenges, and fostering interaction with stakeholders like the progressive metal community (as in this paper), can significantly contribute to this goal. We propose that guitar tablature can enhance musical diversity in MIR. Unlike MIDI, the dominant format, tablature excels at representing string instrument-specific expressive techniques, expanding the scope of representable music within the field.

7. CONCLUSION

We conducted a listening and reflective study which examined listeners perspectives on the quality of symbolic AI-generated compositions in the rock and progressive metal genres. The study provided both a subjective evaluation of recent Transformer-based music generation models and an exploration of listeners’ perceptions of AI and human compositions. We found that participants preferred human-composed music over AI-generated music, though they were generally not able to fully distinguish between AI- and human-composed music. Participants were able to distinguish between the two genres well. Cherry-picked examples in the progressive metal genre were rated similarly to the human-composed examples in several compositional metrics despite not being liked as much. With this methodology, we hope our work helps researchers better evaluate their generative models using a mixed methods approach through a listening and reflective study, as well as show the merit in increasing musical data diversity within MIR.

8. ACKNOWLEDGEMENTS

This work is supported by the EPSRC UKRI Centre for Doctoral Training in Artificial Intelligence and Music (Grant no. EP/S022694/1).

9. REFERENCES

- [1] J. Loth, P. Sarmiento, C. Carr, Z. Zukowski, and M. Barthet, “ProgGP: From GuitarPro Tablature Neural Generation To Progressive Metal Production,” in *The 16th International Symposium on Computer Music Multidisciplinary Research*, 2023.
- [2] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, and I. Sutskever, “Jukebox: A Generative Model for Music,” 2020. [Online]. Available: <https://github.com/openai/jukebox>.
- [3] A. Agostinelli, T. I. Denk, Z. Borsos, J. Engel, M. Verzett, A. Caillon, Q. Huang, A. Jansen, A. Roberts, M. Tagliasacchi *et al.*, “MusicLM: Generating Music from Text,” *arXiv preprint arXiv:2301.11325*, 2023.
- [4] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Défossez, “Simple and Controllable Music Generation,” in *37th Conference on Neural Information Processing Systems (NeurIPS)*, 2023.
- [5] Z. Evans, C. Carr, J. Taylor, S. H. Hawley, and J. Pons, “Fast Timing-Conditioned Latent Audio Diffusion,” in *Proceedings of the 41st International Conference on Machine Learning*, 2024.
- [6] J. Barnett, “The ethical implications of generative audio models: A systematic literature review,” in *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society (AIES ’23)*, 2023, p. 10, 1 figure.
- [7] D. Robinson, “An Exploration of the Various Compositional Approaches to Modern Progressive Metal,” MA Thesis, University of Huddersfield, 2019.
- [8] J. Wagner, *Mean Deviation: Four Decades of Progressive Heavy Metal*. Bazillion Points Books, 2010.
- [9] C. Hannan, “Hearing Form in Progressive Metal: Motivic Return, Genre Borrowing, and Sonata Form in Between the Buried and Me’s Parallax II’,” MA Thesis, Columbia University New York, 2019.
- [10] P. Sarmiento, “Perspectives on the Future for Sonic Writers,” *Journal of Science and Technology of the Arts*, vol. 13, no. 1, pp. 110–114, 2021.
- [11] N. Meade, N. Barreyre, S. C. Lowe, and S. Oore, “Exploring Conditioning for Generative Music Systems with Human-Interpretable Controls,” Tech. Rep., 2019.
- [12] B. L. Sturm, J. F. Santos, O. Ben-Tal, and I. Korshunova, “Music Transcription Modelling and Composition Using Deep Learning,” in *Proc. on the 1st Conf. on Computer Simulation of Musical Creativity*, 2016.
- [13] H. H. Tan and D. Herremans, “Music FaderNets: Controllable Music Generation Based On High-Level Features via Low-Level Feature Modelling,” in *Proc. of the 21st Int. Soc. for Music Information Retrieval Conf.*, Montréal, Canada, 2020, pp. 109–116.
- [14] H.-W. Dong and Y.-H. Yang, “Convolutional Generative Adversarial Networks with Binary Neurons for Polyphonic Music Generation,” in *Proc. of the 19th Int. Soc. for Music Information Retrieval Conf.*, Paris, France, 2018, pp. 190–198.
- [15] C.-Z. A. Huang, A. Vaswani, J. Uszkoreit, N. Shazeer, I. Simon, C. Hawthorne, A. M. Dai, M. D. Hoffman, M. Dinculescu, and D. Eck, “Music Transformer: Generating Music with Long-term Structure,” in *Proc. of the 7th Int. Conf. on Learning Representations*, New Orleans, LA, USA, 2019.
- [16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention Is All You Need,” in *Proc. of the 31st Conf. on Neural Information Processing Systems*, Long Beach, CA, USA, 2017.
- [17] C. Payne, “Musenet,” 2019, Last accessed: 12 Jun 2022. [Online]. Available: <https://openai.com/blog/musenet>
- [18] Y.-S. Huang and Y.-H. Yang, “Pop Music Transformer: Beat-based Modeling and Generation of Expressive Pop Piano Compositions,” in *Proc. of the 28th ACM Int. Conf. on Multimedia*, Seattle, WA, USA, 2020, pp. 1180–1188.
- [19] P. Sarmiento, A. Kumar, C. Carr, Z. Zukowski, M. Barthet, and Y.-H. Yang, “DadaGP: a Dataset of Tokenized GuitarPro Songs for Sequence Models,” in *Proc. of the 22nd Int. Soc. for Music Information Retrieval Conf.*, 2021, pp. 610–618.
- [20] P. Sarmiento, A. Kumar, Y.-H. Chen, C. Carr, Z. Zukowski, and M. Barthet, “GTR-CTRL: Instrument and Genre Conditioning for Guitar-Focused Music Generation with Transformers,” in *Proceedings of the EvoMUSART Conference*, 2023.
- [21] S. Adkins, P. Sarmiento, and M. Barthet, “LooperGP: A Loopable Sequence Model for Live Coding Performance using GuitarPro Tablature,” in *Proceedings of the EvoMUSART Conference*, 2023.
- [22] P. Sarmiento, A. Kumar, D. Xie, C. Carr, Z. Zukowski, and M. Barthet, “ShredGP: Guitarist Style-Conditioned Tablature Generation,” in *The 16th International Symposium on Computer Music Multidisciplinary Research*, Tokyo, Japan, 2023.
- [23] L.-C. Yang and A. Lerch, “On the Evaluation of Generative Models in Music,” *Neural Computing and Applications*, vol. 32, 05 2020.

- [24] K. Déguernel, H. Maruri-Aguilar, and B. L. T. Sturm, “Investigating the Relationship Between Liking and Belief in AI Authorship in the Context of Irish Traditional Music,” in *CREAI 2022 Workshop on Artificial Intelligence and Creativity*, 2022.
- [25] Y. Ren, J. He, X. Tan, T. Qin, Z. Zhao, and T.-Y. Liu, “PopMAG: Pop Music Accompaniment Generation,” in *Proceedings of the 28th ACM International Conference on Multimedia*. ACM, pp. 1198–1206. [Online]. Available: <https://dl.acm.org/doi/10.1145/3394171.3413721>
- [26] S. Dai, Z. Jin, C. Gomes, and R. B. Dannenberg, “Controllable Deep Melody Generation Via Hierarchical Music Structure Representation,” in *Proc. of the 22nd Int. Society for Music Information Retrieval Conf.*
- [27] H.-W. Dong, W.-Y. Hsiao, L.-C. Yang, and Y.-H. Yang, “MuseGAN: Multi-Track Sequential Generative Adversarial Networks for Symbolic Music Generation and Accompaniment,” in *Proc. of the 32nd AAAI Conf. on Artificial Intelligence (AAAI)*, 2018.
- [28] P. Lu, X. Tan, B. Yu, T. Qin, S. Zhao, and T.-Y. Liu, “MeloForm: Generating Melody with Musical Form based on Expert Systems and Neural Networks,” in *Proc. of the 23rd Int. Society for Music Information Retrieval Conf.*
- [29] Y.-K. Wu, C.-Y. Chiu, and Y.-H. Yang, “JukeDrummer: Conditional Beat-aware Audio-domain Drum Accompaniment Generation via Transformer VQ-VAE,” in *Proc. of the 23rd Int. Society for Music Information Retrieval Conf.* arXiv. [Online]. Available: <http://arxiv.org/abs/2210.06007>
- [30] S. Ji, X. Yang, and J. Luo, “A Survey on Deep Learning for Symbolic Music Generation: Representations, Algorithms, Evaluations, and Challenges,” *ACM Computing Surveys*, vol. 56, no. 1, pp. 7:1–7:39. [Online]. Available: <https://dl.acm.org/doi/10.1145/3597493>
- [31] L.-C. Yang, S.-Y. Chou, and Y.-H. Yang, “MIDINet: A Convolutional Generative Adversarial Network for Symbolic-domain Music Generation,” in *International Society for Music Information Retrieval Conference*, 2017, pp. 324–331.
- [32] G. Hadjeres, F. Pachet, and F. Nielsen, “DeepBach: A Steerable Model for Bach Chorales Generation,” in *International Conference on Machine Learning*, 2017, pp. 1362–1371.
- [33] S. Yang, C. N. Reed, E. Chew, and M. Barthet, “Examining Emotion Perception Agreement in Live Music Performance,” *IEEE Transactions on Affective Computing*, vol. 14, no. 2, pp. 1442–1460, 2023.
- [34] C. J. Steinmetz and J. D. Reiss, “pyloudnorm: A Simple yet Flexible Loudness Meter in Python,” in *150th AES Convention*, 2021.
- [35] D. Müllensiefen, B. Gingras, J. Musil, and L. Stewart, “The Musicality of Non-musicians: An Index for Assessing Musical Sophistication in the General Population,” *PloS one*, vol. 9, no. 2, p. e89642, 2014.
- [36] V. Braun and V. Clarke, “Using Thematic Analysis in Psychology,” *Qualitative research in psychology*, vol. 3, no. 2, pp. 77–101, 2006.
- [37] G. Born, “Diversifying MIR: Knowledge and Real-World Challenges, and New Interdisciplinary Futures,” *Transactions of the International Society for Music Information Retrieval*, vol. 3, no. 1, pp. 193–204, 2020.
- [38] E. Gómez, P. Herrera, and F. Gómez-Martín, “Computational Ethnomusicology: Perspectives and Challenges,” *Journal of New Music Research*, vol. 42, no. 2, pp. 111–112, 2013, special issue on ‘Computational Ethnomusicology’.
- [39] A. Seeger, “I found it, how can I use it? Dealing with the Ethical and Legal Constraints of Information Access,” in *International Society for Music Information Retrieval Conference*, 2003, keynote presentation.
- [40] B. L. T. Sturm, M. Iglesias, O. Ben-Tal, M. Miron, and E. Gómez, “Artificial Intelligence and Music: Open Questions of Copyright Law and Engineering Praxis,” *Arts*, vol. 8, no. 3, 2019. [Online]. Available: <https://www.mdpi.com/2076-0752/8/3/115>
- [41] H. W. Dong, K. Chen, J. McAuley, and T. Berg-Kirkpatrick, “MusPY: A Toolkit for Symbolic Music Generation,” in *Proc. of the 21st Int. Soc. for Music Information Retrieval*, 2020, pp. 101–108.

COMBINING AUDIO CONTROL AND STYLE TRANSFER USING LATENT DIFFUSION

Nils Demerlé

Philippe Esling

Guillaume Doras

David Genova

Ircam, STMS Lab, Sorbonne Université, CNRS

demerle@ircam.fr, esling@ircam.fr, doras@ircam.fr, genova@ircam.fr

ABSTRACT

Deep generative models are now able to synthesize high-quality audio signals, shifting the critical aspect in their development from audio quality to control capabilities. Although text-to-music generation is getting largely adopted by the general public, explicit control and example-based style transfer are more adequate modalities to capture the intents of artists and musicians.

In this paper, we aim to unify explicit control and style transfer within a single model by separating local and global information to capture musical structure and timbre respectively. To do so, we leverage the capabilities of diffusion autoencoders to extract semantic features, in order to build two representation spaces. We enforce disentanglement between those spaces using an adversarial criterion and a two-stage training strategy. Our resulting model can generate audio matching a timbre target, while specifying structure either with explicit controls or through another audio example. We evaluate our model on one-shot timbre transfer and MIDI-to-audio tasks on instrumental recordings and show that we outperform existing baselines in terms of audio quality and target fidelity. Furthermore, we show that our method can generate cover versions of complete musical pieces by transferring rhythmic and melodic content to the style of a target audio in a different genre.

1. INTRODUCTION

Deep generative models are now particularly successful at synthesising high-quality, realistic audio signals. Hence, the major impediment to their broader use in creative workflows is not their audio quality anymore, but rather how end-users can have complete control over the generation process. Following early works on unconditional generation [1, 2], multiple methods proposed to enable control by conditioning generation on semantic tags or audio-descriptors [3, 4]. However, such supervised approaches remain limited to the use of explicit descriptors and are constrained by their reliance on annotated datasets. The

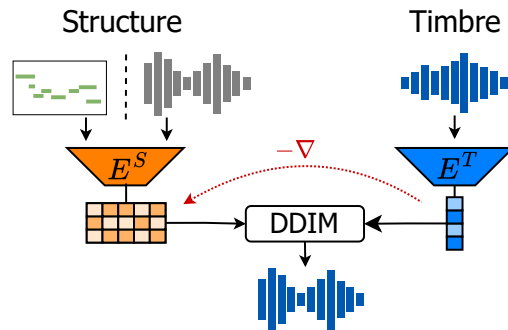


Figure 1. General overview of our method. We extract timbre and structure representations from waveform and/or MIDI inputs using encoders E_T and E_S respectively. Those representations condition a latent diffusion model, enabling both explicit and example-based control.

recent development of language models and representation learning led to impressive performance in text-conditioned generation, mainly relying on transformers [5, 6] or diffusion models [7–9]. However, concepts such as timbre, musical style or genres boundaries are usually elusive and highly subjective. Hence, text descriptions might remain limited to common sounds and insufficient to precisely capture musical intentions.

A parallel stream of research to alleviate those issues is to guide specific aspects of the generation process by providing audio examples. Most approaches in this audio-to-audio editing are focused on *timbre transfer*, where the timbre of a given sound is applied on the content of another. While some works can transfer any audio to the timbre of a given training set [10]; others achieve many-to-many *timbre transfer* but only between a small set of predefined instrument classes [11, 12]. *One-shot timbre transfer* between different instrument recordings have been achieved using Variational Autoencoders (VAE) [13, 14], but these models rely on a latent bottleneck to enforce disentanglement between timbre and pitch, which hampers their ability to generate high-quality audio on real-world data. More recently, a text-inversion technique was proposed to perform musical style transfer between arbitrary content and style examples [15], but it relies on a large pretrained text-to-music model and requires optimisation prior to each transfer, resulting in very slow inference.



In this paper, we aim to unify explicit control through audio descriptors or MIDI sequences and style transfer within a single model. To do so, we separate local, time-varying factors of variations and global information, capturing musical structure and timbre in two separate representation spaces. We slightly abuse the terms structure and timbre: by structure, we designate time-varying features, e.g. melody, loudness; by timbre, we designate global features such as actual timbre, but also style or genre. The principle of our method is depicted in Figure 1. Our approach is based on the recently proposed diffusion autoencoder [16], which trains a semantic encoder to condition a diffusion model, in order to achieve both high-quality generation while being able to extract and control high-level features from the data. We extend this approach by building separate representations for timbre and structure, while enforcing their disentanglement with an adversarial criterion combined with a two-stage training strategy. Our method can generate audio matching a timbre target, while specifying the musical structure either with explicit controls (such as MIDI data input) or an audio example. For computation efficiency, our diffusion model operates in the latent space of pretrained autoencoders, resulting in faster than real-time inference on GPU¹.

First, we benchmark our model on a *one-shot timbre transfer* tasks and demonstrate that our model improves upon existing baselines in terms of audio quality, timbre similarity as well as note onsets and pitch accuracy. On the same dataset, we show that our model can also generate audio from MIDI input and a target timbre example with performances superior to a state-of-the-art MIDI-to-audio baseline. Finally, we show that our method can be applied to complete musical pieces and generate cover versions of a track by transferring its rhythmic and melodic content to the style of a target audio in a different genre. We provide audio examples, additional experiments and source code on a supporting webpage²

2. BACKGROUND

2.1 Diffusion models

Diffusion models (DMs) are a family of generative models that learn to reverse a stochastic process that gradually adds noise to the input data. These models benefit from high-quality generation, stable training and conditioning abilities, which led to their widespread adoption in image [17] and audio generation [18].

Formally, we define a *forward process* $q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=1}^T q(\mathbf{x}_t|\mathbf{x}_{t-1})$, which is a Markov chain that increasingly adds noise to the data \mathbf{x}_0 by relying on the conditional distribution

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1 - \beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}), \quad (1)$$

where β_t are hyperparameters defining the noise levels at times $t \in \{0, T\}$, with $T \in \mathbb{N}$. We are inter-

ested in learning the reverse diffusion process $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$, from which we can iteratively denoise a random sample $\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$ to a data sample $\mathbf{x}_0 \sim p(\mathbf{x}_0)$. In a recent study, [19] made a connection between diffusion and denoising score matching [20], leading to a simplified formulation and improved experimental results. The authors show that the reverse process can be approximated by learning a denoising network ϵ_θ that predicts the noise $\epsilon \sim \mathcal{N}(\epsilon, \mathbf{0}, \mathbf{I})$ used to corrupt the data. This results in a simpler training objective

$$\min_{\theta \in \Theta} \mathbb{E}_{t, \mathbf{x}_0, \epsilon} [\|\epsilon_\theta(\sqrt{\alpha_t}\mathbf{x}_0 + \sqrt{1 - \alpha_t}\epsilon, t) - \epsilon\|], \quad (2)$$

where $\alpha_t = \prod_{s=1}^t (1 - \beta_s)$, and ϵ_θ is usually parametrized by a UNet [21].

2.2 Diffusion autoencoders

DMs naturally yield a series of latent variables $\mathbf{x}_{1:T}$ through their *forward process*. However, these stochastic variables built from increasingly adding noise do not capture much semantic information over the data. Although the more recent proposal of Denoising Diffusion Implicit Models (DDIMs) [22] extends the original diffusion formulation to a deterministic process allowing each data input to be mapped to a unique latent code \mathbf{x}_T , it still fails to extract and organise high-level features from the data. Diffusion autoencoders [16] alleviate this issue by employing a learnable encoder that compress the data to a semantic latent code $\mathbf{z}_{sem} = E_\phi(\mathbf{x}_0)$, which then conditions a diffusion decoder. The semantic encoder and the DDIM decoder are trained jointly, following the objective

$$\min_{\theta, \phi} \mathbb{E}_{t, \mathbf{x}_0, \epsilon} [\|\epsilon_\theta(\sqrt{\alpha_t}\mathbf{x}_0 + \sqrt{1 - \alpha_t}\epsilon, E_\phi(\mathbf{x}_0), t) - \epsilon\|] \quad (3)$$

On image applications, the authors show that the semantic code captures high-level attributes such as person identity, smile or presence of glasses, and can be used for downstream tasks such as conditional generation and attributes manipulation, while achieving state-of-art reconstruction. This approach was also successfully applied to audio [8], where the authors encode magnitude spectrograms into a semantic latent space, allowing to achieve high quality text-conditioned waveform generation.

2.3 Control in audio generation

A straightforward approach to extend unconditional generative models in order to provide instruments befitting artistic control is to introduce conditioning on explicit controls. The DDSF model [3] proposes explicit pitch and loudness conditioning, while FaderRave [4] extended explicit control to non-differentiable time-varying attributes, but both methods remain limited to explicit descriptors and annotated datasets. While recent text-to-music methods like MusicGen [5] and Music ControlNet [23] have incorporated melody conditioning capabilities, their expressiveness remains constrained by the need to define subjective timbre properties through text prompts. Li et al. [15]

¹ Experiments were conducted on a NVIDIA A5000 GPU

² <https://nilsdem.github.io/control-transfer-diffusion/>

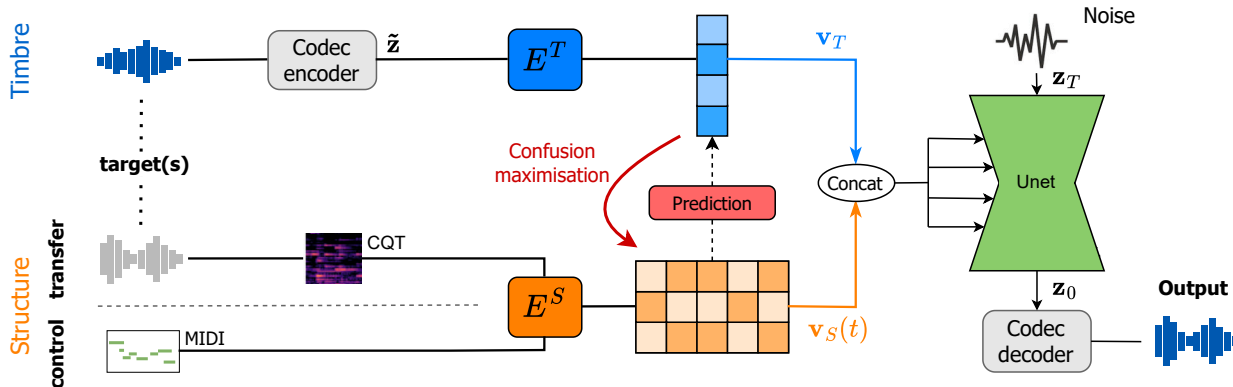


Figure 2. Detailed overview of our method. Input signal(s) are passed to structure and timbre encoders, which provides semantic encodings that are further disentangled through confusion maximization. These are used to condition a latent diffusion model to generate the output signal. Input signals are identical during training and but distinct at inference.

proposed to use text-inversion in order to extract pseudo-words that represent timbre directly from audio, but their method is computationally intensive and requires to perform an optimisation for each new timbre target. Timbre conditioning directly from a waveform example was also recently proposed the context of bass accompaniment generation [24].

2.4 Unsupervised disentanglement in sequential data

Many works proposed to model sequential data as a combination of local (time-variant) and global (time-invariant) factors of variation. Notably, the disentangled sequential autoencoder [25] relies on simple architecture biases and parameter tuning to obtain disentangled local and global latent variables. Following this work, multiple methods improved the learned representation by explicitly minimizing the mutual information between the two learned variables [26–28]. It was shown that disentanglement can be further improved with contrastive learning and domain-specific transformations that preserve local or global attributes [26, 29].

More specifically in audio generation, SS-VAE [14] employs a Vector-Quantized VAE to achieve disentanglement through compression on quantized structure latent codes, combined with timbre-preserving data augmentations. Luo et al. [13] follow a two-stage training strategy similar to ours, and improve disentanglement by enforcing the consistency of the global and local latent variables in style or content transfer. However, both disentanglement strategies degrades reconstruction accuracy, which on top of spectrogram inversion based synthesis leads in poor audio quality.

3. METHOD

Our approach is based on the assumption that musical audio samples can be seen as specific instances of a set of latent features that are separated between global features that capture style, and time-varying features that capture the local evolution of the signal. Although diffusion models are capable of high-quality conditional generation, they

are computationally expensive when dealing with high-dimensional data. Hence, we employ an invertible audio codec to first compress the audio into a low-dimensional latent space, onto which we can build an efficient generative model. We extend the DiffAE [16] architecture to two semantic encoders in order to extract separately timbre and structure features from input samples. To further disentangle the learned features and improve transfer as well as explicit control performances, we employ an adversarial training strategy during training. In this section, we detail our proposed model depicted in Figure 2.

3.1 Audio codec

We build our audio codec as a convolutional autoencoder based on the RAVE model [10] architecture, featuring the adversarial discriminator recently proposed in [30]. The model compresses audio waveforms \mathbf{x} into an invertible latent sequence $\mathbf{z} \in \mathbb{R}^{L \times D}$, where D and L are the embedding space and time dimensions respectively. On top of the reconstruction and adversarial training objectives of RAVE, we introduce a penalty on the latent codes $f(\mathbf{z}) = \max(0, |\mathbf{z}| - 1)$ to enforce that most latent codes are distributed between -1 and 1 .

3.2 Model structure

We extract a timbre representation $\mathbf{v}_T \in \mathbb{R}^{D_T}$ from an audio target using an encoder E_ϕ^T applied to the latent sequence \mathbf{z} obtained with our audio codec. For structure, we extract a temporal representation $\mathbf{v}_S \in \mathbb{R}^{L \times D_S}$ from either an audio input or an explicit control signal \mathbf{c} (such as a MIDI sequence), using an encoder E_ψ^S . In the case of an audio input, it would be natural to also infer it from the latent sequence \mathbf{z} . However, we found experimentally that it is particularly difficult to extract fine structure information from the highly-compressed representation \mathbf{z} . Hence, we instead infer structure from the Constant Q Transform (CQT) [31] of the target signal, which has been shown to be a well-suited representation for pitch extraction tasks [32]. For explicit control, the sequence \mathbf{c} is directly used

as input for E_ψ^S .

To generate audio, we sample a noise vector \mathbf{z}_T and decode it to a latent code \mathbf{z}_0 through latent diffusion conditioned on representations \mathbf{v}_S and \mathbf{v}_T . As diffusion formulation, we leverage the recent improvements introduced in [17] and parameterise our denoiser network D_θ to predict the data \mathbf{z}_0 instead of ϵ . E_ψ^S , E_ϕ^T and D_θ are trained end-to-end to minimise the following loss function

$$L_{diff} = \mathbb{E}_{t, \mathbf{z}_0, \epsilon} \|D_\theta(\sqrt{\alpha_t}\mathbf{z}_0 + \sqrt{1 - \alpha_t}\epsilon, \mathbf{v}_S, \mathbf{v}_T, t) - \mathbf{z}_0\| \quad (4)$$

We parameterise D_θ as a 1D convolutional UNet with residual blocks and self-attention layers. The two encoders share a similar architecture as the encoding half of the UNet, with the difference that the timbre encoder compresses the input temporally and applies average pooling on the time dimension of the last layer. We condition the UNet architecture on \mathbf{v}_S through concatenation with each block inputs. For the timbre vector \mathbf{v}_T we use Adaptive Group Normalisation (AdaGN) [33].

3.3 Style and content disentanglement

Although splitting the semantic content between two vectors that are constrained on their dimensions already encourages disentanglement between timbre and structure information, there is no theoretical guarantee regarding their separation. Furthermore, the appropriate feature dimensions are highly dependent on the task and dataset at hand. Hence, to enforce disentanglement without constraints on the dimensions, we introduce a two-stage training combined with an adversarial strategy. First, we freeze the structure encoder and train the model to build an adequate timbre representation. To avoid \mathbf{v}_T to encode all of the information required to reconstruct the target \mathbf{z} , we extract timbre from a different sample $\tilde{\mathbf{z}}$ coming from the same track, following the assumption that it shares the same timbre as \mathbf{z} but with a different structure.

In the next stage, we introduce a discriminator D_ζ that tries to predict \mathbf{v}_T from \mathbf{v}_S and is trained to minimise

$$L_D = \mathbb{E}_{\mathbf{v}_S, \mathbf{v}_T} [\|\mathbf{v}_T - D_\zeta(\mathbf{v}_S)\|]. \quad (5)$$

We train the discriminator alternatively with the encoders and denoising network, which try to minimize the following objective

$$L_{total} = L_{diff} - \gamma L_D, \quad (6)$$

where γ is an hyperparameter balancing between the reconstruction objective and disentanglement. Indeed, increasing L_D maximises the confusion of the timbre information in the structure space. This restrains the diffusion model from reconstructing \mathbf{z} solely from \mathbf{v}_S and enables independent control of structure and timbre at inference.

4. EXPERIMENTS

We aim to assess the ability of our model to generate high-quality audio samples that match characteristics of structure and timbre targets, with the structure being either

taken from an audio example or through an explicit control signal.

MIDI-to-audio For explicit structure control, we evaluate the capability of our model to generate audio from a MIDI score and a target recording for timbre. We compare it to a state-of-art baseline in MIDI-to-audio generation.

Timbre transfer We evaluate the efficiency of our disentanglement strategy on a task of *one-shot timbre transfer* between polyphonic mono-instrument recordings. In this case, we consider that structure designates the notes being played (in terms of onset timing, pitch and loudness), while timbre corresponds to the remaining characteristics of the sound. We evaluate our model by randomly sampling timbre and structure examples and evaluate audio quality, timbre similarity as well as note accuracy. We compare our model with two example-based timbre transfer methods on synthetic and real recordings.

4.1 Dataset

Synthetic Data The Synthesized Lakh Dataset (SLAKH) [34] was generated from the LAKH MIDI collection using professional-grade sample-based virtual instruments. Synthesis parameters as well as audio effects settings were randomly chosen resulting in a very diverse set of timbres. We retain only the individual stems of non-percussive instruments, resulting in 400 hours of audio.

Real Data To the best of our knowledge there is no multi-instrumental dataset of real recordings that contain a very large number of hours of audio. Hence, we combined the following three datasets to conduct our experiments :

- **MaestroV2** : Maestro [35] is a piano dataset recorded on Disklavier pianos, capturing both audio and notes played, resulting in approximately 200hours of annotated piano recordings.
- **GuitarSet** : Guitarset [36] is a collection of live guitar performances with solos and accompaniment from various genres and play styles, with a total audio duration of 6 hours.
- **URMP** : The URMP dataset [37] is composed of pieces played by a large variety of classical instruments. For each piece we retain the mono-instrumental recordings, resulting in approximately 4 hours of audio.

As the GuitarSet and URMP are low sample-size datasets, we add synthetic data stems from SLAKH to the training set to facilitate learning. Furthermore, as the different datasets are greatly imbalanced in terms of sample size, we apply a sampling strategy to even the model performance on each dataset. Following [38], if n_i is the number of samples in a given dataset, we draw examples from this dataset during training with probability $(n_i / \sum_j n_j)^{0.3}$.

4.2 Evaluation metrics

We aim to evaluate how our method is able to match the timbre and structure targets characteristics, while maintaining high-quality audio.

		Quality (FAD) ↓		Timbre similarity ↑		Onset F1 score ↑	
		Rec.	Transfer	Rec.	Transfer	Rec.	Transfer
MIDI-to-audio	Spectrogram diffusion [38]	3.46	-	0.76	-	0.32	-
	Ours w/o E_S	1.22	1.41	0.87	0.77	0.40	0.38
	Ours	0.88	1.06	0.89	0.83	0.36	0.23
Timbre transfer	SS-VAE [14]	2.83	3.23	0.75	0.69	0.29	0.15
	Music Style Transfer [15]	2.95	2.77	0.84	0.60	0.22	0.17
	Ours w/o adversarial - $D_S = 4$	0.95	1.75	0.91	0.75	0.36	0.23
	Ours w/o adversarial - $D_S = 8$	0.95	1.65	0.91	0.73	0.36	0.26
	Ours	1.13	1.42	0.91	0.82	0.36	0.23

Table 1. Experimental results in terms of audio quality, timbre similarity and note accuracy on the SLAKH dataset, for MIDI-to-audio generation (**up**) and timbre transfer (**down**). "Rec." corresponds to samples generated from identical structure and timbre targets, while "Transfer" designates randomly chosen timbre targets.

Audio quality We rely on the widely used *Frechet Audio Distance* (FAD) [39] to evaluate how the generated audio distribution matches the dataset distribution, both for reconstructed and transferred samples. We use the available reference implementation of FAD³ and use VGGish [40] embeddings of the samples to compute the distance

Timbre Similarity To evaluate timbre similarity we employ the metric proposed in the SS-VAE baseline [14]. It relies on a triplet network trained to predict if samples are played by the same instrument based on Mel-frequency cepstral coefficients 2-13. We use their implementation and train the metric on the *mixing-secrets*⁴ dataset.

Structure To evaluate if our model is able to reproduce the notes of the structure target, we employ a transcription model [41] and compare its output to the ground-truth MIDI data. As metric, we use Onset F1 score from *mir-eval*, where two notes are considered identical if they have identical pitch and onsets within $\pm 50ms$ of each other.

4.3 Baselines

For the timbre transfer experiments, we compare our method to SS-VAE [14] and Music Style Transfer [15] presented in Section 2. We train both models on the real and synthetic datasets, using the official implementation. For explicit control, we evaluate our method against a MIDI-to-audio model [38] that was also trained on the SLAKH dataset. We use the *small* configuration of the publicly available pretrained model, as larger models do not fit on our NVIDIA A5000 GPU.

4.4 Training details

We start by training our audio codec for 1M steps before training our diffusion model for 500k steps, with an initial timbre learning stage of 100k steps. The overall training takes one day on NVIDIA A5000 GPU. For all experiments we rely on the AdamW optimizer [42] with a constant learning rate of $1e^{-4}$ and a batch size of 48. For inference we use the deterministic sampler proposed in [17] with 40 diffusion steps.

³ <https://github.com/gudgud96/frechet-audio-distance/tree/main>

⁴ <https://www.cambridge-mt.com/ms/mtk/>

5. RESULTS

5.1 MIDI-to-audio

First, we evaluate our model performance in MIDI-to-audio generation in terms of audio quality, timbre similarity and Onset F1 score. We detail our results in Table 1 for reconstruction and transfer setups, where the target timbre corresponds to either the instrument of the MIDI sequence or a different sample. In both cases, we obtain higher similarity, as our dedicated timbre embedding captures timbre much more precisely than simple label conditioning on instrument categories. Interestingly, we also obtain better F1 scores, although we did not design our model specifically for MIDI inputs as opposed to [38] where authors employ a dedicated note sequence embedding strategy.

To assess the benefit of our disentanglement strategy, we experiment with bypassing the structure encoder and directly conditioning the UNet on the MIDI sequence (*Ours w/o E_S* entry in Table 1). This results in overall better Onset F1 score, but degrades timbre similarity and FAD. This demonstrates that our disentanglement strategy improves the capability of the model to precisely render the timbre of the target recording. As described in Section 4.2, the Onset F1 score characterises the difference between the generated notes and the input MIDI sequence. The lower accuracy obtained with our full model in the transfer setup can be explained by the fact that some MIDI sequence are not plausible scores for some target instruments (such as playing chords with a flute). Through the disentangled structure encoding, our model is capable of adapting the input MIDI sequence to the range and capabilities of the target instrument, which results in more realistic sounding samples. We encourage the reader to listen to the examples on our supporting webpage that support this statement.

5.2 Timbre transfer

Synthetic data Here, we first evaluate *timbre transfer* on synthetic data and display the results in Table 1. The two baselines appear to provide low audio quality and timbre similarity, and both methods obtain lower Onset F1 scores indicating that they are not able to adequately control structure and timbre independently. Our method improve upon the baselines on all three evaluated aspects, and is inter-

estingly able to reach a comparable performance as in the explicitly conditioned MIDI-to-audio setup.

We also performed an ablation study, by evaluating the effect of applying an information bottleneck on the structure latent space instead of using our adversarial strategy. As mentioned in Section 3.3, the model is able to transfer timbre when D_S is small but achieves a low F1 score. Increasing the latent dimensions improves structure fidelity at the cost of degrading timbre similarity. Using our disentanglement strategy, we are able to employ a 32-dimensional latent space and achieve higher timbre similarity with a slight decrease in note accuracy. Although multiple definitions of timbre transfer are possible, we argue that the most convincing timbre transfer do not necessarily imply a perfect note structure F1 score. In the case of a transfer between monophonic instruments playing in the same pitch range, we can expect all the notes from one recording to be transferred to the other. However, when performing transfer between recordings with very distinct timbre such as a piano playing in its high range and a bass, an interesting transfer would rather be the bass playing the main melodic line a few octaves lower than the piano, which would result in a low F1 score. The improvement in terms of FAD between the distribution of transferred samples and the original data obtained with our disentanglement strategy supports that our method generates more realistic transfers, as an instrument playing notes outside its usual range would be considered as out-of-distribution.

	FAD ↓	Timbre ↑	F1 ↑
SS-VAE [14]	9.26	0.58	0.19
Music Style Tr. [15]	10.2	0.57	0.17
Ours w/o adv.	2.14	0.81	0.43
Ours	1.36	0.88	0.28

Table 2. Experimental results for timbre transfer on real instruments in terms of FAD, timbre similarity and onset F1 score. Metrics are averaged between the three datasets.

Real data. We present our results for transfer between real instrumental recordings in Table 5.2. Our model improves even further on the existing baselines for which real instruments timbre seems particularly challenging. Even without adversarial regularisation, our model obtains better FAD, timbre similarity and note accuracy. Our disentanglement strategy further improves timbre match, although the relative decrease in note accuracy appears to be greater than on synthetic data. We believe this is mainly due to a necessary simplification of note structure when transferring complex piano recordings to the mainly monophonic URMP instruments. The improvement of transfer quality captured by the FAD and timbre similarity supports this interpretation.

Qualitatively, we found that on top of generating realistic samples with the appropriate structure, the model is also able to add characteristic sound artefacts of the target instrument such as fret or hammer noises, as well as matching precise acoustic features of the original recording such as reverb or background noise.

6. COMPLETE MUSIC STYLE TRANSFER

	FAD ↓	Cover (%) ↑	Genre ↑
MusicGen [5]	-	37.6	0.48
Ours w/o adv.	3.99	48.5	0.44
Ours	3.31	52.2	0.55

Table 3. Style transfer results on musical pieces, evaluated through FAD, cover identification and genre classification.

Finally, we apply our model to the task of creating cover versions of a song that match the style of an example in a different genre. We rely on an in-house dataset of 200 hours of jazz, dub, lofi hip-hop and rock. We use the same model architecture to extract structure from the original track and timbre from the cover targets, with two minor modifications. First, we introduce temporal compression in the structure encoder to avoid v_T to capture information that is too precisely located in time. Second, we condition the UNet on a BPM time series to help it generating coherent rhythms. Without those modifications, the rhythmic elements from timbre and structure targets are conflicting with each other, resulting in somewhat chaotic generations. We evaluate our model using the cover detection algorithm proposed in [43], which outputs a cover probability based on melodic and harmonic similarities between tracks. To assess style transfer, we rely on the text and audio joint-embedding model CLAP [44] and classify genre based on the cosine similarity between the audio and genre label embeddings. We compare our method to MusicGen [5], a text-to-music generation model with audio-based melody conditioning. We derive an input prompt from the target genre and use the structure target for melody.

Our model without regularisation obtains a better cover identification than MusicGen, and our disentanglement strategy further improves transfer resulting in higher genre accuracy. Qualitatively, MusicGen seems to only extract the main melodic idea from the structure audio, whereas our method is able to capture most of the harmonic and melodic content. Furthermore, as our model extract style directly from audio rather than through a text prompt, it transfers the different structural elements towards the instruments actually present in the timbre target rather than just the typical instruments of the genre.

7. CONCLUSION

We presented a simple method to learn disentangled timbre and structure representations. To the best of our knowledge, this is the first model capable of generating realistic, high-quality audio through transfer and MIDI rendering. We leave for future works improvements on the trade-off between reconstruction and disentanglement and applications to more complex musical datasets. Furthermore, we aim to initiate a reflection on the characterisation of the elusive concept of musical style transfer, which we believe to be an exciting stream of research towards a broader use of deep generative models in artistic work-flows.

8. REFERENCES

- [1] S. Mehri, K. Kumar, I. Gulrajani, R. Kumar, S. Jain, J. Sotelo, A. Courville, and Y. Bengio, “SAMPLERNN: An unconditional end-to-end neural audio generation model,” *arXiv preprint arXiv:1612.07837*, 2016.
- [2] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, and I. Sutskever, “Jukebox: A generative model for music,” *arXiv preprint arXiv:2005.00341*, 2020.
- [3] J. Engel, L. Hantrakul, C. Gu, and A. Roberts, “Ddsp: Differentiable digital signal processing,” *arXiv preprint arXiv:2001.04643*, 2020.
- [4] N. Devis, N. Demerlé, S. Nabi, D. Genova, and P. Esling, “Continuous descriptor-based control for deep audio synthesis,” in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.
- [5] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Défossez, “Simple and controllable music generation,” *Advances in Neural Information Processing Systems*, vol. 36, 2024.
- [6] A. Agostinelli, T. I. Denk, Z. Borsos, J. Engel, M. Verzett, A. Caillon, Q. Huang, A. Jansen, A. Roberts, M. Tagliasacchi *et al.*, “MusicLM: Generating music from text,” *arXiv preprint arXiv:2301.11325*, 2023.
- [7] Q. Huang, D. S. Park, T. Wang, T. I. Denk, A. Ly, N. Chen, Z. Zhang, Z. Zhang, J. Yu, C. Frank *et al.*, “Noise2music: Text-conditioned music generation with diffusion models,” *arXiv preprint arXiv:2302.03917*, 2023.
- [8] F. Schneider, O. Kamal, Z. Jin, and B. Schölkopf, “Mousai: Text-to-music generation with long-context latent diffusion,” *arXiv preprint arXiv:2301.11757*, 2023.
- [9] Z. Evans, C. Carr, J. Taylor, S. H. Hawley, and J. Pons, “Fast timing-conditioned latent audio diffusion,” *arXiv preprint arXiv:2402.04825*, 2024.
- [10] A. Caillon and P. Esling, “Rave: A variational autoencoder for fast and high-quality neural audio synthesis,” *arXiv preprint arXiv:2111.05011*, 2021.
- [11] N. Mor, L. Wolf, A. Polyak, and Y. Taigman, “A universal music translation network,” *arXiv preprint arXiv:1805.07848*, 2018.
- [12] A. Bitton, P. Esling, and T. Harada, “Vector-quantized timbre representation,” *arXiv preprint arXiv:2007.06349*, 2020.
- [13] Y.-J. Luo, S. Ewert, and S. Dixon, “Towards robust unsupervised disentanglement of sequential data—a case study using music audio,” *arXiv preprint arXiv:2205.05871*, 2022.
- [14] O. Cífka, A. Ozerov, U. Şimşekli, and G. Richard, “Self-supervised vq-vae for one-shot music style transfer,” in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 96–100.
- [15] S. Li, Y. Zhang, F. Tang, C. Ma, W. Dong, and C. Xu, “Music style transfer with time-varying inversion of diffusion models,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 1, 2024, pp. 547–555.
- [16] K. Preechakul, N. Chatthee, S. Wizadwongsa, and S. Suwajanakorn, “Diffusion autoencoders: Toward a meaningful and decodable representation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 10 619–10 629.
- [17] T. Karras, M. Aittala, T. Aila, and S. Laine, “Elucidating the design space of diffusion-based generative models,” *Advances in Neural Information Processing Systems*, vol. 35, pp. 26 565–26 577, 2022.
- [18] H. Liu, Z. Chen, Y. Yuan, X. Mei, X. Liu, D. Mandic, W. Wang, and M. D. Plumbley, “Audioldm: Text-to-audio generation with latent diffusion models,” *arXiv preprint arXiv:2301.12503*, 2023.
- [19] J. Ho, A. Jain, and P. Abbeel, “Denoising diffusion probabilistic models,” *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.
- [20] Y. Song and S. Ermon, “Generative modeling by estimating gradients of the data distribution,” *Advances in neural information processing systems*, vol. 32, 2019.
- [21] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *Medical image computing and computer-assisted intervention—MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*. Springer, 2015, pp. 234–241.
- [22] J. Song, C. Meng, and S. Ermon, “Denoising diffusion implicit models,” *arXiv preprint arXiv:2010.02502*, 2020.
- [23] S.-L. Wu, C. Donahue, S. Watanabe, and N. J. Bryan, “Music controlnet: Multiple time-varying controls for music generation,” *arXiv preprint arXiv:2311.07069*, 2023.
- [24] M. Pasini, M. Grachten, and S. Lattner, “Bass accompaniment generation via latent diffusion,” in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 1166–1170.
- [25] Y. Li and S. Mandt, “Disentangled sequential autoencoder,” *arXiv preprint arXiv:1803.02991*, 2018.

- [26] J. Bai, W. Wang, and C. P. Gomes, “Contrastively disentangled sequential variational autoencoder,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 10 105–10 118, 2021.
- [27] Y. Zhu, M. R. Min, A. Kadav, and H. P. Graf, “S3vae: Self-supervised sequential vae for representation disentanglement and data generation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6538–6547.
- [28] J. Han, M. R. Min, L. Han, L. E. Li, and X. Zhang, “Disentangled recurrent wasserstein autoencoder,” *arXiv preprint arXiv:2101.07496*, 2021.
- [29] T. Haga, H. Kera, and K. Kawamoto, “Sequential variational autoencoder with adversarial classifier for video disentanglement,” *Sensors*, vol. 23, no. 5, p. 2515, 2023.
- [30] R. Kumar, P. Seetharaman, A. Luebs, I. Kumar, and K. Kumar, “High-fidelity audio compression with improved rvqgan,” *Advances in Neural Information Processing Systems*, vol. 36, 2024.
- [31] J. C. Brown, “Calculation of a constant q spectral transform,” *The Journal of the Acoustical Society of America*, vol. 89, no. 1, pp. 425–434, 1991.
- [32] R. M. Bittner, B. McFee, J. Salamon, P. Li, and J. P. Bello, “Deep salience representations for f0 estimation in polyphonic music.” in *ISMIR*, 2017, pp. 63–70.
- [33] P. Dhariwal and A. Nichol, “Diffusion models beat gans on image synthesis,” *Advances in neural information processing systems*, vol. 34, pp. 8780–8794, 2021.
- [34] E. Manilow, G. Wichern, P. Seetharaman, and J. Le Roux, “Cutting music source separation some slakh: A dataset to study the impact of training data quality and quantity,” in *2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2019, pp. 45–49.
- [35] C. Hawthorne, A. Stasyuk, A. Roberts, I. Simon, C.-Z. A. Huang, S. Dieleman, E. Elsen, J. Engel, and D. Eck, “Enabling factorized piano music modeling and generation with the maestro dataset,” *arXiv preprint arXiv:1810.12247*, 2018.
- [36] Q. Xi, R. M. Bittner, J. Pauwels, X. Ye, and J. P. Bello, “Guitarset: A dataset for guitar transcription.” in *ISMIR*, 2018, pp. 453–460.
- [37] B. Li, X. Liu, K. Dinesh, Z. Duan, and G. Sharma, “Creating a multitrack classical music performance dataset for multimodal music analysis: Challenges, insights, and applications,” *IEEE Transactions on Multimedia*, vol. 21, no. 2, pp. 522–535, 2018.
- [38] C. Hawthorne, I. Simon, A. Roberts, N. Zeghidour, J. Gardner, E. Manilow, and J. Engel, “Multi-instrument music synthesis with spectrogram diffusion,” *arXiv preprint arXiv:2206.05408*, 2022.
- [39] K. Kilgour, M. Zuluaga, D. Roblek, and M. Sharifi, “Fr\`echet audio distance: A metric for evaluating music enhancement algorithms,” *arXiv preprint arXiv:1812.08466*, 2018.
- [40] S. Hershey, S. Chaudhuri, D. P. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold *et al.*, “Cnn architectures for large-scale audio classification,” in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 131–135.
- [41] R. M. Bittner, J. J. Bosch, D. Rubinstein, G. Meseguer-Brocal, and S. Ewert, “A lightweight instrument-agnostic model for polyphonic note transcription and multipitch estimation,” in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 781–785.
- [42] I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” *arXiv preprint arXiv:1711.05101*, 2017.
- [43] M. Abrassart and G. Doras, “And what if two musical versions don’t share melody, harmony, rhythm, or lyrics?” in *ISMIR 2022*, 2022.
- [44] Y. Wu, K. Chen, T. Zhang, Y. Hui, T. Berg-Kirkpatrick, and S. Dubnov, “Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation,” in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

COMPUTATIONAL ANALYSIS OF YAREDAWI YEZEMA SILT IN ETHIOPIAN ORTHODOX TEWAHEDO CHURCH CHANTS

Mequanent Argaw Muluneh^{1,2} Yan-Tsung Peng² Li Su¹

¹ Institute of Information Science, Academia Sinica, Taipei, Taiwan

² Department of Computer Science, National Chengchi University, Taipei, Taiwan

mequanenta@gmail.com, ytpeng@cs.nccu.edu.tw, lisu@iis.sinica.edu.tw

ABSTRACT

Despite its musicological, cultural, and religious significance, the Ethiopian Orthodox Tewahedo Church (EOTC) chant is relatively underrepresented in music research. Historical records, including manuscripts, research papers, and oral traditions, confirm Saint Yared's establishment of three canonical EOTC chanting modes during the 6th century. This paper attempts to investigate the EOTC chants using music information retrieval (MIR) techniques. Among the research questions regarding the analysis and understanding of EOTC chants, *Yaredawi YeZema Silt*, namely the mode of chanting adhering to Saint Yared's standards, is of primary importance. Therefore, we consider the task of *Yaredawi YeZema Silt* classification in EOTC chants by introducing a new dataset and showcasing a series of classification experiments for this task. Results show that using the distribution of stabilized pitch contours as the feature representation on a simple neural-network-based classifier becomes an effective solution. The musicological implications and insights of such results are further discussed through a comparative study with the previous ethnomusicology literature on EOTC chants. By making this dataset publicly accessible, our aim is to promote future exploration and analysis of EOTC chants and highlight potential directions for further research, thereby fostering a deeper understanding and preservation of this unique spiritual and cultural heritage.

1. INTRODUCTION

The Ethiopian Orthodox Tewahedo Church (EOTC) chants hold immense cultural and religious significance in Ethiopia, yet they are largely overlooked [1].¹ The EOTC chant is believed to have originated with Saint Yared (505–571), who composed the three EOTC chanting modes

¹ The Eritrean Orthodox Tewahedo Church, which separated from the EOTC administration system a few decades ago, also utilizes these chants. We acknowledge its important role in preserving this sacred form of church music.

(*YeZema Siltoch* in Amharic language),² namely *Ge'ez*,³ *Ezil* and *Araray*. Saint Yared's pioneering musical compositions, liturgical chants, and associated dance movements had a significant impact on Ethiopian sacred music tradition [2]. The *Debterawoch* (also called *Merigeta-woch*), who are the expert musicians and heirs of Saint Yared, play a crucial role in the transmission and performance of the sacred music [1]. Ethiopian sacred music has been preserved through oral and written traditions, with written documents supporting and reinforcing the ongoing oral traditions [3]. The significance of the EOTC chants in Ethiopian culture and worldwide is evident through the two major spiritual mass celebrations that have been recognized by UNESCO as intangible cultural heritages: the Commemoration Feast of the Finding of the True Holy Cross of Christ (in 2013) and the Ethiopian Epiphany (in 2019).⁴ These two celebrations, primarily accompanied by the EOTC chants, are among the five intangible cultural heritages from Ethiopia registered by UNESCO.

Despite its long history and development, the research of EOTC chants was quite rare. Among them, a renowned ethnomusicological work from Western academia was by Shelemay et al. [3, 4], based on the analysis of a series of EOTC chants collected in Addis Ababa, 1975. They discussed the oral and written tradition of EOTC chants, the EOTC chant music notation system, and further the definition of the three chanting modes, specifically the pitch sets used in each of the modes. It should be noted that in this work, all the recordings were transcribed and analyzed by ear. As stated in the paper, the analysis, for a limited number of recordings, was sometimes challenging when transcribing the non-Western music scales. With no indigenous classification of their pitch materials [3], *YeZema Siltoch* remains a primary research topic in the music theory of EOTC chants.

This paper is a study on *YeZema Siltoch* of the EOTC chants from computational perspectives. Our contributions in this paper are three-fold. First, we propose a new dataset for *YeZema Silt* classification and analysis. Second, we

² *Siltoch* is the plural form of *silt*. For simplicity, the Amharic phrase *YeZema Silt* and the English phrase chanting mode will be used interchangeably throughout this paper.

³ The term *Ge'ez* holds various connotations depending on context; here, it denotes one of the three chanting styles. Conversely, it also refers to the language and may have other applications.

⁴ <https://ich.unesco.org/en/state/ethiopia-ET?info=elements-on-the-lists>



benchmark the *YeZema Silt* classification on the dataset using neural network classifiers with a number of features, primarily the pitch contour features which have been verified useful in analyzing various kinds of music [5–11]. Third, we perform a comparative study with [3,4] to echo, and to revise their statements as well: while the pitch sets used in *Ezil* and *Araray* was regarded as the same [3], our numerical results indicate notable difference in between them. In the rest of this paper, we will have a background introduction of EOTC chants in Section 2. The proposed dataset, benchmarks and the comparative study will be in Sections 3, 4, and 5, respectively. Conclusion and future works will be given in Section 6.

2. BACKGROUND OF EOTC CHANTS

2.1 Features and Performance Traditions

The spiritual schools of the EOTC have several departments, locally known as *Guba'e bet*. These departments include *Nibab-bet* (reading practice), *Zema-bet* (introductory to advanced level offices chanting), *Qidase-bet* (or *Kidase-bet*, liturgical chants), *Qine-bet*⁵ (or *Kine-bet*, poetry), *Aquaquam-bet* (or *Akuakuam-bet*, advanced chanting with accompaniments), and *YeMetsahift Tirguame-bet* (exegesis of scriptures). The knowledge and skills acquired from each *Guba'e bet* are crucial for understanding the chants. Each *Guba'e bet*, which focused on chanting, has two or more slightly different vocal and performance styles [12]. For example, *Zema-bet* has *Bethlehem*, *Achabir*, *Qoma*, and *Tegulet*, and *Qidase-bet* has *Sellekula* and *Debre Abay*. These nominations are based on the names of places where the center of excellence, that approves a senior student to be a teacher, is located. Such *Guba'e bet*, for example, *Bethlehem* has a slightly different vocal style, ornamentation, and notation complexity compared to *Qoma*, and it also has its own swaying and religious dancing tradition with its own drumbeat.

The EOTC chants incorporate monophonic, antiphonal, and choral ritual performances. Our dataset is derived from *Qidase-bet*, which primarily focuses on monophonic and antiphonal ritual performance components without accompaniments. In contrast, *Aquaquam-bet* emphasizes religious dance and movements, primarily choral with some monophonic and antiphonal components. It is accompanied by prayer staffs known as *mequamia*, drums, and sistrums [12, 13]. The content of the chants - the text, whether poetic or unpoetic, is directly or indirectly based on the Holy Bible. The lyrics primarily employ Ge'ez (ግዕዝ), an ancient Semitic language with a distinct script known as Fidäl. These chants play an essential role in the religious practices of nearly 43.5% of the country's population, or over 32 million Orthodox Tewahedo Christians, according to the 2007 national census [14].⁶

The social groups involved in the chants include priests, deacons, and laypeople who attend the service hours. Tra-

ditionally, the chants were transmitted orally, with singers memorizing a repertoire of phrases and melodies to perform during liturgical celebrations. Several decades ago, chant manuscripts were handwritten on parchment, which refers to processed goat or sheep skins. Even today, some scholars adhere to this practice to uphold the church's cultural traditions. However, in recent decades, transmission has been supported by printed manuscripts for training along with oral traditions for actual performance.

Despite its rich heritage, the tradition of EOTC chants faces significant challenges. Many training centers are closing down due to absence of government support, insufficient community support for students [12], and the dominance of modern education since the 20th century. Despite the contributions of printing and recording advancements, the computational contribution to the Ge'ez language and the chants remains underdeveloped. Except for a few works on MIR [13] and music generation [15], computational research on the EOTC chants is not as developed as it is for some other secular music. These issues highlight the need for more research on the EOTC chants. Our research aims to contribute to MIR-related tasks on the EOTC chants, addressing this gap.

2.2 YeZema Siltoch - Chanting Modes

The EOTC chants encompass three primary *YeZema Siltoch* (modes): *Ge'ez*, *Ezil*, and *Araray*. They are typically employed sequentially or intermixed, sometimes aligning with the church calendar's seasons. Notably, during fasting periods, the *Ge'ez* and *Araray* modes predominate, while the *Ezil* mode mostly reserved for holidays. These modes serve as conduits for conveying distinct emotions and seasonal themes within the EOTC chants [1].

- **Ge'ez:** Characterized by a foundational, low tone, *Ge'ez* chanting evokes a sense of despondency and solemnity. Rendered in a relaxed, subdued manner devoid of rhythmic constraints, it encapsulates feelings of despair, disappointment, and sorrow [1]. In [3], the Ge'ez mode is interpreted as a *chain of third* (*a-c'-e'*) with "chromatic auxiliary notes around the outer fifth" ($\sharp g/bb$ around *a*, and $\sharp d'/f$ around *e'*).
- **Ezil:** Positioned within a mid-range vocal register, the *Ezil* (or *Izil*) mode assumes a secondary role, characterized by its unassuming, moderate cadence. Emotionally neutral in essence, it is seldom utilized during fasting periods, maintaining a comfortable, ordinary vocal expression. Shelemay et al. [3] stated that "*Ezil* uses the same pitch set as in *Araray*," but this pitch set is rendered as either *c-d-e-g-a* or *c-d-f-g-a*, implying that the third note lies in between *e* and *f* and causes ambiguity for Western ears.
- **Araray:** Distinguished by its high-pitched rendition, embellished with ornamental flourishes and a brisk tempo, the *Araray* mode exudes vitality and jubilation. It serves as a vehicle for conveying animated expressions, elation, and manifestations of compassion, happiness and fulfillment.

⁵ *-ne* is pronounced as in 'Nelson'

⁶ The Ethiopian and Eritrean faithful worldwide served by the chants is additional to the data reported in [14].

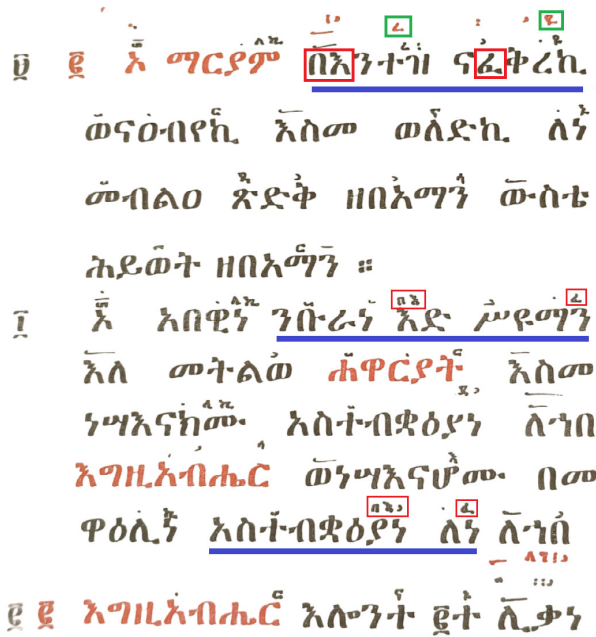


Figure 1. Interlinear letter-based notations with interspersed neumes. From the first underlined two words, the letters enclosed in red rectangles are used as short-form representations of the melody to be used over the other underlined words, sung with the same melody.

The EOTC chants rely on a sophisticated system of interlinear notations, encompassing neumatic signs interspersed between letter-based representations [1, 3]. This notation system serves as the cornerstone of melodic expression in chanting. Although some notations are common across different chanting modes, they produce distinct melodies depending on the mode, making it challenging to identify a specific mode solely based on notation. Figure 1 provides an example of the notation system used in the EOTC chants.

3. DATASET

The dataset was manually collected from the *Eat the Book* website,⁷ a hub of numerous audio books for most of the teachings in the EOTC school departments, with full and partial coverage. From the available audio books, we selected the *Se'atat Zema* (Horologium chant), which is part of *Qidase-bet* department. All the audios selected for our dataset were recorded by a single scholar at a sampling rate of 44,100 Hz and in stereo channel.

Our first step in the audio arrangement process involved narrowing the gap between the longest and shortest duration among the audios. Long audios, such as those over 13 minutes, were segmented into shorter audios of less than three minutes (180 seconds) in a way that preserves meaningful segments. This segmentation process also applied to audios that contained multiple chanting modes. For example, if an audio had 160 seconds of *Araray* mode followed

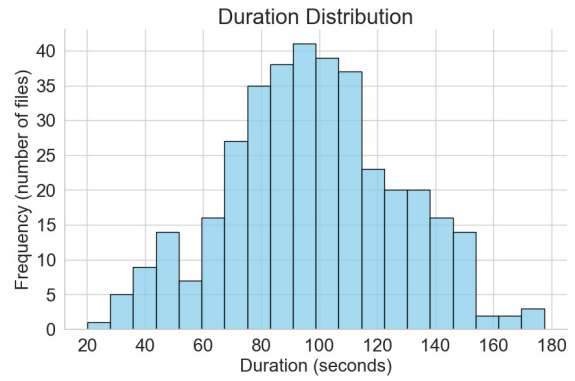


Figure 2. Distribution of audio recording length (in secs).

Modes	Shelemay and Jeffery [4]		This work	
	#	Length	#	Length
Araray	8	11m36s	118	192m36s
Ezil	6	9m56s	176	291m29s
Ge'ez	10	21m12s	75	118m6s
Total	24	42m44s	369	602m11s

Table 1. Data distribution among the chanting modes.

by 22 seconds of *Ezil* mode content, it would be segmented into two separate audios of 160 seconds and 22 seconds. Recordings that were less than three minutes but still had multiple modes were also segmented based on the respective duration of the included chanting modes.

On the other hand, short audios, like a 19-second audio, were merged with neighboring context audios when applicable to our assumptions. If no neighboring audio with the same mode was found, it would be counted as a separate audio. As we arranged all audios to be in a single mode, we have a corresponding mode label for each audio. Another audio cleaning process was removing non-chant segments as the recordings included short explanatory statements about the corresponding chants. We manually removed them to ensure that the full audio content will be for chanting. In this process we also have shortened the duration of significant silent regions, resulted in only two silent regions above two seconds, particularly 2.25 and 2.14 seconds. After such cleaning procedures, the overall duration distribution of our dataset, ranging from 20.142 seconds to 177.476 seconds, is shown in Figure 2. As our immediate future work, we are working on expanding our dataset by including annotation of word-level lyrics to audio alignment as well as other features, which are not uncovered in this paper to keep the focus. We will do more research regarding other possible additional features.

Table 1 presents the comparison between the previously used dataset (i.e., the recordings collected by Shelemay *et al.* in [4]) and our dataset. While the previous dataset, with a total of 24 instances, is less than one hour, our dataset accounts for more than 10 hours, with a total of 369 instances. The chanting mode annotations of the dataset are available on <https://github.com/mequanent/ChantingModeClassification.git>.

⁷ <https://eathebook.org/>, We acknowledge the website's administrators for their invaluable contribution.

4. YAREDAWI YEZEMA SILT CLASSIFICATION

As a preliminary study, we only consider using time-averaged audio features (i.e., the features ignoring the information lying in the temporal dimension) for *Yaredawi YeZema Silt* classification. Focusing on such features also supports our subsequent discussion on the pitch distributions of different chanting modes (see Section 5).

4.1 Feature Representations and Classifiers

Following previous works on the analysis of various kinds of music [5–11], we consider pitch distribution, the distribution of the frame-level pitch values, for the classification task. Our pipeline of feature extraction mostly resembles [10, 16], by having the stages of pitch contour extraction, stable region extraction, and pitch drift calibration. First, the pitch detection algorithm pYIN [17] is utilized for pitch contour extraction. It sets the time resolution to 128 samples (5.8 ms) while the frequency resolution to 10 cents. After having the pitch contour, the pitch distribution is obtained by having a histogram over the frame-level pitch values with a frequency resolution of also 10 cents. To analyze the time-averaged aspects of the chanting modes, extracting the stable regions of the pitch contour while discarding the sliding, ornamental or other unstable components might be helpful. We therefore reimplement two stable region extraction methods, namely the *morphic* method and the *masking* method, both proposed in [5]. There is also observable pitch drift during the performance. With an investigation of the data, we found that the pitch drifting along the whole recording is relatively small (around 1 semitone upward for the whole recording), so the pitch calibration process can be done straightforwardly with a linear regression. More specifically, the regression is performed on the pitch values 1 semitones around the global maximum of the pitch histogram. With the regression line with slope s , the pitch contour $f[t]$ indexed by time t is calibrated to $f_{\text{calibrated}}[t]$ by having $f_{\text{calibrated}}[t] := f[t] - st$.

The pitch distribution features are therefore based on the six types of pitch contours: three stabilization modes (no stabilization, stabilization with morphetic method, and stabilization with masking method) times two calibration modes (with and without calibration). Besides, several audio features are also compared: time-average mel-spectrogram, mel frequency cepstral coefficient (MFCC), and chromagram. The melspectrogram and MFCC are extracted using the `torchaudio` package [18], while the chromagram is extracted with the `librosa` package [19]. The time-average features of them are obtained simply by taking average over the time axis.

For the classifiers, we utilize the M5 (0.5M) model architecture proposed in [20]. The model is a fully convolutional network containing only 1-D convolution layers, max pooling layers and a global average pooling layer. Such a design has small number of training parameters and can capture the invariance in data [21]. While this network was taken for raw waveform, we adapt it to operate in the frequency domain regarding it as an operator invariant to

pitch shifting. To customize the model to our extracted features, we changed the receptive fields in the first convolutional layer from 80 to 3 when running on the non-raw-audio features in our experiments. For all the experiments, we adopt the categorical cross entropy loss function, Adam optimizer, learning rate of 0.001, batch size of 32, and 50 epochs, due to model convergence.

4.2 Experiment Settings

To observe how the characteristics of *YeZema Silt* vary across different recordings, we consider both the within-dataset and cross-dataset experiments. For the within-dataset experiment, we perform 5-fold cross validation (CV) on the proposed dataset and report the average classification accuracy. For the cross-dataset case, the model is trained on the proposed dataset and then tested on the recordings performed by a chanter from a different chanting department, specifically Zema-bet, in different time and location [4]. The recordings we used from [4], described in Table 1, have a sampling rate of 44100 Hz with stereo channel with 0.33 and 4.05 seconds of shortest and longest audio recordings, respectively. Lastly, to examine the reasonable identifiable audio duration among the chanting modes and how the duration affects the performance, we consider four input durations, namely 5 seconds, 10 seconds, 20 seconds and full length.

4.3 Results

Table 2 lists the classification accuracy of all the experiment settings. First, the results of full length audio show that all the pitch distribution greatly outperform other audio features by a gap of over 25 percentage points. Also, the pitch distribution is more robust than the other audio features in the cross-dataset scenario, with a performance drop by 7 to 23 percentage points. However, comparing the six pitch distributions, it is not clear which calibration or stabilization mode is better. The best accuracy over all in the CV scenario is the calibrated but non-stabilized pitch distribution, but the trend does not apply to the cross-dataset case. Besides, we observe that 1) pitch contour stabilization does help on the accuracy for most of the cases, 2) using stabilization tends to reduce the performance gap between within-dataset and cross-dataset scenarios, and 3) the masking method can reduce this gap better than the morphetic method does, though the morphetic method typically has better classification accuracy. Lastly, there is a clear trend that a longer input audio leads to a better performance. This implies that *YeZema Silt* is a long-term, song-level music concept, while it can also be signified to some extent upon a 10- to 20-sec duration, which is around the duration of a set of music notation.

Table 3 shows two example confusion matrices for both the within-dataset and cross-dataset cases. For the within-dataset case, the accuracy of each class basically follows the amount of data ($Ezil > Araray > Ge'ez$, see Table 1). The trend is different for the cross-dataset case: all classification errors occur between *Ezil* and *Araray*, a result being in line with the experience of analysis [3].

Feature representation			Within-dataset (5-fold CV)				Cross-dataset				
Pitch contour	Calibration	Stabilization	full	20 sec	10 sec	5 sec	full	20 sec	10 sec	5 sec	
	No	No		96.20	91.51	87.93	81.60	87.50	82.98	72.96	74.76
		Morphetic		95.13	87.23	83.47	73.35	83.33	84.40	76.67	70.97
		Masking		94.85	83.85	76.85	64.32	87.50	74.47	68.52	55.41
Yes	No		98.11	89.92	88.02	80.78	75.00	80.85	77.04	69.64	
	Morphetic		95.66	87.71	80.39	70.58	79.17	73.05	70.00	69.07	
	Masking		92.94	84.05	76.32	63.22	83.33	78.01	79.63	62.43	
Time-average chromagram			68.01	66.63	62.28	55.93	62.50	50.43	42.68	45.33	
Time-average mel-spectrogram			64.20	59.20	55.16	54.61	37.50	48.72	50.41	47.91	
Time-average MFCC			68.52	66.62	66.16	65.42	37.50	35.90	36.18	39.17	

Table 2. Results (classification accuracy, in %) of Yaredawi YeZema Silt classification.

5-fold CV				Cross-dataset			
	G	E	A		G	E	A
G	92.0	2.67	5.33	G	100.0	0.0	0.0
E	1.14	97.73	1.14	E	0.0	83.33	16.67
A	2.54	2.54	94.92	A	0.0	25.0	75.0

Table 3. Confusion matrices over the *Ge'ez* (G), *Ezil* (E) and *Araray* (A) classes. The reported classifier is trained on calibrated pitch contour with masking stabilization.

5. ANALYSIS OF YAREDAWI YEZEMA SILT

The goal of our analysis of *YeZema Silt* is using computational tools to individually identify the pitches utilized in the three chanting modes. Any attempt to this relies on some music theoretical assumptions. The classification results presented in Section 4.3 supports two assumptions that facilitate the analysis: first, *YeZema Silt* is a song-level property that can be satisfactorily described with time-average pitch distributions; second, *YeZema Silt* can be identified by a classifier invariant to pitch-shifting (i.e. convolution). On the other hand, the classification results also expose a few technical limitations. While the raw pitch distribution (i.e., without pitch contour stabilization) yields the best classification accuracy, it is highly noisy and therefore less applicable for our analysis purpose. In fact, we found in our study that the raw and the morphetic pitch distribution are relatively deficient in the below-mentioned analysis process. Therefore, instead of advocating a specific setting in terms of classification accuracy, we decided to use the calibrated pitch contour with masking stabilization method on the full length audio for subsequent analysis, although its performance is not the most favorable. It is worth noting that in this case, the performance gap between within-dataset CV and cross-dataset is relatively small among all settings.

Our approach, which partly resembles [10], contains three steps: 1) shift the pitch distributions of each recording such that each of them are best correlated (i.e., best aligned); 2) compute the average of the aligned pitch distribution for all the recording of the same chanting mode; 3) employ the Gaussian Mixture Model (GMM) to estimate the representative pitch set from the distribution.

Specifically, the pitch distributions of two recordings p_i

and p_j are aligned through pitch-shifting p_j by ξ_{ij} such that their cross-correlation $R_{ij} := R_{ij}[\xi]$ is maximized:

$$\xi_{ij} = -\xi_{ji} := \arg \max_{\xi} R_{ij}[\xi]. \quad (1)$$

The recording which has the highest average correlation with all the other recordings is considered as an anchor: the pitch distributions of all the other recordings are pitch-shifted to this anchor according to their optimal ξ and are then averaged for GMM fitting. The mean (μ), variance (σ^2) and weight (w) of each GMM component then represents the pitch center, pitch variance and pitch weight. The GMM fitting process is initialized by user-specified mean values to enhance convergence [10]. To facilitate the discussion, only the components having variance smaller than 100 cents are considered as representative pitches.

The top row of Fig. 3 illustrates the aligned pitch distributions for the three chanting modes and the two datasets. We observe that the recordings in the same chanting modes typically have similar pitch distributions over the two datasets. Such a consistent trend is also observed from the average pitch distributions (middle row of Fig. 3), which shows that only one pitch (the third peak from the left) from the two dataset in *Araray* is somehow different.

The bottom row of Fig. 3 shows the GMM-estimated pitch distributions for all the recordings from both datasets. By selecting the pitches summing up to maximal weights within one octave, we obtain three representative pitches for *Ge'ez* (denoted as g_1 , g_2 and g_3 , from low to high), five for *Ezil* (denoted as e_1 , e_2 , e_3 , e_4 and e_5) and also five for *Araray* (denoted as a_1 , a_2 , a_3 , a_4 and a_5).⁸ Other representative pitches outside this octave are also notated: the pitch being one octave below g_3 is denoted as G_3 , while the pitch one octave above g_1 is denoted as \hat{g}_1 . The same naming rules also apply for *Ezil* and *Araray*.

Table 4 shows the GMM-estimated parameters for the three modes. First, the pitches used in the *Ge'ez* mode are more flexible than other two modes, as can be observed by their variances than the pitches in other two modes. Among them, only g_2 has the variance less than 10 cents.

⁸ Here, the subscript number does not imply the hierarchical order of the musical scale (e.g., g_1 does not mean “the tonic of the *Ge'ez* mode”). The hierarchy of these pitches is another research question and will be considered as future work.

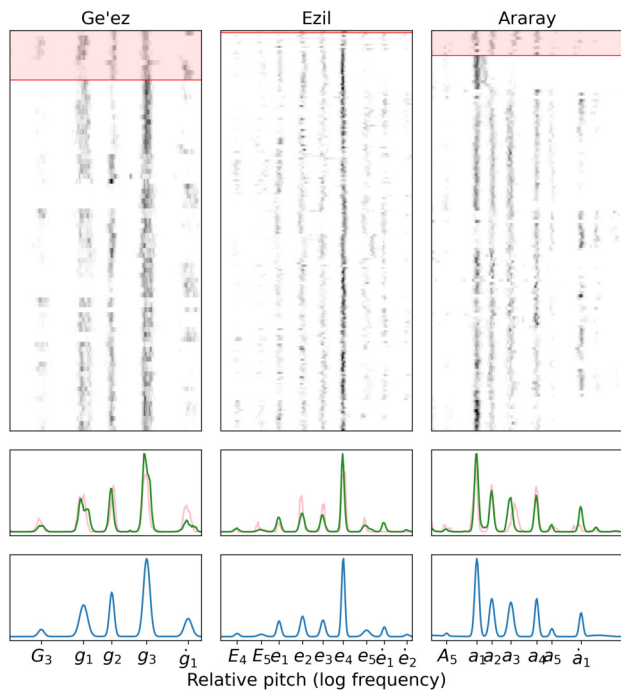


Figure 3. Illustration of pitch distributions for the three *YeZema Siltoch*. Top: the aligned pitch distributions of all the recordings. A row in the 2-D illustration represents the pitch distribution of one recording. Darker color represents larger values. Red background represents pitch distributions of the recordings in [4]. Middle: the average pitch distribution of the proposed dataset (green) and [4] (red). Bottom: GMM-estimated pitch distributions for all the recordings from both datasets. The pitch value of each note name under the bottom row is listed in Table 4.

g_3 and g_2 form a major third ($\Delta\mu = 400$ cents) while g_1 and g_2 form approximately a minor third g_2 ($\Delta\mu = 324$ cents). The pitches of g_1 and g_3 can vary by more or less semitones. Besides, we also observe that the octaves of g_1 and g_3 (i.e., G_3 and \dot{g}_1) also have large variances. This implies that such variance (flexibility of pitch) depend on the pitch name rather than the register. These findings are basically in line with the statements (a scale $\sharp g-a-bb-c'-\sharp d'-e'-f'$ while $g-c'-e'$ are the stem pitches) made in [4].

Both the *Ezil* and *Araray* modes have five representative pitches within one octave. However, the five representative pitches of them are different. For *Ezil*, all the intervals lie between 200 cents (major second) and 300 cents (minor third), while for *Araray*, the intervals distribute from 172 cents (less than a major second) to 347 cents (in between a minor third and a major third). In other words, there is a consistent trend that the intervals in *Ezil* are more equally distributed than *Araray*. There are also some flexible usage of pitch, for example, e_5 (E_5) in *Ezil*. These suggest that the pitch sets found in [4] needs revision: from our observation, each of the pitch sets used in the three EOTC chanting modes is distinctive. Besides, a mode is characterized by not only its pitch centers, but also its pitch variances.

Mode	Note name	μ	σ^2	w	$\Delta\mu$
Ge'ez	G_3	361	11	0.034	486
	g_1	847	21	0.211	324
	g_2	1171	7	0.171	400
	g_3	1571	14	0.419	476
	\dot{g}_1	2047	18	0.112	
Ezil	E_4	189	6	0.022	258
	E_5	447	14	0.023	223
	e_1	670	6	0.106	270
	e_2	940	7	0.151	234
	e_3	1174	7	0.123	232
	e_4	1406	3	0.416	268
	e_5	1674	15	0.068	204
	\dot{e}_1	1878	5	0.059	261
	\dot{e}_2	2139	5	0.013	
Araray	A_5	173	3	0.008	347
	a_1	520	6	0.318	172
	a_2	692	8	0.173	218
	a_3	910	10	0.176	297
	a_4	1207	5	0.134	174
	a_5	1381	4	0.027	335
	\dot{a}_1	1716	5	0.084	

Table 4. GMM-estimated mean (μ , in cents), variance (σ^2 , in cents), weight (w) of the representative note pitches in the Ge'ez, Ezil, and Araray modes. Reference pitch (0 cent) is 82.4 Hz. The intervals (difference between two neighboring pitches, $\Delta\mu$) are listed in the last column.

6. CONCLUSION

In this paper, we presented a research on a relatively under-explored music genre, the Ethiopian Orthodox Tewahedo Church (EOTC) chant, from three computational perspectives. First, through a rigorous data cleaning and annotation process, we presented a new and high-quality EOTC chant dataset, which can be extended for various music information retrieval (MIR) and music generation tasks. Second, we conducted a chanting mode (*YeZema Silt*) recognition task using our dataset and achieved promising results. Additionally, this paper is, to our knowledge, the first to computationally analyze the pitch sets of the EOTC chanting modes, specifically *YeZema Siltoch*, with new musicological insights. In the future, we plan to keep enriching the annotations of the datasets, by incorporating more details like lyrics, chanting options, reading tones and other potential features. Analyzing *YeZema Siltoch* using the features in the temporal dimension and the new data annotations are also our ongoing projects.

The EOTC chants encompass a wide range of styles and forms. In this paper, we specifically concentrated on the *Se'atat Zema* (Horologium chant), which falls under the *Qidase-bet* department. Our objective is to encourage responsible research on EOTC chants, as computational research in this area can lead to technological advancements that enhance the learning process and increase accessibility. Diversifying the data and MIR of EOTC chants for the protection and promotion of this spiritual-cultural heritage is also our future work in the long term.

7. ETHICS STATEMENT

The chants we are working on belongs to the Ethiopian Orthodox Tewahedo Church (EOTC). The oral traditions and beliefs that mainly preserve these chants for more than 1500 years should be credited properly. This work aims to contribute on promoting the chants and finding solutions for their easy understanding. There is no intention to modify the chants in any form.

No explicit permission request is done to the Eathbook.org administrators as recordings are open to the public education and we are using for educational purpose. We value their great effort in making the chants publicly available.

8. REFERENCES

- [1] A. Kebede, “The sacred chant of Ethiopian monotheistic churches: Music in black Jewish and Christian communities,” in *The Black Perspective in Music*, J. Southern, Ed. Brandeis University: JSTOR, 1980, pp. 20–34.
- [2] K. Shelemay, “The musician and transmission of religious tradition: The multiple roles of the Ethiopian Debtera,” *Journal of Religion in Africa*, pp. 242-260, vol. 22, 1992.
- [3] K. K. Shelemay, P. Jeffery, and I. Monson, “Oral and written transmission in Ethiopian Christian chant,” *Early Music History*, vol. 12, pp. 55–117, 1993.
- [4] K. K. Shelemay and P. Jeffery, *Ethiopian Christian Liturgical Chant: An Anthology, Part 1 to Part 3*. AR Editions, Inc., 1993, vol. 1.
- [5] S. Rosenzweig, F. Scherbaum, and M. Müller, “Detecting stable regions in frequency trajectories for tonal analysis of traditional Georgian vocal music,” in *International Society for Music Information Retrieval Conference (ISMIR)*, Delft, The Netherlands, 2019, pp. 352–359.
- [6] D. Han, R. C. Repetto, and D. Jeong, “Finding tori: Self-supervised learning for analyzing korean folk song,” in *International Society of Music Information Retrieval Conference (ISMIR)*, Milan, Italy, 2023, p. 440–447.
- [7] B. Nikzat and R. Caro Repetto, “KDC: an open corpus for computational research of dastgāhi music,” in *International Society of Music Information Retrieval Conference (ISMIR)*, Bengaluru, India, 2022, pp. 321–328.
- [8] S. Nadkarni, S. Roychowdhury, P. Rao, and M. Clayton, “Exploring the correspondence of melodic contour with gesture in raga alap singing,” in *International Society of Music Information Retrieval Conference (ISMIR)*, Milan, Italy, 2023, pp. 21–28.
- [9] R. Caro Repetto and X. Serra, “Creating a corpus of jingju (beijing opera) music and possibilities for melodic analysis,” in *International Society of Music Information Retrieval Conference (ISMIR)*, Taipei, Taiwan, 2014, pp. 313–318.
- [10] F. Scherbaum, N. Mzhavanadze, S. Rosenzweig, and M. Müller, “Tuning systems of traditional georgian singing determined from a new corpus of field recordings,” *Musicologist*, vol. 6, no. 2, pp. 142–168, 2022.
- [11] A. Vidwans, P. Verma, and P. Rao, “Classifying cultural music using melodic features,” in *2020 International Conference on Signal Processing and Communications (SPCOM)*, 2020, pp. 1–5.
- [12] M. Tsegaye, “Traditional education of the ethiopian orthodox church and its potential for tourism development (1975-present),” Master’s Thesis, Addia Ababa University, Addis Ababa, Ethiopia, 2011. [Online]. Available: <http://etd.aau.edu.et/handle/123456789/248>
- [13] B. T. Dagneu, “Ethiopian Orthodox Tewahido Church Aquaquam Zema classification model using deep learning,” Master’s Thesis, Bahir Dar University, Bahir Dar, Ethiopia, 2023.
- [14] “Ethiopian Statistical Agency: Population and Housing Census. (2007). 2007 Census Results. retrieved from https://www.statethiopia.gov.et/wp-content/uploads/2019/06/population-and-housing-census-2007-national_statistical.pdf,” Online, accessed: April 11, 2024.
- [15] G. Zemedu and Y. Assabie, “Concatenative hymn synthesis from Yared notations,” in *Advances in Natural Language Processing*, A. Przepiórkowski and M. Ogrodniczuk, Eds. Cham: Springer International Publishing, 2014, pp. 400–411.
- [16] S. Rosenzweig, F. Scherbaum, and M. Müller, “Computer-assisted analysis of field recordings: A case study of Georgian funeral songs,” *ACM Journal on Computing and Cultural Heritage*, vol. 16, no. 1, dec 2022. [Online]. Available: <https://doi.org/10.1145/3551645>
- [17] M. Mauch and S. Dixon, “Pyin: A fundamental frequency estimator using probabilistic threshold distributions,” in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 659–663.
- [18] J. Hwang, M. Hira, C. Chen, X. Zhang, Z. Ni, G. Sun, P. Ma, R. Huang, V. Pratap, Y. Zhang, A. Kumar, C.-Y. Yu, C. Zhu, C. Liu, J. Kahn, M. Ravanelli, P. Sun, S. Watanabe, Y. Shi, Y. Tao, R. Scheibler, S. Cornell, S. Kim, and S. Petridis, “Torchaudio 2.1: Advancing speech recognition, self-supervised learning, and audio processing components for pytorch,” 2023.
- [19] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, “librosa: Audio and music signal analysis in python,” in *In Proceedings of the*

14th python in science conference, vol. 8, 2015, pp. 18–25.

- [20] W. Dai, C. Dai, S. Qu, J. Li, and S. Das, “Very deep convolutional neural networks for raw waveforms,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 421–425.
- [21] J. Thickstun, Z. Harchaoui, D. P. Foster, and S. M. Kakade, “Invariances and data augmentation for supervised music transcription,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 2241–2245.

LYRICS TRANSCRIPTION FOR HUMANS: A READABILITY-AWARE BENCHMARK

Ondřej Cífka Hendrik Schreiber Luke Miner Fabian-Robert Stöter

AudioShake

ondrej@audioshake.ai, fabian@audioshake.ai

ABSTRACT

Writing down lyrics for human consumption involves not only accurately capturing word sequences, but also incorporating punctuation and formatting for clarity and to convey contextual information. This includes song structure, emotional emphasis, and contrast between lead and background vocals. While automatic lyrics transcription (ALT) systems have advanced beyond producing unstructured strings of words and are able to draw on wider context, ALT benchmarks have not kept pace and continue to focus exclusively on words. To address this gap, we introduce Jam-ALT, a comprehensive lyrics transcription benchmark. The benchmark features a complete revision of the JamendoLyrics dataset, in adherence to industry standards for lyrics transcription and formatting, along with evaluation metrics designed to capture and assess the lyric-specific nuances, laying the foundation for improving the readability of lyrics. We apply the benchmark to recent transcription systems and present additional error analysis, as well as an experimental comparison with a classical music dataset.

1. INTRODUCTION

Recent general-purpose automatic speech recognition (ASR) models trained on large datasets [1, 2] have shown a remarkable level of generalization, even improving the performance of automatic lyrics transcription (ALT) [3–5]. Remarkably, these state-of-the-art ASR models are able to take in larger temporal contexts and produce natural text with long-term coherence which, in the case of Whisper [2], includes punctuation and capitalization [6]. One may therefore ask how well these capabilities transfer from speech to lyrics. Moreover, producing a high-quality lyrics transcript suitable for user-facing music industry applications (e.g. to be displayed on streaming platforms or lyrics websites) presents some unique challenges, namely the need for specific formatting (e.g. line break placement, parentheses around background vocals) [7–9]. This calls

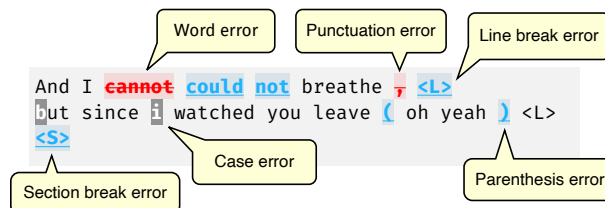


Figure 1: Error types captured by our metrics. Each token is classified as a word, punctuation mark, or parenthesis (enclosing background vocals). Special tokens are added in place of line and section breaks. Each token type is covered by a separate metric; differences in letter case are handled separately.

for a new approach to ALT evaluation and development that accounts for these distinctive nuances.

In ASR, the primary goal is a clear representation of what was said. To that end, formatting is helpful for improving the readability of transcripts [10]. Likewise, fillers like *um*, *uh*, *like*, and *you know* can be omitted to improve readability. Recent work [11] attempts to formalize this concern for clarity, proposing a novel metric geared towards assessing human readability. It employs human labelers, instructed to disregard filler words while, on the other hand, taking account of punctuation and capitalization errors that impact readability or alter the meaning of the text.

In music, on the other hand, lyrics are not simply a means of communicating meaning; they are a form of artistic expression, closely tied to the rhythm, melody, and emotionality of the song. For this reason, lyrics transcription requires a different set of considerations. Line breaks, often missing or arbitrarily placed in speech transcripts, are essential in lyrics for capturing rhyme, meter, and musical phrasing. Fillers like *oh yeah*, non-word sounds like *la-la-la* and contractions such as *I'ma* (vs. *I'm gonna*, *I am going to*) have prosodic significance, and their omission would disrupt the song's rhythm and rhyme scheme. Far from being an impediment to readability, they are key to any faithful rendition of a song for artist and fan alike.

We believe that readability-aware models for lyrics transcription have the potential to facilitate novel applications extending beyond the realms of metadata extraction and relatively crude karaoke subtitles. However, in order to advance in this research direction, the ability to accurately evaluate ALT systems in the aforementioned aspects is vi-



© O. Cífka, H. Schreiber, L. Miner, and F.-R. Stöter. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** O. Cífka, H. Schreiber, L. Miner, and F.-R. Stöter, "Lyrics Transcription for Humans: A Readability-Aware Benchmark", in *Proc. of the 25th Int. Society for Music Information Retrieval Conf.*, San Francisco, United States, 2024.

tal. To the best of our knowledge, existing ALT literature not only overlooks readability, but evaluates on datasets (e.g. [12–15]) that have not been designed specifically for ALT and lack some or all of the desirable features discussed above.

One of the datasets widely adopted by recent works [3, 4, 16–18] as an ALT test set is JamendoLyrics [14], originally a lyrics alignment benchmark. Its most recent (“MultiLang”) version [19] contains four languages and a diverse set of genres, making it attractive as a testbed for lyrics-related tasks. However, we found that, in addition to lacking in the aspects discussed above, the lyrics are sometimes inaccurate or incomplete. While such lyrics may be perfectly acceptable as input for lyrics alignment (and indeed representative of a real-world scenario for that task), they are less suitable as a target for ALT.

To address these issues and help to guide future ALT research, we present the **Jam-ALT** benchmark, consisting of: (1) a revised version of JamendoLyrics MultiLang following a newly created annotation guide that unifies the music industry’s conventions for lyrics transcription and formatting (in particular, regarding punctuation, line breaks, letter case, and non-word vocal sounds); (2) a comprehensive set of automated evaluation metrics designed to capture and distinguish different types of errors relevant to (1). The dataset and the implementation of the metrics are available via the project website.¹ Additionally, to explore the applicability of the proposed metrics to other datasets, we present results on the *Schubert Winterreise Dataset* (SWD) [20].

2. DATASET

Our first contribution is a revision of the JamendoLyrics MultiLang dataset [19] to make it more suitable as a lyrics transcription test set. Different sets of guidelines for lyrics transcription and formatting exist within the music industry; we consider guidelines by Apple [7], LyricFind [8], and Musixmatch [9], from which we extracted the following general rules:

1. Only transcribe words and vocal sounds audible in the recording; exclude credits, section labels, style markings, non-vocal sounds, etc.
2. Break lyrics up into lines and sections; separate sections by a single blank line.
3. Include each word, line and section as many times as heard. Do not use shorthands to indicate repetitions.
4. Start each line with a capital letter; respect standard capitalization rules for each language.
5. Respect standard punctuation rules, but never end a line with a comma or a period.
6. Use standard spelling, including standardized spelling for slang where appropriate.
7. Mark elisions (incomplete words) and contractions with an apostrophe.
8. Transcribe background vocals and non-word vocal sounds if they contribute to the content of the song.

9. Place background vocals in parentheses.

The original JamendoLyrics dataset adheres to rules 1, 3, and 8, partially 2 and 6 (up to some missing diacritics, misspellings, and misplaced line breaks), but lacks punctuation and is lowercase, thus ignoring rules 4, 5, 7, and 9. Moreover, as mentioned above, we found that the lyrics do not always accurately correspond to the audio.

To address these issues, we revised the lyrics in order for them to obey all of the above rules and to match the recordings as closely as possible. As the above rules are fairly unspecific, we created a detailed annotation guide where we have attempted to resolve minor discrepancies among the source guidelines [7–9] and fill in missing details (including language-specific nuances). This annotation guide is released together with the dataset.

Each lyric file was revised by a single annotator proficient in the language, then reviewed by two other annotators. In coordination with the authors of [19], one of the 20 French songs was removed following the detection of potentially harmful content.

Examples of lyrics before and after revision can be found on the project website.

3. METRICS

In this section, we first discuss our adaptation of the conventional *word error rate* (WER) metric and then our proposed precision and recall measures for punctuation and formatting. Our goal here is to design a comprehensive set of metrics that covers all possible transcription errors while allowing us to distinguish between different types of errors (see Fig. 1 for a visual overview of the error types). Note, however, that our goal is *not* to create metrics that completely align with the rules put forth in Section 2 or correlate with a specific notion of readability; the metrics should be general enough to apply to any plain-text lyrics dataset and adapt to its formatting style.

3.1 Word Error Rates

The standard speech recognition metric, WER, is defined as the edit distance (a.k.a. Levenshtein distance) between the *hypothesis* (predicted transcription) and the *reference* (ground-truth transcript), normalized by the length of the reference. If D , I , and S are the number of word *deletions*, *insertions*, and *substitutions* respectively, for the minimal sequence of edits needed to turn the reference into the hypothesis, and H is the number of unchanged words (*hits*), then:

$$\text{WER} = \frac{S + D + I}{S + D + H} = \frac{S + D + I}{N}, \quad (1)$$

where N is the total number of reference words.

Typically, the hypothesis and the reference are pre-processed to make the metric insensitive to variations in punctuation, letter case, and whitespace, but no single standard pre-processing procedure exists. In this work, we apply Moses-style [21] punctuation normalization and tokenization, then remove all non-word tokens. Before computing the WER, we lowercase each token to make the met-

¹ <https://audioshake.github.io/jam-alt/>

ric case-insensitive, but also keep track of the token’s original form. To then measure the error in letter case, for every *hit* in the minimal edit sequence, we compare the original forms of the hypothesis and the reference token and count an error if they differ. We then compute a *case-sensitive word error rate* WER' as:

$$WER' = \frac{S + D + I + E_{\text{case}}}{S + D + H} = WER + \frac{E_{\text{case}}}{N}, \quad (2)$$

where E_{case} is the number of casing errors. We include both variants (1) and (2) in our benchmark.

3.2 Punctuation and Line Breaks

Since the output of ASR systems traditionally lacks punctuation, a common ASR post-processing step – *punctuation restoration* [22] – consists of recovering it. This task is usually evaluated using precision and recall:

$$\begin{aligned} P &= \frac{\# \text{ correctly predicted symbols}}{\# \text{ predicted symbols}}, \\ R &= \frac{\# \text{ correctly predicted symbols}}{\# \text{ expected symbols}}. \end{aligned} \quad (3)$$

In this original setting where the system only inserts punctuation and the words remain intact, computing the metrics is trivial. In contrast, in our end-to-end setting, the hypothesis and the reference may use different words, and hence computing the numerator in Eq. (3) requires an alignment between the two. We leverage the same alignment as used in Section 3.1, but computed on text that includes punctuation. Moreover, we extend this approach to account for line breaks, which, though traditionally ignored in speech data, are particularly important for lyrics.

We use the pre-processing from Section 3.1, but preserve punctuation tokens and, as in [23, 24], add special tokens in place of line and section breaks; this leaves us with five token types: word \bar{W} , punctuation \bar{P} , parenthesis \bar{B} (separate due to its distinctive function), line break \bar{L} , and section break \bar{S} .² After computing the alignment between the hypothesis tokens and the reference tokens, we iterate through it in order to count, for each token type $T \in \{\bar{W}, \bar{P}, \bar{B}, \bar{L}, \bar{S}\}$, its number of deletions D_T , insertions I_T , substitutions S_T , and hits H_T . In general, each edit operation is simply attributed to the type of the token affected (e.g. the insertion of a punctuation mark counts towards I_P). However, a substitution of a token of type T by a token of type $T' \neq T$ is counted as two operations: a deletion of type T (counting towards D_T) and an insertion of type T' (counting towards $I_{T'}$).

We can now use these counts to define a precision, recall, and F-1 metric for each token type:

$$\begin{aligned} P_T &= \frac{H_T}{H_T + S_T + I_T}, & R_T &= \frac{H_T}{H_T + S_T + D_T}, \\ F_T &= \frac{2}{P_T^{-1} + R_T^{-1}}. \end{aligned} \quad (4)$$

² We define a section break as one or more blank lines. Hence, every section break is explicitly preceded by a line break in our representation.

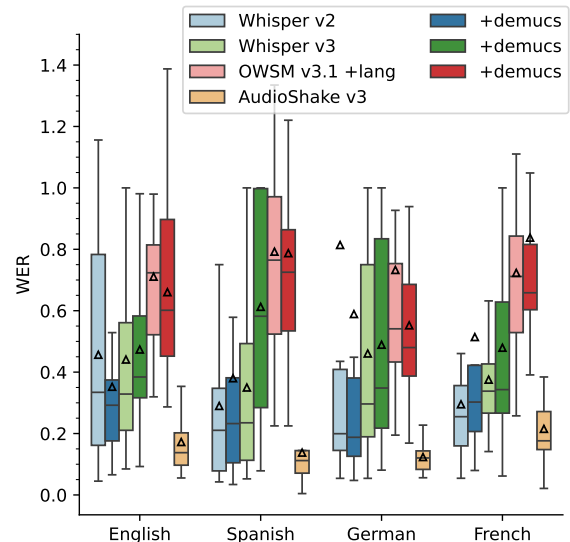


Figure 2: Song-level word error rates by language. Note that strong outliers occur; for clarity, they are not displayed here, but affect the means, which are indicated by triangles.

4. RESULTS

4.1 Benchmark Results

Table 1 shows the performance of various transcription systems on our benchmark. Fig. 2 shows the distributions of song-level word error rates by language.

We include two recent, freely available models capable of transcribing long, unsegmented audio: Whisper [2] (*large-v2* and *large-v3*) and OWSM 3.1 [25] (*owsm_v3.1_ebf*). For both models, we use Whisper-style long-form transcription with a beam size of 5. Both models have language identification capabilities, but may perform better if the correct language is specified; for Whisper, we evaluate both options, while for OWSM, for simplicity, we only evaluate with the language provided. For Whisper, which exhibits great variation between runs due to its stochastic decoding strategy, we report averages over 5 runs. We optionally use HTDemucs [26] to isolate the vocals from the input audio.

Whisper and OWSM are general-purpose speech recognition models and are not designed for lyrics transcription. To make a fairer comparison, we apply simple post-processing to their outputs to improve the formatting: (1) The models do not produce line breaks, but split their output into timestamped segments; we insert line breaks between these segments. (2) We remove unwanted end-of-line punctuation (all non-word characters except for `! ? ' " »`) and uppercase the first letter of every line.³

We also evaluate LyricWhiz [4], a lyrics transcription system combining Whisper with the commercially available instruction-following language model ChatGPT [27]. We report averages over two outputs per song (English only), kindly provided by the LyricWhiz authors. Finally,

³ Although we observed that this transformation tends to improve the outputs for Whisper and OWSM, in general, it may make evaluation results worse if the line break predictions are incorrect. For this reason, we do not include this step as a fixed part of our benchmark.

	All languages						English						Spanish		German		French	
	WER	WER'	F_P	F_B	F_L	F_S	WER	WER'	F_P	F_B	F_L	F_S	WER	WER'	WER	WER'	WER	WER'
Whisper v2	37.8	42.1	44.2	—	69.3	3.3	43.8	47.5	31.5	—	63.0	11.2	25.8	31.5	54.5	59.3	27.7	31.1
+lang	27.9	32.6	45.0	—	70.4	3.7	39.7	43.7	34.9	—	65.5	11.6	21.9	27.7	19.9	26.0	27.1	30.5
+demucs	44.5	49.8	41.6	—	61.2	—	33.3	39.1	42.2	—	53.9	—	39.6	46.5	65.2	70.4	43.3	46.9
+lang	33.5	39.3	39.4	—	60.6	—	35.6	41.3	41.8	—	53.4	—	34.9	42.2	23.9	30.4	38.2	42.1
Whisper v3	35.5	39.7	43.0	—	73.5	1.0	37.7	42.5	41.4	—	71.5	2.6	28.6	33.6	40.7	44.6	34.7	38.0
+lang	32.6	37.2	43.7	—	73.9	0.6	36.4	41.4	41.8	—	72.5	2.6	22.4	28.0	35.9	40.4	34.7	38.0
+demucs	48.0	51.6	33.0	—	65.7	—	43.0	47.2	25.8	—	66.9	—	61.5	64.9	43.5	47.4	44.9	48.2
+lang	46.6	50.4	33.7	—	65.8	—	43.0	47.2	25.8	—	66.9	—	58.6	62.1	40.8	44.9	44.9	48.3
OWSM v3.1+lang	69.3	75.0	22.5	0.6	37.8	—	68.6	74.0	22.3	—	42.7	—	73.3	78.5	63.3	71.8	71.6	75.7
+demucs	66.5	72.6	20.0	0.0	41.1	—	63.4	69.4	21.5	0.0	47.3	—	70.8	76.0	51.8	62.0	78.5	82.1
LyricWhiz	—	—	—	—	—	—	24.6	28.0	34.0	—	74.0	1.4	—	—	—	—	—	—
AudioShake v3	16.1	20.1	57.0	29.4	84.4	73.9	17.3	20.9	65.3	37.9	84.3	84.8	12.6	17.7	12.6	17.5	20.8	23.5
JamendoLyrics	11.1	29.6	—	—	93.3	85.3	14.4	29.6	—	—	88.1	77.9	14.0	29.1	5.0	37.6	10.3	23.3

Table 1: Benchmark results (all metrics shown as percentages). WER is word error rate, WER' is case-sensitive WER, the rest are F-measures. +demucs indicates vocal separation using HTDemucs; +lang indicates that the language of each song was provided to the model instead of relying on auto-detection. Whisper results are averages over 5 runs with different random seeds, LyricWhiz over 2 runs; OWSM and AudioShake are deterministic, hence the results are from a single run. The best results achieved by open-source systems are shown in **bold**. LyricWhiz and AudioShake are listed separately, because they rely on proprietary technology. The last row shows metrics computed between the original JamendoLyrics dataset as the hypotheses and our revision as the reference. For full results by language, see the project website.

	All			EN	ES	DE	FR
	WER	F_L	F_S	WER			
Whisper v2	39.1	70.0	2.8	43.0	31.7	54.7	28.0
+lang	28.8	71.0	2.6	38.8	27.9	19.8	27.4
+demucs	46.2	61.5	—	33.6	43.9	65.5	44.1
+lang	34.8	61.2	—	36.1	39.3	23.9	38.9
Whisper v3	37.7	71.6	1.0	39.3	34.5	40.8	36.1
+lang	34.9	72.3	0.6	38.0	28.9	36.0	36.1
+demucs	49.6	65.3	—	44.3	65.8	43.5	45.7
+lang	48.3	65.4	—	44.3	63.1	40.8	45.7
OWSM v3.1+lang	70.3	39.0	—	69.9	75.7	63.5	71.9
+demucs	67.5	41.6	—	65.0	72.7	51.7	79.1
LyricWhiz	—	—	—	23.7	—	—	—
AudioShake v3	19.4	82.3	64.5	22.5	18.7	13.8	21.7
Jam-ALT	11.5	94.0	85.1	15.7	14.4	5.0	10.4

Table 2: Results with the original JamendoLyrics (i.e. before revision) as reference. The last row corresponds to our revision. See also the caption of Table 1.

as an example of an ALT system built with formatting and readability in mind, we include our in-house lyrics transcription system, which integrates vocal separation.

As a first general observation, consistent with previous studies [4, 5], the performance of Whisper models is relatively good, considering that they were not specifically designed for lyrics transcription. Among the formatting metrics, we highlight a high accuracy in line break prediction. This shows that, although the segments output by Whisper do not always impose a meaningful structure, in music, they do in many cases coincide with lyric lines.

Somewhat counter-intuitively, for Whisper, inputting isolated vocals (+demucs) tends to substantially degrade the results (with the single exception of large-v2 for English). Whisper’s language identification mechanism also turns out to have a significant effect, in that disabling

it and instead inputting the known language of the song (+lang) tends to result in a sizeable drop in WER, especially on languages different from English. This suggests that the language detected by Whisper is often incorrect.

We also observe that Whisper v3 does not necessarily perform better on lyrics than v2. In fact, the WER increases from 27.9 to 32.6 when comparing Whisper v2 +lang to v3 +lang.

The improvement of LyricWhiz over plain Whisper in terms of WER is clear and even sharper than reported in [4]. We also see some improvement in terms of line breaks and punctuation.

Regarding OWSM, its performance is far behind Whisper, with differences far larger than reported in [25] for speech, strongly suggesting that OWSM is poorly suited for ALT, at least without finetuning. With isolated vocals as input, the error is slightly reduced, but still large.

As for our own system, it outperforms all of the above on all metrics shown in Table 1, by a large margin, e.g. with a 57% reduction in overall WER compared to Whisper v2. It is also the only one achieving acceptable accuracy for parentheses (B) and section breaks (S).

4.2 Effect of Revisions

The revisions described in Section 2 have enabled us to compute metrics related to letter case and punctuation, features that are missing from the original dataset. However, the revisions also involved correcting words and line breaks; to measure the effect of these corrections, we present in Table 2 the relevant metrics computed on the original JamendoLyrics data. Comparing Tables 1 and 2, we note that the revisions have mostly improved the results, notably reducing the overall WER (by 1.7, or 5.3%, on average) for all systems, with Spanish seeing the sharpest drop (4.7, or 17.4%, on average, likely due to fre-

quently missing accents in the original data). The general trends – in particular, the ranking based on WER and F_L – remain mostly unchanged.

To quantify the extent of our revisions more directly, we also evaluate both versions of the lyrics against each other and include the results as the last row in Tables 1 and 2. Remarkably, in terms of word tokens, Jam-ALT differs from JamendoLyrics by about 11 % (around 15 % for English and Spanish), which is substantially more than the difference between system performance on the two dataset versions. One potential explanation is that a significant number of the corrections correspond to low-intelligibility singing, which is prone to transcription errors, or to background vocals, which are susceptible to being omitted by transcription systems.

4.3 Error Analysis

In this section, we further analyze the errors made by selected systems on our benchmark.

First, we visualize in Fig. 3 how each type of edit operation contributes to the WER. Besides the basic edit operations (hits, substitutions, insertions, deletions), we include *case errors* from Section 3.1; that is, a hit with a difference in letter case is shown as a case error instead. Moreover, to account for small spelling differences, we consider a substitution as a *near hit* when the replacement differs from the reference in at most two letters.⁴

With Whisper, we observe that inputting separated vocals causes more insertions (and longer output) in v2, but more deletions (and shorter output) in v3. Upon inspecting the outputs, we find that Whisper has a general tendency to omit parts of the lyrics (often the entire song) and instead produce generic or irrelevant text, and that this is more frequent with separated vocals, especially with v3. On the other hand, OWSM shows a slight improvement with separated vocals, but its predictions contain significantly more substitutions, suggesting that they are more often incorrect on a word-by-word basis.

Next, we focus on errors in punctuation and formatting and investigate how often different token types are substituted for each other. To this end, we count the edit operations as in Section 3.2, but preserve the information about substitutions across the four non-word token types (P, B, L, S). We then present this information in a form akin to a *confusion matrix*, adding a special “null” token type \emptyset to account for insertions and deletions.

The result is shown in Fig. 4 for three selected systems. Most errors are insertions and deletions, but another frequent type of error is the replacement of a line break by a punctuation mark, especially in Whisper models. This is explained by the fact that our guidelines forbid most end-of-line punctuation, and hence, when transcription omits a line break, inserting a punctuation mark in its place is often needed to maintain grammatical correctness.

⁴ More precisely, we count a *near hit* if, after removing apostrophes from the two words, their character-level Levenshtein distance is at most 2, and strictly less than half the length of the longer of the two words. Examples include *an/and*, *gon'/gonna*, *therel/their/they/them*, but not *alan* or *this/that*.

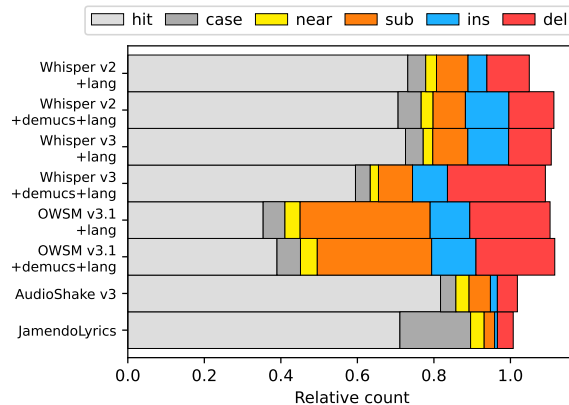


Figure 3: Word edit operation frequencies on our benchmark (one run per system). *Near* are substitutions that differ in few characters, *sub* are the remaining substitutions. *case* are hits with case errors, *hit* are the remaining (case-sensitive) hits. The rest are *insertions* and *deletions*. The frequencies are normalized by the reference length, so that:

- $hit + case + near + sub + del = 1$,
- $WER = near + sub + ins + del$,
- $WER' - WER = case$,
- $hit + case + near + sub + ins$ corresponds to the length of the prediction.

By manual inspection of the transcriptions, we find that Whisper tends to produce much longer lines than in the reference and frequently outputs periods (forbidden by our annotation guide as a sentence separator) and, occasionally, spuriously repeated punctuation.

4.4 Schubert Winterreise Dataset

To explore the application of the proposed metrics to other datasets, we additionally perform an evaluation on the *Schubert Winterreise Dataset* (SWD) [20]. SWD comprises nine audio versions of Franz Schubert’s 24-song cycle *Winterreise*, along with symbolic representations, lyrics, and other annotations. An example of Romantic music based on early 19th century German poetry, it contrasts with JamendoLyrics and presents an interesting challenge for ALT. For our evaluation, we pick a single version, SC06 (a 2006 live recording of singer Randall Scarlata), one of the two with audio publicly available.

The lyrics in SWD are formatted as poems – containing line and section breaks –, but their spelling and punctuation, mirroring an 1827 edition of the score [28], does not exactly match our annotation guide. To make them adhere to our punctuation and capitalization rules, we apply a simple transformation to the lyrics: replace all unwanted punctuation (. ; : -) with commas, then remove all end-of-line commas and uppercase the first letter of each line. Note, however, that even after this transformation, the lyrics’ obsolete spelling – predating the 1996 German orthography reform – violates our annotation guide to some extent (mainly in the usage of the letter β and the treatment of elisions), which is expected to distort the WER.

We evaluate all models with the language provided (i.e.

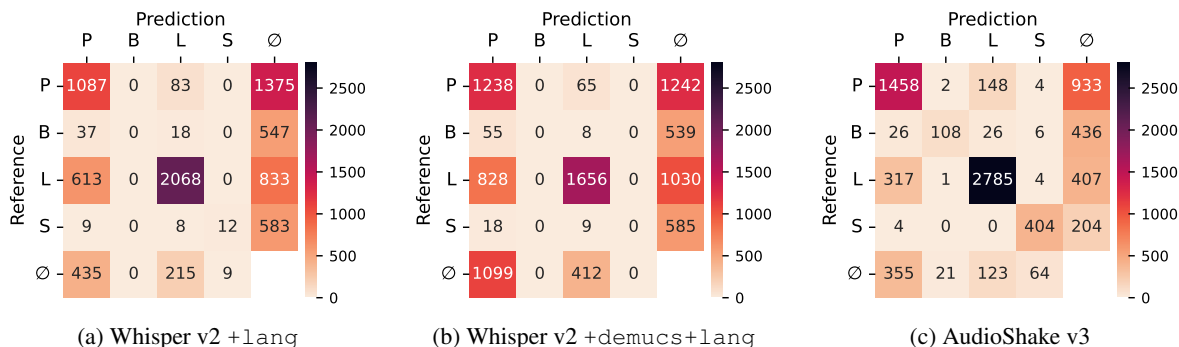


Figure 4: Edit operation counts on non-word (punctuation and formatting) tokens by token type (P = punctuation, B = parenthesis, L = line break, S = section break). ∅ denotes the absence of a token, i.e. it stands for insertion (on the *reference* axis) or deletion (on the *prediction* axis). Substitution of *of/by* a *word* token is counted as an insertion/deletion, respectively. Only a single run per system is considered.

	WER	WER'	F _P	F _L	F _S
Whisper v2	34.5	40.4	42.6	66.2	—
+demucs	41.4	47.2	38.0	61.4	—
Whisper v3	59.0	63.8	40.0	63.6	—
+demucs	52.3	58.6	34.7	63.3	0.0
OWSM v3.1	75.6	82.5	12.9	39.6	4.9
+demucs	82.9	91.8	17.0	39.2	—
AudioShake v3	24.3	29.1	50.9	80.0	72.0

Table 3: Results on performance SC06 from SWD. Only punctuation (P), line breaks (L) and section breaks (S) are included, as the ground truth lyrics do not contain any parentheses. Whisper results are averages over 5 runs with different random seeds. The best result in each column, excluding AudioShake, is shown in **bold**. For full results, see the project website.

disabling language identification). The results are shown in Table 3 and further error analysis in Fig. 5. We notice substantially worse performance on SWD than the German section of our benchmark (Table 1): for example, WER for Whisper v2 + lang increased from 19.9 to 34.5. This likely reflects the more challenging nature of the dataset, but also possibly the mismatched spelling, as suggested by a higher frequency of near hits (see Fig. 5) than seen in Section 4.3 (Fig. 3).

5. DISCUSSION

Given our focus on formatting and punctuation, the question arises to what extent they are in fact dependent on the audio. In particular, could line and section boundaries be accurately predicted just from the textual context, e.g. based on metrical patterns, rhyme, syntax, and semantics? To answer this, we suggest an experiment where a human annotator is tasked with formatting given lyrics first without and then with access to the audio. Such a task would, however, be highly time-consuming and require expert annotators unfamiliar with the songs. As a proxy, one might instead train a *formatting restoration* model on lyrics or use a general-purpose instruction-following language model. Our attempts in this regard have only had limited success

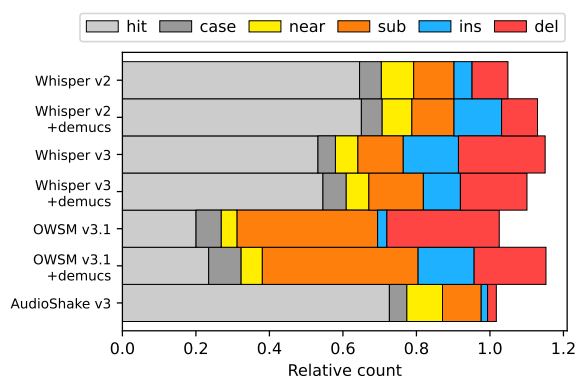


Figure 5: Word edit operation frequencies on SWD. See the caption of Fig. 3.

and we therefore leave such experiments for future work.

Another issue is that there may not always be a single correct division into lines and sections. For example, in a song with relatively short lines, it may be acceptable to join pairs of adjacent lines, especially in the absence of rhyme. Likewise, 4-line sections may be joined to create 8-line sections and so forth. However, it is not obvious how to relax the metrics to allow for this kind of variation. Doing so rigorously would likely require additional annotations, which is contrary to our goal of creating a set of generally applicable metrics. A possible solution compatible with this idea is to create multiple references and pick the best-scoring one during evaluation.

6. CONCLUSION

We have proposed Jam-ALT, a new benchmark for ALT, based on the music industry’s lyrics guidelines. Our results show how existing systems differ in their performance on different aspects of the task, and we hope that the benchmark will be beneficial in guiding future ALT research.

7. ACKNOWLEDGMENT

We would like to thank Laura Ibáñez, Pamela Ode, Mathieu Fontaine, Claudia Faller, Constantinos Dimitriou,

and Kateřina Apolínová for their help with data annotation. We are also thankful to Meinard Müller and Hans-Ulrich Berendes for their helpful comments on the manuscript.

8. REFERENCES

- [1] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, “wav2vec 2.0: A framework for self-supervised learning of speech representations,” in *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems*, 2020. [Online]. Available: <https://proceedings.neurips.cc/paper/2020/hash/92d1e1eb1cd6f9fba3227870bb6d7f07-Abstract.html>
- [2] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. Mcleavy, and I. Sutskever, “Robust speech recognition via large-scale weak supervision,” in *Proceedings of the 40th International Conference on Machine Learning*, vol. 202. PMLR, 23–29 Jul 2023, pp. 28 492–28 518. [Online]. Available: <https://proceedings.mlr.press/v202/radford23a.html>
- [3] L. Ou, X. Gu, and Y. Wang, “Transfer learning of wav2vec 2.0 for automatic lyric transcription,” in *Proceedings of the 23rd International Society for Music Information Retrieval Conference (ISMIR 2022)*, Bengaluru, India, 2022, pp. 891–899.
- [4] L. Zhuo, R. Yuan, J. Pan, Y. Ma, Y. Li, G. Zhang, S. Liu, R. B. Dannenberg, J. Fu, C. Lin, E. Benetos, W. Chen, W. Xue, and Y. Guo, “LyricWhiz: Robust multilingual zero-shot lyrics transcription by whispering to ChatGPT,” in *Proceedings of the 24th International Society for Music Information Retrieval Conference (ISMIR 2023)*, Milan, Italy, 2023.
- [5] J. Wang, C. Leong, Y. Lin, L. Su, and J. R. Jang, “Adapting pretrained speech model for Mandarin lyrics transcription and alignment,” in *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU 2023)*. IEEE, 2023, pp. 1–8. [Online]. Available: <https://doi.org/10.1109/ASRU57964.2023.10389800>
- [6] L. R. S. Gris, R. Marcacini, A. C. Júnior, E. Casanova, A. da Silva Soares, and S. M. Aluisio, “Evaluating OpenAI’s Whisper ASR for punctuation prediction and topic modeling of life histories of the Museum of the Person,” *CoRR*, vol. abs/2305.14580, 2023. [Online]. Available: <https://doi.org/10.48550/arXiv.2305.14580>
- [7] Apple, “Review guidelines for submitting lyrics,” 2023, accessed: 2023-09-18. [Online]. Available: <https://web.archive.org/web/20230718032545/https://artists.apple.com/support/1111-lyrics-guidelines>
- [8] LyricFind, “Lyric formatting guidelines,” 2023, accessed: 2023-09-18. [Online]. Available: https://web.archive.org/web/20230521044423/https://docs.lyricfind.com/LyricFind_LyricFormattingGuidelines.pdf
- [9] Musixmatch, “Guidelines,” 2023, accessed: 2023-09-23. [Online]. Available: <https://web.archive.org/web/20230920234602/https://community.musixmatch.com/guidelines>
- [10] D. A. Jones, F. Wolf, E. Gibson, E. Williams, E. Fedorenko, D. A. Reynolds, and M. Zissman, “Measuring the readability of automatic speech-to-text transcripts,” in *Proc. 8th European Conference on Speech Communication and Technology (Eurospeech 2003)*, 2003, pp. 1585–1588.
- [11] Apple, “Humanizing word error rate for ASR transcript readability and accessibility,” 2024, accessed: 2024-04-09. [Online]. Available: <https://machinelearning.apple.com/research/humanizing-wer>
- [12] C.-L. Hsu and J.-S. R. Jang, “On the improvement of singing voice separation for monaural recordings using the MIR-1K dataset,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 2, pp. 310–319, 2010.
- [13] G. Meseguer-Brocal, A. Cohen-Hadria, and G. Peeters, “DALI: A large dataset of synchronized audio, lyrics and notes, automatically created using teacher-student machine learning paradigm,” in *Proceedings of the 19th International Society for Music Information Retrieval Conference (ISMIR 2018)*. ISMIR, Nov. 2018, pp. 431–437. [Online]. Available: <https://doi.org/10.5281/zenodo.1492443>
- [14] D. Stoller, S. Durand, and S. Ewert, “End-to-end lyrics alignment for polyphonic music using an audio-to-character recognition model,” in *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, UK, 2019, pp. 181–185.
- [15] Y. Wang, X. Wang, P. Zhu, J. Wu, H. Li, H. Xue, Y. Zhang, L. Xie, and M. Bi, “Opencpop: A high-quality open source Chinese popular song corpus for singing voice synthesis,” in *Interspeech 2022, 23rd Annual Conference of the International Speech Communication Association, Incheon, Korea, 18-22 September 2022*, H. Ko and J. H. L. Hansen, Eds. ISCA, 2022, pp. 4242–4246. [Online]. Available: <https://doi.org/10.21437/Interspeech.2022-48>
- [16] C. Gupta, E. Yilmaz, and H. Li, “Automatic lyrics alignment and transcription in polyphonic music: Does background music help?” in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 496–500. [Online]. Available: <https://doi.org/10.1109/ICASSP40776.2020.9054567>
- [17] E. Demirel, S. Ahlbäck, and S. Dixon, “MSTRE-Net: Multistreaming acoustic modeling for automatic lyrics transcription,” in *Proceedings of the 22nd International Society for Music Information Retrieval Conference (ISMIR 2021)*, J. H. Lee, A. Lerch,

- Z. Duan, J. Nam, P. Rao, P. van Kranenburg, and A. Srinivasamurthy, Eds., 2021, pp. 151–158. [Online]. Available: <https://archives.ismir.net/ismir2021/paper/000018.pdf>
- [18] E. Demirel, S. Ahlbäck, and S. Dixon, “Low resource audio-to-lyrics alignment from polyphonic music recordings,” in *2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 586–590. [Online]. Available: <https://doi.org/10.1109/ICASSP39728.2021.9414395>
- [19] S. Durand, D. Stoller, and S. Ewert, “Contrastive learning-based audio to lyrics alignment for multiple languages,” in *2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Rhodes Island, Greece, 2023, pp. 1–5.
- [20] C. Weiß, F. Zalkow, V. Arifi-Müller, M. Müller, H. V. Koops, A. Volk, and H. G. Grohganz, “Schubert winterreise dataset: A multimodal scenario for music analysis,” *ACM Journal on Computing and Cultural Heritage*, vol. 14, no. 2, pp. 25:1–25:18, 2021. [Online]. Available: <https://doi.org/10.1145/3429743>
- [21] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, “Moses: Open source toolkit for statistical machine translation,” in *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*. Prague, Czech Republic: Association for Computational Linguistics, Jun. 2007, pp. 177–180. [Online]. Available: <https://aclanthology.org/P07-2045>
- [22] V. F. Pais and D. Tufis, “Capitalization and punctuation restoration: a survey,” *Artificial Intelligence Review*, vol. 55, no. 3, pp. 1681–1722, 2022. [Online]. Available: <https://doi.org/10.1007/s10462-021-10051-x>
- [23] E. Matusov, P. Wilken, and Y. Georgakopoulou, “Customizing neural machine translation for subtitling,” in *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*. Florence, Italy: Association for Computational Linguistics, Aug. 2019, pp. 82–93. [Online]. Available: <https://aclanthology.org/W19-5209>
- [24] A. Karakanta, M. Negri, and M. Turchi, “Is 42 the answer to everything in subtitling-oriented speech translation?” in *Proceedings of the 17th International Conference on Spoken Language Translation*. Online: Association for Computational Linguistics, Jul. 2020, pp. 209–219. [Online]. Available: <https://aclanthology.org/2020.iwslt-1.26>
- [25] Y. Peng, J. Tian, W. Chen, S. Arora, B. Yan, Y. Sudo, M. Shakeel, K. Choi, J. Shi, X. Chang, J. Jung, and S. Watanabe, “OWSM v3.1: Better and faster open Whisper-style speech models based on E-Branchformer,” *CoRR*, vol. abs/2401.16658, 2024. [Online]. Available: <https://doi.org/10.48550/arXiv.2401.16658>
- [26] S. Rouard, F. Massa, and A. Défossez, “Hybrid Transformers for music source separation,” in *2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Rhodes Island, Greece, 2023, pp. 1–5.
- [27] OpenAI, “Introducing ChatGPT,” OpenAI Blog. [Online]. Available: <https://openai.com/blog/chatgpt>
- [28] F. Schubert, “Winterreise. Ein Cyclus von Liedern von Wilhelm Müller,” *Gesänge für eine Singstimme mit Klavierbegleitung*, Edition Peters, No.20a, n.d. Plate 9023, 1827. [Online]. Available: http://ks4.imslp.info/files/imglinks/usimg/9/92/IMSLP00414-Schubert_-_Winterreise.pdf

A CRITICAL SURVEY OF RESEARCH IN MUSIC GENRE RECOGNITION

Owen Green¹ Bob L. T. Sturm² Georgina Born³ Melanie Wald-Fuhrmann¹

¹ Department of Music, Max Planck Institute for Empirical Aesthetics, Frankfurt, Germany

² Division of Speech, Music and Hearing, KTH, Stockholm, Sweden

³ Department of Anthropology, Institute for Advanced Studies, UCL, London, UK

owen.green@ae.mpg.de, bobs@kth.se

ABSTRACT

This paper surveys 560 publications about music genre recognition (MGR) published between 2013–2022, complementing the comprehensive survey of [474], which covered the time frame 1995–2012 (467 publications). For each publication we determine its main functions: a review of research, a contribution to evaluation methodology, or an experimental work. For each experimental work we note the data, experimental approach, and figure of merit it applies. We also note the extents to which any publication engages with work critical of MGR as a research problem, as well as genre theory. Our bibliographic analysis shows for MGR research: 1) it typically does not meaningfully engage with any critique of itself; and 2) it typically does not meaningfully engage with work in genre theory.

1. INTRODUCTION

Despite much more work [1–560] music genre recognition (MGR) still remains a compelling problem to solve by a machine. This work comes on top of the 467 publications surveyed over a decade ago by Sturm [474]. Of principal concern in that survey is the question: “How does one measure the capacity of a system to recognize and discriminate between abstract characteristics of the human phenomenon of music?” The survey catalogues each of the 467 publications along several dimensions. ¹ Sturm determined whether each publication is mainly a review of MGR research, a contribution to evaluation methodology, or a description of experimental work in which an MGR system is built and tested. For each experimental work (438 of 467 publications) Sturm recorded its experimental designs (of which there are 10), datasets (16), and figures of merit (9).

Now that at least 560 more publications have entered the domain, where does the problem of MGR stand? This paper aims to complement and extend the survey of [474] by

¹The data of that survey can be found here: <https://github.com/boblsturm/Music-Genre-Recognition-Survey--1995-2012>



© O. Green, B. Sturm, G. Born, M. Wald-Fuhrmann. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** O. Green, B. Sturm, G. Born, M. Wald-Fuhrmann, “A Critical Survey of Research in Music Genre Recognition”, in *Proc. of the 25th Int. Society for Music Information Retrieval Conf.*, San Francisco, United States, 2024.

exhaustively surveying research published between 2013–2022 related to MGR—as well as any earlier publications discovered to have been missed in Sturm’s original work—such that the two surveys give a comprehensive account up to 2022. It also aims to answer several critical questions. Is the experimental design *Classify* and figure of merit accuracy still the most frequent, despite their noted serious flaws threatening the validity of conclusions drawn from them [470, 471, 561, 562]. Is GTZAN [563] still the most used public dataset, despite its noted faults [400, 467, 471, 564]? How have these faults been considered or even reconsidered in the past decade? How has all this new research in MGR engaged with work that is critical of MGR as a research problem, i.e., [231, 232, 329, 401, 467, 469–471, 473, 475–478, 561, 562, 564–574]? How has all this work engaged with genre theory such as [575–598]?

The next section describes the methodology we use to collect and catalogue publications. Section 3 presents our analyses of this collection along several dimensions. Section 4 gives broad observations and recommendations to guide future work in MGR. The resulting catalogue, bibliography and analysis code are available online. ²

2. METHODOLOGY

We assembled a corpus of 560 publications in the following way. We performed a broad search across *Google Scholar* for publications appearing from 2013 onwards using search terms like ‘music genre’, ‘recognition OR classification “music genre”’. This gave over 67 pages of results that we manually browsed. We supplemented these results with searches of the ISMIR proceedings, TISMIR and arXiv. We added each relevant publication we found to a dedicated Zotero collection, ³ which is a convenient means to gather, share and organize bibliographic data.

For each publication in our collection, we read it and manually enter data into a spreadsheet. As done by Sturm [474], we catalogue each publication according to its type, then note the experimental designs, datasets and figures of merit it uses. Additionally, we note whether each publication cites or engages with two kinds of published work: genre-related work from the social sciences and humanities; and work that is critical of MGR. We note what moti-

² <https://www.kth.se/profile/bobs/page/research-data>

³ <https://zotero.org>

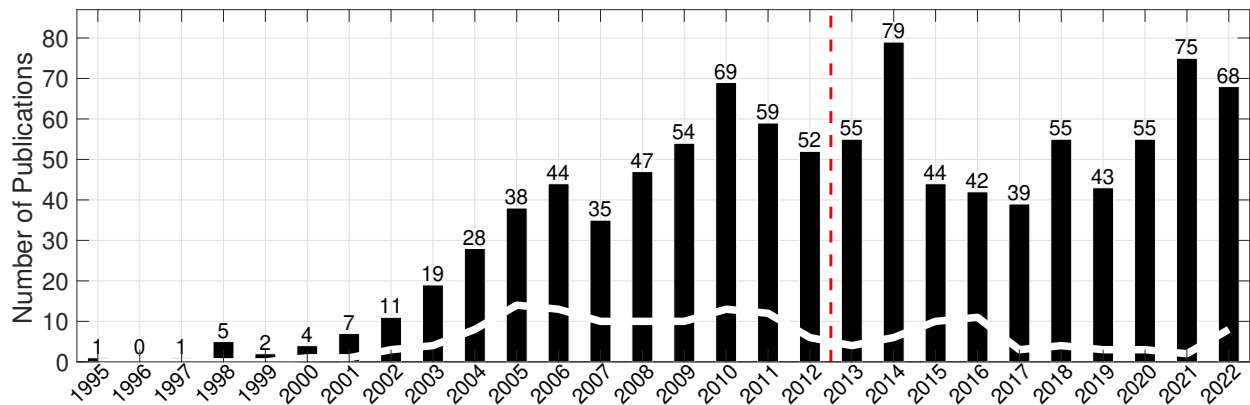


Figure 1. Annual numbers of publications related to music genre recognition between 1995 and 2022. Dashed vertical line demarcates the end date of publications surveyed in Sturm [474]. The present survey adds ten references before 2013 [115, 149, 155, 181, 194, 202, 209, 354, 423, 426], but 550 other publications surveyed herein are published after 2012. The white plot line shows the number of publications appearing at ISMIR or in the Transactions of ISMIR.

vations the paper describes for researching MGR.

To conduct our analyses of the collection, we query entries in our spreadsheet and Zotero library, but also use python and relevant libraries. We export our collection as a BibTex-formatted file, and the spreadsheet as a text file of comma-separated values. In the following subsections we describe in more detail each dimension of our catalogue. (Further details are given in the Supp. Mat.)

Publication Type We assign each publication to at least one of three categories. A *review* publication is concerned with surveying the domain of MGR, e.g., [474, 599]. An *evaluation* publication is concerned with evaluation methodology in MGR, such as proposing a dataset, e.g., [113, 600], experimental design, e.g., [571, 601], or taking a critical look at work in the domain, e.g., [471, 566]. Finally, an *experimental* publication is concerned with engineering and testing MGR systems, e.g., [563, 602].

Dataset For experimental publications, we note what data sets are used, whether private data was used, and the modality of data used: musical audio (waveform or extracted features), symbolic data, or other types (e.g., lyrics, WWW, playlists). We also note if a publication works with non-Western musics.

Experimental Design Following the categorisations described by Sturm [474, p. 32], we note the design(s) used in the experiments described by a publication. These ten designs are: *Classify, Features, Generalize, Robust, Scale, Cluster, Retrieve, Rules, Compose, Eyeball*. (Supp. Mat. S1.5 describes and gives examples of each of these.)

Figure of Merit (FoM) We note the figure(s) of merit used in the evaluation of experiments [474]. The main FoM we look for include *Accuracy, Recall, Precision, F-measure, Receiver Operating Characteristic (ROC)* and the *Confusion Table*. Where a confusion table has been used we note whether or not there is an accompanying interpretative discussion, and whether specific instances of confusion are discussed. (Supp. Mat. S1.6 defines these.)

Referencing In a direction different from Sturm [474], we record if a publication cites at least one of a collection of 26 publications we deem are *critical* of MGR, i.e.,

[231, 232, 329, 401, 467, 469–471, 473, 475–478, 561, 562, 564–574]. We also determine whether the citation is accompanied by any discussion or concrete effects on the experimental design in the publication. We also record if a publication refers to work on musical genre from the social sciences and humanities, i.e., [575–591, 593–598], and record if that citation is accompanied by substantive discussion. Supp. Mat. S1.7 discusses both sets of references.

Motivation for MGR Where authors explicitly state a motivating rationale for MGR work, we record which of four non-exclusive categories they appeal to: *industrial need, public good, coping with information overload*, or are appealing to *precedent*. There is a degree of unavoidable subjectivity in this designation, and so we opt for parsimony and look only for explicit statements of this kind to avoid unwarranted inference. Supp. Mat. S1.8 provides examples of each of these.

3. ANALYSIS AND RESULTS

We now analyze the 560 publications in this survey in relation to the 467 in Sturm [474]. Figure 1 shows the annual number of publications related to MGR since the earliest reference cited in Sturm [474]—Matityaho and Furst [603]. This shows that MGR publications grew to a high point in 2010 after being established in the MIR community a decade earlier as a “flagship problem” of music information retrieval [604]. Thereafter the mean number of publications related to MGR each year is 56.7 (std dev. 13.1). We see that the annual number of publications related to MGR appearing at ISMIR or in its Transactions since 2010 is less than ten in all but two years (2011, 2016). Our survey does not include 28 publications [605–632] because we cannot get access (e.g., behind a paywall), or the language of the paper is not English.

3.1 Publication Types

Of the 560 publications we survey in this paper, we find only nine review articles or book chapters discussing MGR

[98, 193, 247, 323, 325, 339, 467, 474, 493]. Among these, Corrêa and Rodrigues [98] reviews MGR using symbolic data. Kostek [247] is a review of MIR and has a section about MGR; and Tzanetakis [493] is a book chapter reviewing music informatics generally, where music genre appears in one section. Sturm [467] reviews all publications using the GTZAN dataset up to 2012; and Sturm [474] is the survey we extend in this paper.

We find 19 of our 560 publications that primarily discuss evaluation in MGR research [113, 199, 231, 232, 329, 341, 367, 400, 401, 424, 467, 469–471, 473, 474, 476–478]. In addition to those already described in Sec. 2, Palmason et al. [341] investigates the agreement of music genre ground truth between different stakeholders. Porter et al. [367] discusses enriching the AcousticBrainz audio feature dataset [633] using metadata collected from a variety of online sources, include genre information. Schreiber [424] extends the Million Song Dataset; and Defferrard et al. [113] introduces the FMA dataset. Finally, Hossain and Al Marouf [199] discusses the creation of a dataset of song lyrics exemplifying different genres of Bengali music.

Of the 560 publications we survey in this paper, we find 545 that make experimental contributions. Of these, the next three subsections discuss the datasets, experimental designs, and figures of merit used in this subset of publications, comparing and contrasting with the previous survey [474]. We then go further than Sturm [474]. Subsection 3.7 looks at how all 560 publications we survey engage with work critical of MGR. Subsection 3.8 investigates the extent to which they engage with music genre theory. The penultimate subsection 3.5 looks at how MGR is being motivated as a research problem. Subsection 3.6 looks at the kinds of venues at which MGR research is being published.

3.2 Datasets

Sturm [474] finds the GTZAN dataset [563] from 2002 is the most used public dataset, appearing in 100 out of 435 publications with an experimental component. We find that GTZAN remains the most frequently used dataset, appearing in 254 of 545 publications that have an experimental component. Some publications use GTZAN for learning bases, which are then used for building MGR systems tested in other datasets, e.g., Jao et al. [213] and Markov and Matsui [295]. Some publications we survey use only a portion of GTZAN. For instance, Agarwal et al. [5] uses only five of ten classes; and Rajesh and Bhalke [390] uses only two. Others add classes to GTZAN, e.g., Iloga et al. [205, 206] adds Cameroonian music, Conceicao et al. [95] adds music from Brazil, Wibowo and Wihayati [519] adds Malaysian Dangdut music, and Shashirekha [440] adds songs sung in an Indian language (Kannada). Moving briefly to the entire collection we survey, we find considerations of music from non-Western traditions to appear in only 61 of the 560 publications; Sturm [474] finds 47 in its survey of 467 publications.

Some publications analyze GTZAN as a dataset. Flexer [150] analyzes hubs in GTZAN, and tests methods of outlier detection using it. Lu et al. [286, 287] attempt to auto-

matically find the faults in the dataset identified in Sturm [467]. Rodriguez-Algarra et al. [400] investigates why a particular MGR system performs so well on GTZAN, and finds infrasonic information confounded with labels—more formally explored in Rodriguez-Algarra et al. [401]. Kang and Lin [224] looks at inferring taxonomies of classes in datasets, including GTZAN. Lu et al. [286, 287] use GTZAN as a testbed for anomaly detection.

The next four most popular datasets we find are ISMIR2004 [634] from 2004 (appearing in 50 publications), FMA [635] from 2016 (36), the Million Song Dataset [600] from 2011 (32), and the Latin Music Dataset [636] from 2008 (23). We find that data that is not publicly available (e.g., in-house data, or undisclosed data) appears in 176 of 545 publications. Of those, we find 146 of them exclusively use non-publicly available data.

The predominant data modality in our catalogue is acoustic (or features extracted from acoustic data), which appears in 482 of 545 publications with an experimental component. Sturm [474] finds 344 of 435 publications use such a modality. We find symbolic modalities are used in 40 publications, while Sturm [474] finds them used in 81 publications. We find 11 publications use both modalities [2, 23, 187, 189, 209, 270, 359, 360, 362, 507, 556]. Other modalities (e.g., lyrics, WWW, playlists) appear used in 66 of our publications, while Sturm [474] finds these used in 27 publications.

3.3 Experimental design

The three most used experimental designs we find in the 545 publications we survey with an experimental component match those found by Sturm [474]. The most used design in both is *Classify*: we find 514 publications use this, and of those 264 exclusively use this design; Sturm [474] finds this appears in 397 of 435 publications with an experimental component. The second most used design is *Feature*, which appears in 145 of 545 publications we survey; Sturm [474] finds this appears in 142 of 435 publications. The third most used design is *Generalize*, which appears in 127 of 545 publications we survey; Sturm [474] finds it appears in 69 of 435 publications.

The two least-used designs we find are the same as in Sturm [474]: *Rules* and *Compose*. Among our 545 papers with an experimental component, we find *Rules* appears in six publications [63, 70, 97, 209, 246, 402]. For instance, Campobello et al. [63] derive an analytic formulae for GTZAN genres whereby specific feature values extracted from an audio signal are used to compute the relevance of a class. We find the *Compose* design appears in five [175, 209, 476, 485, 489]. For instance, Sturm [476] inspects the correspondence between randomly generated rhythmic patterns classified with high confidence by a state-of-the-art system trained in the BALLROOM dataset [637], and the classes in that dataset.

3.4 Figure of merit

We find accuracy is the figure of merit that appears the most: 449 out of 545 publications with an experimental

component. Sturm [474] finds this appears in 385 of 435 publications. The confusion table is the next most common figure of merit, appearing in 193 publications surveyed here. In those publications with a confusion table, we find no discussion about it in 78 publications—that is, it is merely presented as a table without interpretation. When a confusion table is discussed, musical motivations for confusions are often given. For example, Chathuranga et al. [73] writes, “[In our confusion table] Rock music is mostly misclassified as country and blue. This is due to the facts that rock music and country music have similar roots and rock music came from a combination of country music and rhythm and blues.” Chen and Ramadge [77] writes, “Rock music is easily confused with other genres—perhaps because of its broad nature.”

Specific instances of confusions are only discussed in nine papers [63, 201, 231, 232, 341, 467, 470, 472, 475]. For instance, Sturm [467] and Campobello et al. [63] show tables of confusions made by classifier for specific excerpts in GTZAN, e.g., Sturm [467] shows GTZAN country excerpt 69 “My Heroes Have Always Been Cowboys” by Willie Nelson is mislabeled “Classical”, and Campobello et al. [63] shows GTZAN filename “country.00069” is mislabeled “Blues”, “Classical” and “Rock”. Sturm et al. [472] do the same but for the Latin Music Dataset. Hsu et al. [201] look at four specific music recordings from a test set they create and inspect how classifications of them differ between systems they test. Finally, Kereliuk et al. [231, 232] and Sturm [470, 475] make use of a set of ten different songs and show how each can be confidently classified in any GTZAN class.

3.5 Justifications of MGR

How is MGR as a research problem being justified? We noted two major shades of justification where one was offered. 155 papers were found that explicitly invoke imagined applications for the music industry (106) or end users (49). In 150 of these papers, applications of MGR were presented as useful or necessary in due contemporary information overload. Conversely, 103 papers were noted to make no direct utilitarian appeal for MGR work, but instead to call upon precedent: MGR problems are important because there has been work on MGR.

In the 150 papers that invoke information overload as a problem that MGR can help solve, a common presumption is that there are problematic quantities of unlabelled musical data that would be more tedious or error-prone to organize by hand than with MGR, e.g [174, 439]. Sometimes, the urgency of dealing with such a problem is emphasised. For example: “a lot of music data has become available recently ... in order for users to benefit from them, an efficient music information retrieval technology is necessary.” [294]; “Given the vast number of current collections, automated genre classification is critical for music organization ...” [376]

Publications seldom motivate through learning something about *music* rather than classifier performance. [301] sets out to understand the temporalities of musical change

over a 50 year period. [329] argues that MGR classifiers can be repurposed towards greater understanding of genre as musico-social. [189, 505] argue that more interpretable models more useful for musicologists and listeners. [348] investigates the potential of MGR for studying underrepresented traditions. [140] examines the ‘tenuous’ relationship between rhythmic similarity and genre.

3.6 Venues for MGR publications

Of the 560 publications we review, 350 are conference papers and 156 appear in journals. The most common venue for MGR publications is the ISMIR conference (43 papers). The next most common conferences are *Int. Conf. Acoustics Speech and Signal Processing* (ICASSP) (12), *Int. Joint Conf. Neural Networks* (IJCNN) (9), and *Inter-speech* (5). Six publications appear at the music computing conferences ICMC, SMC and DaFx. The most common journals were *IEEE Access* (7), *Int. Research Journal of Engineering and Technology* (6), then *IEEE Signal Processing Letters*, *Int. Journal of Computer Applications*, *Journal of Intelligent Information Systems*, *IEEE Trans. on Multimedia*, *Journal of New Music Research*, *Trans. of the Int. Society for Music Information Retrieval*, *Expert Systems with Applications*, and *Applied Soft Computing* (4 publications each).

After stripping edition indicators from conference names, we estimate that papers appeared at 235 unique conferences, of which 195 hosted a single paper in our corpus. Similarly, the 156 journal articles we reviewed appeared across 101 journals, 74 of which hosted a single article from our corpus. 82 publications appear in conferences or journals under the umbrellas of the ACM and IEEE, including ICASSP and WASPAA. In addition to 9 more PhD theses containing work related to MGR [2, 23, 160, 183, 336, 501, 518, 545] we reviewed 14 master’s theses [11, 35, 64, 112, 220, 262, 289, 308, 333, 374, 453, 465].

Elsevier publishes the greatest proportion of the journals encountered (11), followed by Springer (9), IEEE (8), MDPI (5), Hindawi (4), IET (3) and ACM (3). 20 publications were on ArXiv or similar pre-print hosting providers with no corresponding official publication. Many publications appear in venues not specifically concerned with music, audio or informatics, but with computing topics more generally or—more general still—with topics like ‘engineering’ or ‘technology’. Some publications appear in venues apparently unrelated to music informatics. For example, [176] appears in the *International Journal of Early Childhood Special Education*. Others appear in venues whose existence we could not confirm, e.g. [193] was apparently presented at a 2018 ACM Symposium on Neural Gaze Detection of which we can find no online trace.

3.7 Engagement with work critical of MGR

We now turn to another aspect not explored in Sturm [474]: How does research in MGR engage with work that is critical of MGR? How many of the 560 publications we survey cite 26 critical publications [231, 232, 329, 401, 467, 469–

471, 473, 475–478, 561, 562, 564–574]? Do any implement proposed recommendations or alternatives? We find that only 163 of the 560 publications cite at least one of these critical references; and of these, only 77 engage in some way with the critique. Such engagement can be simply applying artist filtering. It can also be motivating the use of a specific experimental design.

Let us look specifically at criticism around GTZAN [563], the most-used dataset in MGR research. Ten years after its creation, this dataset was carefully analyzed by Sturm [467, 471, 564], resulting in a catalogue of its faults and an index of its contents. Kereliuk et al. [231, 232] created and used the first partitioning of GTZAN that considers its contents.⁴ (See Supp. Mat. S2 for an overview of the on-line availability of these materials.)

Considering the faults of GTZAN have been known since 2012, let us focus on the 250 papers published after 2012 that use GTZAN with a *Classify* experimental design reporting accuracy as a figure of merit. Of these, we find only 62 acknowledge the existence of faults in GTZAN, and 46 of those essentially ignore or dismiss them. For instance, Sigtia and Dixon [447], Nanni et al. [320], Jeong and Lee [218], Senac et al. [428] and Palmason et al. [340] are five papers that dismiss consideration of the faults by appealing to the popularity of GTZAN as a benchmark dataset: “Although the GTZAN dataset has some shortcomings [564], it has been used as a benchmark for genre classification tasks” [447]. Others claim that their experimental results are not harmed by such problems, e.g., “Despite [its faults], we still used [GTZAN] because these small problems can not seriously damage our results” [205]. We find 15 publications use the fault-filtered partitions of Kereliuk et al. [231, 232]: [66, 81, 144, 145, 218, 236, 267, 271, 302, 305, 329, 349, 364, 540, 554]. Foleiss and Tavares [152] create their own partitioning following Sturm [467], which was then used by Ng et al. [327] and Cai and Zhang [62]. Three other publications [15, 63, 374] acknowledge the faults in GTZAN and perform their own fault-filtering and partitioning.

The fact that 188 of these 250 publications using GTZAN do not mention its faults could be partly explained by the fact that websites linking to the dataset make no mention of them. At least up to March 20 2022, the original source of GTZAN⁵ makes no mention of any faults or of the cataloguing work of Sturm [467, 471, 564]. There currently exist several online copies of GTZAN (or features computed from the dataset), but none of these mention faults; and at this time we find only two online repositories of GTZAN that mention faults (See Supp. Mat. S2).

Another criticized aspect of MGR research is its use of *Classify* as an experimental design. Sturm [470, 471, 473, 478, 561, 562] argue that this design is essentially a “horse show”: systems are tasked with tapping their hooves the correct number of times, but no reliable measurement of musical intelligence can be made without controlling for numerous independent variables. While the survey in

Sturm [474] finds *Classify* appears in 91.3% of its surveyed publications, the present survey finds it appears in 94.3%. Furthermore, we find 264 publications *only* use *Classify*. While we see at least some work in MGR has cited and engaged with the faults in GTZAN, very few publications in the present survey (outside of those by Sturm and collaborators) meaningfully engage with the criticism of *Classify*. To the best of our knowledge, there are no publications that dispute the argument of Sturm [470, 471, 473, 478, 561, 562]; and we find only 17 publications citing those six publications and engaging with them in any meaningful way when it comes to experimental design [47, 48, 63, 67, 98, 140, 175, 244, 286, 327, 329, 374, 375, 393, 421, 507, 518], e.g., cautiously interpreting results of classification, or motivating additional experimental designs.

3.8 Engagement with genre theory

We now turn to the question: how have the 560 publications in our survey engaged with genre theory from the social sciences and humanities? We find 36 references to such work citing 23 sources [575–598]. Of these 36 publications, 10 go further than just citation [139, 146, 219, 329, 341, 375, 421, 453, 469, 471]. Useful indicators of the ways in which musical genre is a more complex concept than just distributions of acoustic or other features are scattered across these contributions. The following key points emerge from engagements with genre theory in our corpus:

Relational The interrelationships between genres are crucial to understanding them, yet more complex than can be captured through simple taxonomies [329, 341, 375, 469, 471].

More-than-sonic The character of genres is determined not only through sonic traits but that other modalities can be of crucial importance, often as proxies for the social roots of genres [146, 329, 375, 421].

Social The relationships between genres and social formations / identities is complex and bidirectional: genres can articulate identities, but genres can also be used as part of demarcating social groupings. The agency in defining and consolidating genre terms is distributed across different social planes, including the institutions of industry, as well as musicians, critics and fans [219, 329, 341, 421].

Perspectival Genres and their relationships can be understood quite differently by different groups of people [375, 469, 471].

Dynamic No aspect of musical genre stands still. Their salient sonic and other features, interrelationships, connections to social formations are all in constant, unpredictable motion. Crucially this means that the association of particular musical texts and tastes with genres is also subject to change [329, 471].

⁴ Both the catalogue of GTZAN and the fault-filtered partition are available here: <https://github.com/boblsturm/GTZAN>.

⁵ <http://marsyas.info/downloads/datasets.html>.

4. DISCUSSION

Based on a survey of 560 MGR publications from 2013–2022, we find some continuity with the previous survey by Sturm [474]. GTZAN, *Classify* and *Accuracy* remain respectively the most widely used dataset, experiment and figure of merit. Despite an increase in the number of public datasets available to MGR researchers, we also see that the proportion of publications dealing with non-Western musical formations has not changed appreciably. Although the use of alternative modalities of data (e.g., lyrics, WWW, playlists) has roughly doubled, such work remains a minority and treating MGR as an audio-similarity problem still prevails. However, our survey goes further to find that MGR work has by and large not engaged with any critique of its accepted methodologies, critique of the research task itself, or of work in the social sciences and humanities related to genre.

We note that the engagement that there is has introduced to MGR key facets of the challenges of studying musical genre, which warrant greater consideration—we commend the introductory chapter of [577] as a comprehensive overview. Crucially, each of the factors outlined in Sec. 3.8 not only provide key pointers towards threats to validity for MGR [638] but also indicate that for MIR to contend with genre as a *musical* topic, a greater diversity of approaches is called for [639]. However, given the quantity of the surveyed publications that appear outside ISMIR or core MIR venues, it is not certain whether MGR remains a “flagship problem” of music information retrieval [604].

Has it, rather, become autonomous of MIR, as a convenient downstream task for computer scientists that has the appearances of addressing a domain-specific useful problem? Our reading of the given motivations for MGR research found in this corpus supports the plausibility of this interpretation, given the frequency of appeals to vaguely described industrial and user applications. One way to investigate this further would be through a bibliometric analysis of this corpus, geared towards identifying possible clusters of work through co-citation or collaboration.⁶ A possibility is that MIR specialists have shifted their attention to auto-tagging and away from MGR, which as a ‘superset’ problem of MGR, we do not cover here. However, an interesting area for follow-up research could be to perform a similar survey of auto-tagging research along with a bibliometric comparison, which may shed some light on the movement of research within MIR.

Nevertheless, we put forward the normative position that *music* informatics researchers should be oriented to *musical* questions open to investigating their complexities in collaboration with music scholars. Doing so likely involves shifts in how this research is pursued. Currently, what dominates is exploratory analysis where progress is assessed through benchmarks. [641] provides a framework for assessing the suitability of such ‘outcome reasoning’ along dimensions of measurement, adaptability, re-

silience and compatibility with stakeholder beliefs. These are telling questions, as it’s not clear that for much MGR research who the stakeholders are. If one group of potential stakeholders is other music researchers, rather than the imagined needs of music platforms or their users, then this points to more *explanatory* work in MGR. However, benchmark-driven predictive modelling need not be abandoned. In [642] the authors point to ways in which benchmark-driven work can be incorporated and redirected towards richer ends and [643] shows how predictive models can be integrated into contemporary approaches to causal inference, suited to theory-driven, explanatory registers of research.

5. CONCLUSION

We close with some recommendations. Broadly, MGR suffers from threats to validity [638] that warrant more attention. Some progress could be made through a greater role for theory in MGR, both through authors being more explicit about the theoretical perspective on genre at work in a given study, as well as deeper engagement with theories of musical genre.

In particular, a close and critical reading of the introductory discussion of [577] in terms of its implications for MIR genre research could be generative for the field. Specifically, it is the sociality, temporality and heterogeneity of genres that are least addressed in the work we have surveyed, and these bring interesting challenges. One fruitful direction to engaging with the social can be found in [644]: by acknowledging that culture is present in every part of the MIR ‘value-chain’ [645], the authors propose a technical intervention on recommender system design in pursuit of a normative social outcome (‘commonality’) often considered to be outside the scope of engineering concerns. Some of the surveyed work already moves towards dealing with genre as temporal [301], and social-temporal [329]. This suggests a possible intersection with work in music and cultural evolution [646], which could serve as a constructive ‘interface’ between MIR and music studies.

Finally, to contend with the heterogeneity of genres means dealing not only with their sonic variability, but also variability across the many other dimensions that may define a particular genre for a particular aesthetic coalition. On the first point, we would recommend more genre-specific computational-musicological work like [647–649] to cast light on the relationships between computed features and aesthetic saliences for groups of listeners. More fine-grained, genre-specific datasets as in [276] might help here. On the second point, engaging with ‘live’ genres in motion implies the need for work in nonexperimental settings [650] that can cope with diverse, noisy and incomplete data.

6. ACKNOWLEDGMENTS

This work was supported by the European Research Council under the European Union’s Horizon 2020 research and innovation programme (Grant agreements No. 864189 and No. 101019164).

⁶ Open scholarly databases such as OpenAlex [640] could automate at least some of this, although we note that its coverage of references for the papers in this corpus doesn’t extend to around half of its ISMIR papers.

7. REFERENCES

- [1] M. Abbasi Layegh, S. Haghypour, and Y. Najafi Sarem, "Classification of the Radif of Mirza Abdollah a Canonic Repertoire of Persian Music Using SVM Method," *Gazi University Journal of Science Part A Engineering and Innovation*, vol. 1, no. 4, pp. 57–66, 4 2013.
- [2] J. Abeßer, "Automatic Transcription of Bass Guitar Tracks applied for Music Genre Classification and Sound Synthesis," Ph.D. dissertation, Technischen Universität Ilmenau, Oct. 23, 2014.
- [3] Adiyansjah, A. A. S. Gunawan, and D. Suhartono, "Music recommender system based on genre using convolutional recurrent neural networks," in *The 4th International Conference on Computer Science and Computational Intelligence (ICCSCI 2019) : Enabling Collaboration to Escalate Impact of Research Results for Society*, vol. 157, Jan. 1, 2019, pp. 99–109.
- [4] D. Afchar, R. Hennequin, and V. Guigue, "Learning Unsupervised Hierarchies of Audio Concepts," in *Proceedings of the 23rd International Society for Music Information Retrieval Conference*, Bengaluru, India: ISMIR, Dec. 4, 2022, pp. 427–436.
- [5] P. Agarwal, H. Karnick, and B. Raj, "A comparative study of indian and western music forms," in *Proceedings of the 14th International Society for Music Information Retrieval Conference*, 2013.
- [6] N. Agera, S. Chapaneri, and D. Jayaswal, "Exploring textural features for automatic music genre classification," in *2015 International Conference on Computing Communication Control and Automation*, Feb. 2015, pp. 822–826.
- [7] M. Agrawal and A. Nandy. "A novel multimodal music genre classifier using hierarchical attention and convolutional neural network." arXiv: 2011.11970 [cs, eess]. (Nov. 24, 2020), [Online]. Available: <http://arxiv.org/abs/2011.11970> (visited on 03/07/2023), preprint.
- [8] R. L. Aguiar, Y. M. Costa, and C. N. Silla, "Exploring data augmentation to improve music genre classification with ConvNets," in *2018 International Joint Conference on Neural Networks (IJCNN)*, Jul. 2018, pp. 1–8.
- [9] A. N. Ahmad, C. Sekhar, and A. Yashkar, "Music Genre Classification Using Music Information Retrieval and Self Organizing Maps," in *Proceedings of the Third International Conference on Soft Computing for Problem Solving*, M. Pant, K. Deep, A. Nagar, and J. C. Bansal, Eds., New Delhi: Springer India, 2014, pp. 625–634.
- [10] F. Ahmed, P. P. Paul, and M. Gavrilova, "Music genre classification using a gradient-based local texture descriptor," in *Intelligent Decision Technologies 2016*, I. Czarnowski, A. M. Caballero, R. J. Howlett, and L. C. Jain, Eds., Cham: Springer International Publishing, 2016, pp. 455–464.
- [11] R. Ajoodha, "Automatic music genre classification," M.S. thesis, University of the Witwatersrand, Johannesburg, 2014.
- [12] R. Ajoodha, R. Klein, and B. Rosman, "Single-labelled music genre classification using content-based features," in *2015 Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference (PRASA-RobMech)*, Nov. 2015, pp. 66–71.
- [13] H. Akalp, E. Furkan Cigdem, S. Yilmaz, N. Bolucu, and B. Can, "Language representation models for music genre classification using lyrics," in *2021 International Symposium on Electrical, Electronics and Information Engineering*, ser. ISEEIE 2021, New York, NY, USA: Association for Computing Machinery, Jul. 20, 2021, pp. 408–414.
- [14] A.-K. Al-Tamimi, M. Salem, and A. Al-Alami, "On the use of feature selection for music genre classification," in *2020 Seventh International Conference on Information Technology Trends (ITT)*, Nov. 2020, pp. 1–6.
- [15] A. Alexandridis, E. Chondrodima, G. Paivana, M. Stogiannos, E. Zois, and H. Sarimveis, "Music genre classification using radial basis function networks and particle swarm optimization," in *2014 6th Computer Science and Electronic Engineering Conference (CEEC)*, Sep. 2014, pp. 35–40.
- [16] M. A. Ali and Z. A. Siddiqui, "Automatic music genres classification using machine learning," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 8, 2017.
- [17] S. Allamy and A. L. Koerich, "1D CNN architectures for music genre classification," in *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*, Dec. 2021, pp. 01–07.
- [18] M. A. Al Mamun, I. Kadir, A. S. A. Rabby, and A. Al Azmi, "Bangla music genre classification using neural network," in *2019 8th International Conference System Modeling and Advancement in Research Trends (SMART)*, Nov. 2019, pp. 397–403.
- [19] F. C. F. Almeida, G. Bernardes, and C. Weiss, "Mid-level Harmonic Audio Features for Musical Style Classification," in *Proceedings of the 23rd International Society for Music Information Retrieval Conference*, Bengaluru, India: ISMIR, Dec. 4, 2022, pp. 210–217.

- [20] P. Alonso-Jiménez, D. Bogdanov, J. Pons, and X. Serra, “Tensorflow Audio Models in Essentia,” in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2020, pp. 266–270.
- [21] J. Andén and S. Mallat, “Deep Scattering Spectrum,” *IEEE Transactions on Signal Processing*, vol. 62, no. 16, pp. 4114–4128, Aug. 2014.
- [22] N. Andreas, P. Maria, R. Ioannou, N. Petkov, and C. N. Schizas, “A machine learning approach for clustering western and non-western folk music using low-level and mid-level,” in *Proceedings 6th International Workshop on Machine Learning and Music*, 2013, pp. 55–58.
- [23] A. Anglade, “Logic-based Modelling of Musical Harmony for Automatic Characterisation and Classification,” Ph.D. dissertation, Queen Mary University of London, Apr. 30, 2014.
- [24] P. G. Antunes, D. M. de Matos, R. Ribeiro, and I. Trancoso. “Automatic fado music classification.” arXiv: 1406.4447 [cs]. (Jun. 17, 2014), [Online]. Available: <http://arxiv.org/abs/1406.4447> (visited on 03/02/2023), preprint.
- [25] J. Arenas-Garcia, K. B. Petersen, G. Camps-Valls, and L. K. Hansen, “Kernel Multivariate Analysis Framework for Supervised Subspace Learning: A Tutorial on Linear and Kernel Multivariate Methods,” *IEEE Signal Processing Magazine*, vol. 30, no. 4, pp. 16–29, Jul. 2013.
- [26] T. Arjannikov and J. Zhang, “An association-based approach to genre classification in music,” in *Proceedings of the 15th International Society for Music Information Retrieval Conference*, Taipei, Taiwan, Oct. 27, 2014.
- [27] T. Arjannikov and J. Z. Zhang, “An empirical study on structured dichotomies in music genre classification,” in *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, Dec. 2015, pp. 493–496.
- [28] T. Arjannikov and J. Z. Zhang, “Do nested dichotomies help in automatic music genre classification? An empirical study,” in *Proceedings of the International Computer Music Conference (ICMC)*, 2016.
- [29] M. G. Armentano, W. A. De Noni, and H. F. Cardoso, “Genre classification of symbolic pieces of music,” *Journal of Intelligent Information Systems*, vol. 48, no. 3, pp. 579–599, Jun. 1, 2017.
- [30] K. Aryafar and A. Shokoufandeh, “Multimodal sparsity-eager support vector machines for music classification,” in *2014 13th International Conference on Machine Learning and Applications*, Dec. 2014, pp. 405–408.
- [31] M. Astefanoaei and N. Collignon, “Hyperbolic embeddings for music taxonomy,” in *Proceedings of the 1st Workshop on NLP for Music and Audio (NLP4MusA)*, 2020, pp. 38–42.
- [32] Y. Atahan, A. Elbir, A. Enes Keskin, O. Kiraz, B. Kirval, and N. Aydin, “Music genre classification using acoustic features and autoencoders,” in *2021 Innovations in Intelligent Systems and Applications Conference (ASYU)*, Oct. 2021, pp. 1–5.
- [33] K. M. Athulya and S. Sindhu, “Deep learning based music genre classification using spectrogram,” in *Proceedings of the International Conference on IoT Based Control Networks & Intelligent Systems - ICICNIS 2021*, Rochester, NY, Jul. 10, 2021.
- [34] H. Bahuleyan. “Music genre classification using machine learning techniques.” arXiv: 1804.01149 [cs, eess]. (Apr. 3, 2018), [Online]. Available: <http://arxiv.org/abs/1804.01149> (visited on 03/03/2023), preprint.
- [35] V. Bajpai, “Evaluation of state of the art for genre classification in large datasets,” M.S. thesis, Sep. 15, 2018.
- [36] M. Banitalebi-Dehkordi and A. Banitalebi-Dehkordi, “Music Genre Classification Using Spectral Analysis and Sparse Representation of the Signals,” *Journal of Signal Processing Systems*, vol. 74, no. 2, pp. 273–280, Feb. 1, 2014.
- [37] B. K. Baniya, D. Ghimire, and J. Lee, “Evaluation of different audio features for musical genre classification,” in *SiPS 2013 Proceedings*, Oct. 2013, pp. 260–265.
- [38] B. K. Baniya, D. Ghimire, and J. Lee, “Music Genre Classification Based on Timbral Texture and Rhythmic Content Features,” in *Proceedings of the 39th Korean Information Processing Society Spring Conference*, Korea, 2013.
- [39] B. K. Baniya, J. Lee, and Z.-N. Li, “Audio feature reduction and analysis for automatic music genre classification,” in *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Oct. 2014, pp. 457–462.
- [40] B. K. Baniya, D. Ghimire, and J. Lee, “A novel approach of automatic music genre classification based on timbral texture and rhythmic content features,” in *16th International Conference on Advanced Communication Technology (ICACT)*, Feb. 2014, pp. 96–102.
- [41] B. K. Baniya, D. Ghimire, and J. Lee, “Automatic music genre classification using timbral texture and rhythmic content features,” in *2015 17th International Conference on Advanced Communication Technology (ICACT)*, Jul. 2015, pp. 434–443.
- [42] B. K. Baniya and J. Lee, “Importance of audio feature reduction in automatic music genre classification,” *Multimedia Tools and Applications*, vol. 75, no. 6, pp. 3013–3026, Mar. 1, 2016.

- [43] N. Bassiou, C. Kotropoulos, and A. Papazoglou-Chalikiakias, "Greek folk music classification into two genres using lyrics and audio via canonical correlation analysis," in *2015 9th International Symposium on Image and Signal Processing and Analysis (ISPA)*, Sep. 2015, pp. 238–243.
- [44] J. F. Bernabeu Briones, C. Pérez-Sancho, P. J. Ponce de León Amador, J. M. Iñesta, and J. Calvo-Zaragoza, "A multimodal genre recognition prototype," presented at the III Workshop de Reconocimiento de Formas y Análisis de Imágenes, WSRFAI, Madrid, Spain, Sep. 2013, pp. 13–16.
- [45] J. K. Bhatia, R. D. Singh, and S. Kumar, "Music genre classification," in *2021 5th International Conference on Information Systems and Computer Networks (ISCON)*, Oct. 2021, pp. 1–4.
- [46] A. Bhowmik and A. E. Chowdhury, "Genre of bangla music: A machine classification learning approach," *AIUB Journal of Science and Engineering (AJSE)*, vol. 18, no. 2, pp. 66–72, 2 Aug. 31, 2019.
- [47] D. Bisharad and R. H. Laskar, "Music genre recognition using residual neural networks," in *TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON)*, Oct. 2019, pp. 2063–2068.
- [48] D. Bisharad and R. H. Laskar, "Music genre recognition using convolutional recurrent neural network architecture," *Expert Systems*, vol. 36, no. 4, e12429, 2019.
- [49] Z. Bodo and E. Szilágyi, "Connecting the last.fm dataset to LyricWiki and MusicBrainz. Lyrics-based experiments in genre classification," *Acta Universitatis Sapientiae Informatica*, vol. 10, pp. 158–182, Dec. 20, 2018.
- [50] D. Bogdanov, A. Porter, P. Herrera Boyer, and X. Serra, "Cross-collection evaluation for music classification tasks," in *Proceedings of the 17th International Society for Music Information Retrieval Conference*; ISMIR, 2016, pp. 379–385.
- [51] D. Bogdanov, M. Won, P. Tovstogan, A. Porter, and X. Serra, "The MTG-jamendo dataset for automatic music tagging," in *Proceedings International Conference on Machine Learning, ICML*, 2019.
- [52] D. Bogdanov, A. Porter, H. Schreiber, J. Urbano, and S. Oramas, "The AcousticBrainz Genre Dataset: Multi-Source, Multi-Level, Multi-Label, and Large-Scale," in *Proceedings of the 20th International Society for Music Information Retrieval Conference*, Delft, The Netherlands: ISMIR, Nov. 4, 2019, pp. 360–367.
- [53] P. Boonmatham, S. Pongpinigpinyo, and T. Soonklang, "Musical-scale characteristics for traditional Thai music genre classification," in *2013 International Computer Science and Engineering Conference (ICSEC)*, Sep. 2013, pp. 227–232.
- [54] V. Bruni, M. L. Cardinali, and D. Vitulano, "An MDL-Based Wavelet Scattering Features Selection for Signal Classification," *Axioms*, vol. 11, no. 8, p. 376, 8 Aug. 2022.
- [55] G. Brunner, Y. Wang, R. Wattenhofer, and S. Zhao, "Symbolic music genre transfer with CycleGAN," in *2018 IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI)*, Nov. 2018, pp. 786–793.
- [56] Z. Bu, H. Zhang, and X. Zhu. "GAFX: A General Audio Feature eXtractor." arXiv: 2207.09145 [cs, eess]. (Jul. 19, 2022), [Online]. Available: <http://arxiv.org/abs/2207.09145> (visited on 03/24/2023), preprint.
- [57] A. Buchmüller and C. Gerloff, "Music genre classification using artificial neural networks," in *Learning Deep: Perspectives on Deep Learning Algorithms and Artificial Intelligence*, B. Säfken, A. Silbersdorff, and C. Weisser, Eds., Universitätsverlag Göttingen, 2020, pp. 127–144.
- [58] A. Budhrani, A. Patel, and S. Ribadiya, "Music2vec: Music genre classification and recommendation system," in *2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, Nov. 2020, pp. 1406–1411.
- [59] R. d. D. Bulos, G. F. Go, G. O. Ling, T. C. Uy, and L. J. Yap, "Predictive Analysis Using Data Mining Techniques and SQL," in *Proceedings of the DLSU Research Congress*, De La Salle University, Manila, Philippines, 2014.
- [60] H. Cai, T. Pu, Y. Luo, and X. Zhou, "Music genre prediction based on machine learning," in *2021 IEEE International Conference on Artificial Intelligence and Industrial Design (AIID)*, May 2021, pp. 198–201.
- [61] X. C. Cai and H. Z. Zhang. "Fisher discriminative embedding low-rank sparse representation for music genre classification." (2022), [Online]. Available: https://assets.researchsquare.com/files/rs-1719236/v1_covered.pdf?c=1654629348, preprint.
- [62] X. Cai and H. Zhang, "Music genre classification based on auditory image, spectral and acoustic features," *Multimedia Systems*, vol. 28, no. 3, pp. 779–791, Jun. 1, 2022.
- [63] G. Campobello, D. Dell'Aquila, M. Russo, and A. Segreto, "Neuro-genetic programming for multi-genre classification of music content," *Applied Soft Computing*, vol. 94, p. 106488, Sep. 1, 2020.
- [64] F. Capó Clar, "Impact of audio degradation on music classification," M.S. thesis, Accepted: 2014-09-25T10:05:10Z Publisher: Universitat Politècnica de Catalunya, Barcelona, Jul. 10, 2014.

- [65] R. V. Casaña-Eslava, I. H. Jarman, S. Ortega-Martorell, P. J. G. Lisboa, and J. D. Martín-Guerrero, "Music genre profiling based on fisher manifolds and probabilistic quantum clustering," *Neural Computing and Applications*, vol. 33, no. 13, pp. 7521–7539, Jul. 1, 2021.
- [66] R. Castellon, C. Donahue, and P. Liang, "Codified audio language modeling learns useful representations for music information retrieval," in *Proceedings of the 22nd International Society for Music Information Retrieval Conference*, Online: ISMIR, Nov. 7, 2021, pp. 88–96.
- [67] J. R. Castillo and M. J. Flores, "Web-based music genre classification for timeline song visualization and analysis," *IEEE Access*, vol. 9, pp. 18 801–18 816, 2021.
- [68] H. C. Ceylan, N. Hardalaç, A. C. Kara, and H. Firat, "Automatic music genre classification and its relation with music education," *World Journal of Education*, vol. 11, no. 2, pp. 36–45, 2021.
- [69] W. H. Chak, N. Saito, and D. Weber, "The scattering transform network with generalized morse wavelets and its application to music genre classification," in *2022 International Conference on Wavelet Analysis and Pattern Recognition (ICWAPR)*, Sep. 2022, pp. 25–30.
- [70] Y.-H. Chang and S.-N. Yao, "Artificial Intelligence on the Identification of Beiguan Music," *Archives of Acoustics*, vol. 46, no. 3, pp. 471–478, 2021.
- [71] P.-C. Chang, Y.-S. Chen, and C.-H. Lee, "MS-SincResNet: Joint learning of 1D and 2D kernels using multi-scale SincNet and ResNet for music genre classification," in *Proceedings of the 2021 International Conference on Multimedia Retrieval*, ser. ICMR '21, New York, NY, USA: Association for Computing Machinery, Sep. 1, 2021, pp. 29–36.
- [72] S. Chapaneri, R. Lopes, and D. Jayaswal, "Evaluation of music features for PUK kernel based genre classification," in *International Conference on Advanced Computing Technologies and Applications (ICACTA)*, vol. 45, Jan. 1, 2015, pp. 186–196.
- [73] D. Chathuranga and L. Jayaratne, "Automatic Music Genre Classification of Audio Signals with Machine Learning Approaches," *GSTF Journal on Computing (JoC)*, vol. 3, no. 2, p. 14, Aug. 16, 2013.
- [74] E. Chaudary, S. Aziz, M. U. Khan, and P. Gretschnann, "Music genre classification using support vector machine and empirical mode decomposition," in *2021 Mohammad Ali Jinnah University International Conference on Computing (MAJICC)*, Jul. 2021, pp. 1–5.
- [75] M. Chaudhury, A. Karami, and M. A. Ghazanfar, "Large-scale music genre analysis and classification using machine learning with apache spark," *Electronics*, vol. 11, no. 16, p. 2567, 16 Jan. 2022.
- [76] O. Chavan, N. Kharade, A. Chaudhari, N. Bhalke, and P. Nimbalkar, "Machine learning and noise reduction techniques for music genre classification," *International Research Journal of Engineering and Technology*, vol. 6, no. 12, pp. 225–228, 2019.
- [77] X. Chen and P. J. Ramadge, "Music genre classification using multiscale scattering and sparse representations," in *2013 47th Annual Conference on Information Sciences and Systems (CISS)*, Mar. 2013, pp. 1–6.
- [78] S.-H. Chen, S.-Y. Ko, and S.-H. Chen, "Automatic music genre classification based on sparse representation and wavelet packet transform with discrete trigonometric transform," in *2016 Third International Conference on Computing Measurement Control and Sensor Network (CMCSN)*, May 2016, pp. 134–137.
- [79] S.-H. Chen, S.-Y. Ko, and S.-H. Chen, "Robust music genre classification based on sparse representation and wavelet packet transform with discrete trigonometric transform," *Journal of Network Intelligence*, vol. 1, no. 2, pp. 67–82, 2016.
- [80] Y.-T. Chen, C.-H. Chen, S. Wu, and C.-C. Lo, "A two-step approach for classifying music genre on the strength of AHP weighted musical features," *Mathematics*, vol. 7, no. 1, p. 19, 1 Jan. 2019.
- [81] K. Chen, B. Liang, X. Ma, and M. Gu, "Learning Audio Embeddings with User Listening Data for Content-Based Music Recommendation," in *2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Jun. 2021, pp. 3015–3019.
- [82] C. Chen and X. Steven, "Combined transfer and active learning for high accuracy music genre classification method," in *2021 IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE)*, Mar. 2021, pp. 53–56.
- [83] W. Chen and G. Wu, "A multimodal convolutional neural network model for the analysis of music genre on children's emotions influence intelligence," *Computational Intelligence and Neuroscience*, vol. 2022, e5611456, Aug. 29, 2022.
- [84] Y.-H. Cheng, P.-C. Chang, and C.-N. Kuo, "Convolutional neural networks approach for music genre classification," in *2020 International Symposium on Computer, Consumer and Control (IS3C)*, Nov. 2020, pp. 399–403.
- [85] Y.-H. Cheng, P.-C. Chang, D.-M. Nguyen, and C.-N. Kuo, "Automatic music genre classification based on CRNN," *Engineering Letters*, vol. 29, no. 1, 2020.

- [86] Y.-H. Cheng, P.-C. Chang, and C.-N. Kuo, "Music genre classification based on visualized spectrogram," in *2021 International Conference on Technologies and Applications of Artificial Intelligence (TAAI)*, Nov. 2021, pp. 217–221.
- [87] G. Chettiar and S. Kalaivani, "Music genre classification techniques," *International Journal of Engineering Research and Technology*, vol. 10, no. 11, pp. 158–61, 2021.
- [88] P. Chiliguano and G. Fazekas, "Hybrid music recommender using content-based and social information," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2016, pp. 2618–2622.
- [89] K. Choi, G. Fazekas, M. Sandler, and K. Cho, "Transfer learning for music classification and regression tasks," in *Proceedings of the 18th International Society for Music Information Retrieval Conference, ISMIR*, Sep. 13, 2017, pp. 141–149. arXiv: 1703.09179 [cs].
- [90] K. Choi, G. Fazekas, K. Cho, and M. Sandler, "The effects of noisy labels on deep convolutional neural networks for music tagging," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 2, pp. 139–149, Apr. 2018.
- [91] J. Choi, J. Lee, J. Park, and J. Nam, "Zero-shot Learning for Audio-based Music Classification and Tagging," in *Proceedings of the 20th International Society for Music Information Retrieval Conference, ISMIR*, 2019, pp. 67–74. arXiv: 1907.02670 [cs, eess].
- [92] C.-H. Chou and B.-J. Liao, "Music genre classification by analyzing the subband spectrogram," in *2014 International Conference on Information Science, Electronics and Electrical Engineering*, vol. 3, Apr. 2014, pp. 1677–1680.
- [93] C.-H. Chou and J.-H. Shi, "Time-frequency analysis for music genre classification by using wavelet package decompositions," in *2018 1st IEEE International Conference on Knowledge Innovation and Invention (ICKII)*, Jul. 2018, pp. 134–137.
- [94] Ö. Çoban, "Turkish music genre classification using audio and lyrics features," *Journal of Natural and Applied Sciences*, vol. 21, no. 2, pp. 322–331, 2 May 6, 2017.
- [95] J. L. Conceição, R. de Freitas, B. Gadelha, J. G. Kienen, S. Anders, and B. Cavalcante, "Applying supervised learning techniques to brazilian music genre classification," in *2020 XLVI Latin American Computing Conference (CLEI)*, Oct. 2020, pp. 102–107.
- [96] D. Conklin, "Fusion functions for multiple viewpoints," in *Proceedings of the 6th International Workshop on Machine Learning and Music*, 2013.
- [97] A. D. Coronel, "Building an Initial Fitness Function Based on an Identified Melodic Feature Set for Classical and Non-Classical Melody Classification," in *2013 International Conference on Information Science and Applications (ICISA)*, Jun. 2013, pp. 1–4.
- [98] D. C. Corrêa and F. A. Rodrigues, "A survey on symbolic data-based music genre classification," *Expert Systems with Applications*, vol. 60, pp. 190–210, Oct. 30, 2016.
- [99] Y. Costa, L. Oliveira, A. Koerich, and F. Gouyon, "Music genre recognition based on visual features with dynamic ensemble of classifiers selection," in *2013 20th International Conference on Systems, Signals and Image Processing (IWSSIP)*, Jul. 2013, pp. 55–58.
- [100] Y. Costa, L. Oliveira, A. Koerich, and F. Gouyon, "Music Genre Recognition Using Gabor Filters and LPQ Texture Descriptors," in *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, J. Ruiz-Shulcloper and G. Sanniti di Baja, Eds., ser. Lecture Notes in Computer Science, Berlin, Heidelberg: Springer, 2013, pp. 67–74.
- [101] Y. M. G. Costa, L. S. Oliveira, and C. N. Silla, "An evaluation of convolutional neural networks for music classification using spectrograms," *Applied Soft Computing*, vol. 52, pp. 28–38, Mar. 1, 2017.
- [102] D. A. Cruz, C. C. Cristancho, and J. E. Camargo, "Automatic Identification of Traditional Colombian Music Genres Based on Audio Content Analysis and Machine Learning Techniques," in *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, I. Nyström, Y. Hernández Heredia, and V. Milián Núñez, Eds., Cham: Springer International Publishing, 2019, pp. 646–655.
- [103] J. Dai, W. Liu, C. Ni, L. Dong, and H. Yang, "“Multilingual” deep neural network for music genre classification," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.
- [104] J. Dai, W. Liu, H. Zheng, W. Xue, and C. Ni, "Semi-supervised learning of bottleneck feature for music genre classification," in *Pattern Recognition*, T. Tan, X. Li, X. Chen, J. Zhou, J. Yang, and H. Cheng, Eds., ser. Communications in Computer and Information Science, Singapore: Springer, 2016, pp. 552–562.
- [105] J. Dai, S. Liang, W. Xue, C. Ni, and W. Liu, "Long short-term memory recurrent neural network based segment features for music genre classification," in *2016 10th International Symposium on Chinese Spoken Language Processing (ISCSLP)*, Oct. 2016, pp. 1–5.

- [106] J. Dai, W. Xue, and W. Liu, "Multilingual i-vector based statistical modeling for music genre classification," in *Interspeech*, 2017, pp. 459–463.
- [107] S. Das and A. K. Kolya, "A theoretic approach to music genre recognition from musical features using single-layer feedforward neural network," in *Emerging Technologies in Data Mining and Information Security*, A. Abraham, P. Dutta, J. K. Mandal, A. Bhattacharya, and S. Dutta, Eds., ser. Advances in Intelligent Systems and Computing, Singapore: Springer, 2019, pp. 145–155.
- [108] A. C. M. da Silva, P. R. V. do Carmo, R. M. Marcacini, and D. F. Silva, "Instance selection for music genre classification using heterogeneous networks," in *Anais Do XVIII Simpósio Brasileiro de Computação Musical*, SBC, 2021, pp. 8–16.
- [109] V. H. Da Silva Muniz, J. B. de Oliveira, and S. Filho, "Feature vector design for music genre classification," in *2021 IEEE Latin American Conference on Computational Intelligence (LA-CCI)*, Nov. 2021, pp. 1–6.
- [110] R. de Araújo Lima, R. C. C. de Sousa, H. Lopes, and S. D. J. Barbosa, "Brazilian lyrics-based music genre classification using a BLSTM network," in *Artificial Intelligence and Soft Computing*, L. Rutkowski, R. Scherer, M. Korytkowski, W. Pedrycz, R. Tadeusiewicz, and J. M. Zurada, Eds., ser. Lecture Notes in Computer Science, Cham: Springer International Publishing, 2020, pp. 525–534.
- [111] S. Deepak and B. Prasad, "Music classification based on genre using LSTM," in *2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)*, Jul. 2020, pp. 985–991.
- [112] "Structured auto-encoder with application to music genre recognition," M.S. thesis, Ecole Polytechnique Fédérale de Lausanne, 2015.
- [113] M. Defferrard, K. Benzi, P. Vandergheynst, and X. Bresson, "FMA: A dataset for music analysis," in *Proceedings of the 18th International Society for Music Information Retrieval Conference*, arXiv, Sep. 5, 2017, pp. 316–323. arXiv: 1612.01840 [cs].
- [114] M. Defferrard, S. P. Mohanty, S. F. Carroll, and M. Salathé, "Learning to Recognize Musical Genre from Audio: Challenge Overview," in *Companion Proceedings of the The Web Conference 2018*, ser. WWW '18, Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, Apr. 23, 2018, pp. 1921–1922.
- [115] P. J. P. de León, C. Pérez-Sancho, and J. M. Iñesta, "A shallow description framework for musical style recognition," in *Structural, Syntactic, and Statistical Pattern Recognition*, A. Fred, T. M. Caelli, R. P. W. Duin, A. C. Campilho, and D. de Ridder, Eds., ser. Lecture Notes in Computer Science, Berlin, Heidelberg: Springer, 2004, pp. 876–884.
- [116] F. A. de Leon and K. Martinez, "Music genre classification using polyphonic timbre models," in *2014 19th International Conference on Digital Signal Processing*, Aug. 2014, pp. 415–420.
- [117] A. A. de Lima, R. M. Nunes, R. P. Ribeiro, and C. N. Silla, "Nordic Music Genre Classification Using Song Lyrics," in *Natural Language Processing and Information Systems*, E. Métais, M. Roche, and M. Teisseire, Eds., ser. Lecture Notes in Computer Science, Cham: Springer International Publishing, 2014, pp. 89–100.
- [118] R. de Lima Aguiar, Y. M. e Gomes da Costa, and L. Nanni, "Music genre recognition using spectrograms with harmonic-percussive sound separation," in *2016 35th International Conference of the Chilean Computer Science Society (SCCC)*, Oct. 2016, pp. 1–7.
- [119] G. Deng and Y. C. Ko, "Active learning music genre classification based on support vector machine," *Advances in Multimedia*, vol. 2022, e4705272, Jul. 7, 2022.
- [120] R. De Prisco, D. Malandrino, G. Zaccagnino, R. Zaccagnino, and R. Zizza, "A bio-inspired approach to infer functional rules and aesthetic goals from music genre styles," in *Proceedings of the 2017 International Conference on Computer Science and Artificial Intelligence*, ser. CSAI 2017, New York, NY, USA: Association for Computing Machinery, Dec. 5, 2017, pp. 5–9.
- [121] E. Dervakos, N. Kotsani, and G. Stamou, "Genre Recognition from Symbolic Music with CNNs," in *Artificial Intelligence in Music, Sound, Art and Design*, J. Romero, T. Martins, and N. Rodríguez-Fernández, Eds., ser. Lecture Notes in Computer Science, Cham: Springer International Publishing, 2021, pp. 98–114.
- [122] J. de Sousa, E. Torres Pereira, and L. Ribeiro Veloso, "A robust music genre classification approach for global and regional music datasets evaluation," in *2016 IEEE International Conference on Digital Signal Processing (DSP)*, Oct. 2016, pp. 109–113.
- [123] P. Devaki, A. Sivanandan, R. S. Kumar, and M. Z. Peer, "Music genre classification and isolation," in *2021 International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA)*, Oct. 2021, pp. 1–6.

- [124] M. K. Devi, U. Surya, Unnamalai. K, and Tharani. R. K, "Treatment for insomnia using music genre prediction using convolutional recurrent neural network," in *2022 1st International Conference on Computational Science and Technology (ICCST)*, Nov. 2022, pp. 919–922.
- [125] A. Dhall, Y. V. Srinivasa Murthy, and S. G. Koolagudi, "Music genre classification with convolutional neural networks and comparison with F, Q, and mel spectrogram-based images," in *Advances in Speech and Music Technology*, A. Biswas, E. Wennekes, T.-P. Hong, and A. Wiczorkowska, Eds., ser. *Advances in Intelligent Systems and Computing*, Singapore: Springer, 2021, pp. 235–248.
- [126] J. Dias, V. Pillai, H. Deshmukh, and A. Shah, "Music genre classification & recommendation system using CNN," in *Proceedings of the 7th International Conference on Innovations and Research in Technology and Engineering (ICIRTE-2022)*, Rochester, NY, Apr. 8, 2022.
- [127] D. Diefenbach, P.-R. Lherisson, F. Muhlenbach, and P. Maret, "Computing the semantic relatedness of music genres using semantic web data," presented at the *Semantics 2016*, Sep. 12, 2016.
- [128] S. Dieleman and B. Schrauwen, "Multiscale approaches to music audio feature learning," in *Proceedings of the 14th International Society for Music Information Retrieval Conference*, De Souza Britto Jr., Alceu, Gouyon, Fabien, and Dixon, Simon, Eds., Pontificia Universidade Católica do Paraná, 2013, pp. 116–121.
- [129] I.-J. Ding, C.-T. Yen, C.-W. Chang, and H.-Z. Lin, "Optical music recognition of the singer using formant frequency estimation of vocal fold vibration and lip motion with interpolated GMM classifiers," *Journal of Vibroengineering*, vol. 16, no. 5, pp. 2572–2581, 5 Aug. 15, 2014.
- [130] S. Dokania and V. Singh. "Graph representation learning for audio & music genre classification." arXiv: 1910.11117 [cs, stat]. (Oct. 23, 2019), [Online]. Available: <http://arxiv.org/abs/1910.11117> (visited on 03/03/2023), preprint.
- [131] A. Dorochowicz, P. Hoffmann, A. Majdańczuk, and B. Kostek, "Classification of Music Genres by Means of Listening Tests and Decision Algorithms," in *Intelligent Methods and Big Data in Industrial Applications*, ser. *Studies in Big Data*, R. Bembenik, Ł. Skonieczny, G. Protaziuk, M. Kryszkiewicz, and H. Rybinski, Eds., Cham: Springer International Publishing, 2019, pp. 291–305.
- [132] S. M. Doudpota, S. Guha, and J. Baber, "Mining movies for song sequences with video based music genre identification system," *Information Processing & Management*, vol. 49, no. 2, pp. 529–544, Mar. 1, 2013.
- [133] W. Du, H. Lin, J. Sun, B. Yu, and H. Yang, "A new hierarchical method for music genre classification," in *2016 9th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, Oct. 2016, pp. 1033–1037.
- [134] S. Duggirala and T.-S. Moh, "A novel approach to music genre classification using natural language processing and spark," in *2020 14th International Conference on Ubiquitous Information Management and Communication (IMCOM)*, Jan. 2020, pp. 1–8.
- [135] A. Elbir, H. O. İlhan, G. Serbes, and N. Aydın, "Short time fourier transform based music genre classification," in *2018 Electric Electronics, Computer Science, Biomedical Engineerings' Meeting (EBBT)*, Apr. 2018, pp. 1–4.
- [136] A. Elbir, H. Bilal Çam, M. Emre Iyican, B. Öztürk, and N. Aydın, "Music genre classification and recommendation by using machine learning techniques," in *2018 Innovations in Intelligent Systems and Applications Conference (ASYU)*, Oct. 2018, pp. 1–5.
- [137] A. Elbir and N. Aydın, "Music genre classification and music recommendation by using deep learning," *Electronics Letters*, vol. 56, no. 12, pp. 627–629, 2020.
- [138] A. Eppler, A. Männchen, J. Abeßer, C. Weiß, and K. Frieler, "Automatic style classification of jazz recordings with respect to rhythm, tempo, and tonality," in *Proceedings of the 9th Conference on Interdisciplinary Musicology*, Dec. 4, 2014.
- [139] E. V. Epure, A. Khelif, and R. Hennequin, "Leveraging knowledge bases and parallel annotations for music genre translation," in *Proceedings of the 20th International Society for Music Information Retrieval Conference*, arXiv, Jul. 27, 2019, pp. 839–846. arXiv: 1907.08698 [cs, eess, stat].
- [140] T. M. Esparza, J. P. Bello, and E. J. Humphrey, "From genre classification to rhythm similarity: Computational and musicological insights," *Journal of New Music Research*, vol. 44, no. 1, pp. 39–57, Jan. 2, 2015.
- [141] S. Evstifeev and I. Shanin, "Music genre classification based on signal processing," in *Data Analytics and Management in Data-Intensive Domains*, 2018, pp. 157–161.

- [142] P. B. Falola and S. O. Akinola, "Music genre classification using 1D convolution neural network," *International Journal of Human Computing Studies*, vol. 3, no. 6, pp. 3–21, 2021.
- [143] S. Fan and M. Fu, "Music genre recommendation based on MLP & random forest," in *2022 IEEE 5th International Conference on Information Systems and Computer Aided Education (ICISCAE)*, Sep. 2022, pp. 331–334.
- [144] X. Favory, K. Drossos, T. Virtanen, and X. Serra, "COALA: Co-Aligned Autoencoders for Learning Semantically Enriched Audio Representations," presented at the ICML 2020 Workshop on Self-Supervision in Audio and Speech, arXiv, Jul. 8, 2020. arXiv: 2006.08386 [cs, eess, stat].
- [145] X. Favory, K. Drossos, T. Virtanen, and X. Serra, "Learning Contextual Tag Embeddings for Cross-Modal Alignment of Audio and Tags," in *2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Jun. 2021, pp. 596–600.
- [146] M. Fell and C. Sporleder, "Lyrics-based Analysis and Classification of Music," in *Proceedings of COLING 2014*, Dublin, Ireland, 2014, pp. 620–631.
- [147] L. Feng, S. Liu, and J. Yao. "Music genre classification with paralleling recurrent convolutional neural network." arXiv: 1712.08370 [cs, eess]. (Dec. 22, 2017), [Online]. Available: <http://arxiv.org/abs/1712.08370> (visited on 03/03/2023), preprint.
- [148] A. Ferraro and K. Lemström, "On large-scale genre classification in symbolically encoded music by automatic identification of repeating patterns," in *Proceedings of the 5th International Conference on Digital Libraries for Musicology*, ser. DLfM '18, New York, NY, USA: Association for Computing Machinery, Sep. 28, 2018, pp. 34–37.
- [149] A. Flexer, E. Pampalk, and G. Widmer, "Novelty Detection Based on Spectral Similarity of Songs.," in *Proceedings of the 6th International Conference on Music Information Retrieval*, London: ISMIR, 2005, pp. 260–263.
- [150] A. Flexer, "Hubness-aware outlier detection for music genre recognition," in *Proceedings of the 19th International Conference on Digital Audio Effects (DAFx-16)*, Brno, Czech Republic, 2016.
- [151] J. H. Foleis and T. F. Tavares, "Texture selection for automatic music genre classification," *Applied Soft Computing*, vol. 89, p. 106 127, Apr. 1, 2020.
- [152] J. H. Foleiss and T. F. Tavares. "Random projections of mel-spectrograms as low-level features for automatic music genre classification." arXiv: 1911.04660 [cs, eess]. (Nov. 11, 2019), [Online]. Available: <http://arxiv.org/abs/1911.04660> (visited on 03/09/2023), preprint.
- [153] S. O. Folorunso, S. A. Afolabi, and A. B. Owodeyi, "Dissecting the genre of nigerian music with machine learning models," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, pp. 6266–6279, 8, Part B Sep. 1, 2022.
- [154] H. Foroughmand Aarabi and G. Peeters, "Extending Deep Rhythm for Tempo and Genre Estimation Using Complex Convolutions, Multitask Learning and Multi-input Network," *Journal of Creative Music Systems*, vol. 1, no. 1, Aug. 30, 2022.
- [155] A. Foroughmand Arabi, "Enhanced polyphonic music genre classification using high level features," thesis, Monash University, 2009.
- [156] E. Fotiadou, N. Bassiou, and C. Kotropoulos, "Greek folk music classification using auditory cortical representations," in *2016 24th European Signal Processing Conference (EUSIPCO)*, Aug. 2016, pp. 1133–1137.
- [157] R. Foucard, S. Essid, G. Richard, and M. Lagrange, "Exploring new features for music classification," in *2013 14th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS)*, Jul. 2013, pp. 1–4.
- [158] P. Fulzele, R. Singh, N. Kaushik, and K. Pandey, "A hybrid model for music genre classification using LSTM and SVM," in *2018 Eleventh International Conference on Contemporary Computing (IC3)*, Aug. 2018, pp. 1–3.
- [159] M. N. Furqon, K. Khadijah, S. Suhartono, and R. Kusumaningrum, "Indonesian music genre classification on indonesian regional songs using deep recurrent neural network method," in *2019 3rd International Conference on Informatics and Computational Sciences (ICICoS)*, Oct. 2019, pp. 1–5.
- [160] B. Gao, "Contributions to music semantic analysis and its acceleration techniques," Ph.D. dissertation, Ecole Centrale de Lyon, Dec. 15, 2014.
- [161] S. Geng, G. Ren, and M. Ogihara, "Transforming musical signals through a genre classifying convolutional neural network," in *Proceedings of the First International Workshop on Deep Learning and Music, Joint with IJCNN*, D. Herremans and C.-H. Chuan, Eds., Anchorage, US: arXiv, Jun. 28, 2017. arXiv: 1706.09553 [cs].
- [162] J. George and L. Shamir, "Unsupervised analysis of similarities between musicians and musical genres using spectrograms.," *Artificial Intelligence Research*, vol. 4, no. 2, pp. 61–71, 2015.

- [163] P. Ghaemmaghami and N. Sebe, "Brain and music: Music genre classification using brain signals," in *2016 24th European Signal Processing Conference (EUSIPCO)*, Aug. 2016, pp. 708–712.
- [164] A. Ghildiyal, K. Singh, and S. Sharma, "Music genre classification using machine learning," in *2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, Nov. 2020, pp. 1368–1372.
- [165] A. Ghildiyal and S. Sharma, "Music genre classification using data filtering algorithm: An artificial intelligence approach," in *2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA)*, Sep. 2021, pp. 922–926.
- [166] A. Ghosal, R. Chakraborty, B. C. Dhara, and S. K. Saha, "Genre-Based Classification of Song Using Perceptual Features," in *Intelligent Computing, Networking, and Informatics*, D. P. Mohapatra and S. Patnaik, Eds., ser. *Advances in Intelligent Systems and Computing*, New Delhi: Springer India, 2014, pp. 267–276.
- [167] A. Ghosal, R. Chakraborty, B. C. Dhara, and S. K. Saha, "Perceptual feature-based song genre classification using RANSAC," *International Journal of Computational Intelligence Studies*, vol. 4, no. 1, pp. 31–49, Jan. 2015.
- [168] D. Ghosal and M. Kolekar, "Music genre recognition using deep neural networks and transfer learning," presented at the Interspeech, Sep. 2, 2018, pp. 2087–2091.
- [169] D. Ghosal and M. F. Kolekar, "Musical genre and style recognition using deep neural networks and transfer learning," in *Proceedings, APSIPA Annual Summit and Conference*, vol. 2018, 2018, pp. 12–15.
- [170] S. S. Ghosal and I. Sarkar, "Novel approach to music genre classification using clustering augmented learning method (CALM).," in *AAAI Spring Symposium: Combining Machine Learning with Knowledge Engineering*, 2020.
- [171] P. Ginsel, I. Vatulkin, and G. Rudolph, "Analysis of structural complexity features for music genre recognition," in *2020 IEEE Congress on Evolutionary Computation (CEC)*, Jul. 2020, pp. 1–8.
- [172] A. Girase, A. Advirkar, C. Patil, D. Khadpe, and A. Pokhare, "Lyrics Based Song Genre Classification," *Journal of Computing Technologies*, vol. 3, no. 2, pp. 16–19, 2014.
- [173] I. N. Y. T. Giria, L. A. A. R. Putria, G. A. V. M. Giria, I. G. N. A. C. Putraa, I. M. Widiartha, and I. W. Suprianaa, "Music Genre Classification Using Modified K-Nearest Neighbor (MK-NN)," *JELIKU (Jurnal Elektronik Ilmu Komputer Udayana)*, vol. 10, no. 3, p. 5373, 2022.
- [174] A. Goel, Mohd. Sheezan, S. Masood, and A. Saleem, "Genre classification of songs using neural network," in *2014 International Conference on Computer and Communication Technology (IC-CCT)*, Sep. 2014, pp. 285–289.
- [175] I. Goienetxea, J. M. Martínez-Otzeta, B. Sierra, and I. Mendiadua, "Towards the use of similarity distances to music genre classification: A comparative study," *PLoS ONE*, vol. 13, no. 2, e0191417, Feb. 14, 2018.
- [176] C. K. Gomathy and V. Geetha, "Music classification management system," *International Journal of Early Childhood Special Education (INT-JECSE)*, vol. 10, no. 5, 2022.
- [177] W. Gong and Q. Yu, "A deep music recommendation method based on human motion analysis," *IEEE Access*, vol. 9, pp. 26 290–26 300, 2021.
- [178] R. Gupta, J. Yadav, and C. Kapoor, "Music information retrieval and intelligent genre classification," in *Proceedings of International Conference on Intelligent Computing, Information and Control Systems*, A. P. Pandian, R. Palanisamy, and K. Ntalianis, Eds., ser. *Advances in Intelligent Systems and Computing*, Singapore: Springer, 2021, pp. 207–224.
- [179] R. Gusain, S. Sonker, S. K. Rai, A. Arora, and S. Nagarajan, "Comparison of neural networks and xgboost algorithm for music genre classification," in *2022 2nd International Conference on Intelligent Technologies (CONIT)*, Jun. 2022, pp. 1–6.
- [180] P. Hamel, M. E. P. Davies, K. Yoshii, and M. Goto, "Transfer learning in MIR: Sharing learned latent representations for music audio classification and similarity," in *Proceedings of the 14th International Society for Music Information Retrieval Conference*, A. de Souza Britto Jr., F. Gouyon, and S. Dixon, Eds., 2013, pp. 9–14.
- [181] L. K. Hansen, T. Lehn-Schiøler, K. B. Petersen, J. Arenas-García, J. Larsen, and S. H. Jensen, "Learning and clean-up in a large scale music database," in *2007 15th European Signal Processing Conference*, Sep. 2007, pp. 946–950.
- [182] I. U. Haq, F. Khan, S. Sharif, and A. Shaukat, "Automatic music genres classification as a pattern recognition problem," in *Sixth International Conference on Machine Vision (ICMV 2013)*, vol. 9067, SPIE, Dec. 24, 2013, pp. 438–443.
- [183] M. Haro Berois, "Statistical distribution of common audio features : Encounters in a heavy-tailed universe," Ph.D. dissertation, Universitat Pompeu Fabra, Nov. 22, 2013.
- [184] M. Hartmann, P. Saari, P. Toiviainen, and O. Lartillot, "Comparing Timbre-based Features for Musical Genre Classification," in *Proceedings of the 10th Sound and Music Computing Conference*, Stockholm, Sweden, 2013.

- [185] R. Hasan, S. Hossain, F. I. Alam, and M. Barua, "Bangla music genre classification using fast and scalable integrated ensemble boosting framework," in *2021 3rd International Conference on Sustainable Technologies for Industry 4.0 (STI)*, Dec. 2021, pp. 1–6.
- [186] K. M. Hasib, A. Tanzim, J. Shin, K. O. Faruk, J. A. Mahmud, and M. F. Mridha, "BMNet-5: A novel approach of neural network to classify the genre of bengali music based on audio features," *IEEE Access*, vol. 10, pp. 108 545–108 563, 2022.
- [187] Q. He, "A music genre classification method based on deep learning," *Mathematical Problems in Engineering*, vol. 2022, e9668018, Mar. 29, 2022.
- [188] A. Heakl, A. Abdelgawad, and V. Parque, "A study on broadcast networks for music genre classification," in *2022 International Joint Conference on Neural Networks (IJCNN)*, Jul. 2022, pp. 1–8.
- [189] F. Heerde, I. Vatolkin, and G. Rudolph, "Comparing fuzzy rule based approaches for music genre classification," in *Artificial Intelligence in Music, Sound, Art and Design*, J. Romero, A. Ekárt, T. Martins, and J. Correia, Eds., ser. Lecture Notes in Computer Science, Cham: Springer International Publishing, 2020, pp. 35–48.
- [190] R. Hennequin, J. Royo-Letelier, and M. Mousallam, "Audio Based Disambiguation of Music Genre Tags," in *Proceedings of the 19th International Society for Music Information Retrieval Conference*, Paris, France: ISMIR, Sep. 23, 2018, pp. 645–652.
- [191] M. Henry, W. Chandra, and A. Zahra, "Implementation of apriori algorithm for music genre recommendation," *Jurnal Online Informatika*, vol. 7, no. 1, pp. 110–115, 2022.
- [192] J. Heo, H.-s. Shin, J.-h. Kim, C.-y. Lim, and H.-J. Yu, "Convolution channel separation and frequency sub-bands aggregation for music genre classification." arXiv: 2211 . 01599 [cs, eess]. (Nov. 3, 2022), [Online]. Available: <http://arxiv.org/abs/2211.01599> (visited on 03/23/2023), preprint.
- [193] L. Hoang, "Literature review about music genre classification," in *Woodstock'18: ACM Symposium on Neural Gaze Detection*, Woodstock, NY: ACM, 2018.
- [194] J. Hockman, J. Bello, M. Davies, and M. Plumbley, "Automated rhythmic transformation of musical audio," in *Proceedings - 11th International Conference on Digital Audio Effects, DAFx 2008*, 2008, pp. 177–180.
- [195] P. Hoffmann and B. Kostek, "Music data processing and mining in large databases for active media," in *Active Media Technology*, D. Ślęzak, G. Schaefer, S. T. Vuong, and Y.-S. Kim, Eds., ser. Lecture Notes in Computer Science, Cham: Springer International Publishing, 2014, pp. 85–95.
- [196] P. Hoffmann and B. Kostek, "Music genre recognition in the rough set-based environment," in *Pattern Recognition and Machine Intelligence*, M. Kryszkiewicz, S. Bandyopadhyay, H. Rybinski, and S. K. Pal, Eds., ser. Lecture Notes in Computer Science, Cham: Springer International Publishing, 2015, pp. 377–386.
- [197] P. Hoffmann and B. Kostek, "Bass enhancement settings in portable devices based on music genre recognition," *Journal of the Audio Engineering Society*, vol. 63, no. 12, pp. 980–989, Jan. 6, 2016.
- [198] W. Hongdan, S. SalmiJamali, C. Zhengping, S. Qiaojuan, and R. Le, "An intelligent music genre analysis using feature extraction and classification using deep learning techniques," *Computers & Electrical Engineering*, vol. 100, p. 107978, May 1, 2022.
- [199] R. Hossain and A. Al Marouf, "BanglaMusicStylo: A stylometric dataset of bangla music lyrics," in *2018 International Conference on Bangla Speech and Language Processing (ICBSLP)*, Sep. 2018, pp. 1–5.
- [200] K.-C. Hsu, C.-S. Lin, and T.-S. Chi, "Sparse coding based music genre classification using spectro-temporal modulations.," in *Proceedings of the 17th International Society for Music Information Retrieval Conference*, ISMIR, 2016, pp. 744–750.
- [201] W.-H. Hsu, B.-Y. Chen, and Y.-H. Yang, "Deep Learning Based EDM Subgenre Classification using Mel-Spectrogram and Tempogram Features." arXiv: 2110 . 08862 [cs, eess]. (Oct. 17, 2021), [Online]. Available: <http://arxiv.org/abs/2110.08862> (visited on 03/24/2023), preprint.
- [202] X. Hu, J. S. Downie, K. West, and A. F. Ehmann, "Mining music reviews: Promising preliminary results.," in *Proceedings of the 6th International Conference on Music Information Retrieval*, London, United Kingdom: ISMIR, Sep. 11, 2005, pp. 536–539.
- [203] Y.-F. Huang, S.-M. Lin, H.-Y. Wu, and Y.-S. Li, "Music genre classification based on local feature selection using a self-adaptive harmony search algorithm," *Data & Knowledge Engineering*, vol. 92, pp. 60–76, Jul. 1, 2014.

- [204] I. Ikhsan, L. Novamizanti, and I. N. A. Ramatryana, "Automatic musical genre classification of audio using Hidden Markov Model," in *2014 2nd International Conference on Information and Communication Technology (ICoICT)*, May 2014, pp. 397–402.
- [205] S. Iloga, O. Romain, L. Bendaouia, and M. Tchuenta, "Musical genres classification using Markov models," in *2014 International Conference on Audio, Language and Image Processing*, Shanghai, China: IEEE, Jul. 2014, pp. 701–705.
- [206] S. Iloga, O. Romain, and M. Tchuenté, "A sequential pattern mining approach to design taxonomies for hierarchical music genre recognition," *Pattern Analysis and Applications*, vol. 21, no. 2, pp. 363–380, May 1, 2018.
- [207] D. Imran, H. Wadiwala, M. A. Tahir, and M. Rafi, "Semantic feature extraction using feed-forward neural network for music genre classification," *Asian Journal of Engineering, Sciences & Technology*, vol. 7, no. 2, 2017.
- [208] M. S. Islam *et al.*, "Machine learning-based music genre classification with pre-processed feature analysis," *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika*, vol. 7, no. 3, pp. 491–502, 2021.
- [209] K. Itoyama, M. Goto, K. Komatani, T. Ogata, and H. G. Okuno, "Query-by-example music retrieval approach based on musical genre shift by changing instrument volume," in *Proceedings of the 12th Int. Conference on Digital Audio Effects (DAFx-09)*, Como, Italy, 2009.
- [210] R. Jain, R. Sharma, P. Nagrath, and R. Jain, "Music genre classification ChatBot," in *Proceedings of Second International Conference on Computing, Communications, and Cyber-Security*, P. K. Singh, S. T. Wierzchoń, S. Tanwar, M. Ganzha, and J. J. P. C. Rodrigues, Eds., ser. Lecture Notes in Networks and Systems, Singapore: Springer, 2021, pp. 393–408.
- [211] M. Jakubec and M. Chmulik, "Automatic music genre recognition for in-car infotainment," *Transportation Research Procedia*, TRANSCOM 2019 13th International Scientific Conference on Sustainable, Modern and Safe Transport, vol. 40, pp. 1364–1371, Jan. 1, 2019.
- [212] D. Jang and S.-J. Jang, "Very short feature vector for music genre classification based on distance metric learning," in *2014 International Conference on Audio, Language and Image Processing*, Jul. 2014, pp. 726–729.
- [213] P.-K. Jao, L. Su, and Y.-H. Yang, "Analyzing the dictionary properties and sparsity constraints for a dictionary-based music genre classification system," in *2013 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*, Oct. 2013, pp. 1–8.
- [214] P.-K. Jao, C.-C. M. Yeh, and Y.-H. Yang, "Modified lasso screening for audio word-based music classification using large-scale dictionary," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, pp. 5207–5211.
- [215] P.-K. Jao and Y.-H. Yang, "Music annotation and retrieval using unlabeled exemplars: Correlation and sparse codes," *IEEE Signal Processing Letters*, vol. 22, no. 10, pp. 1771–1775, Oct. 2015.
- [216] G. JawaherlalNehru, S. Jothilakshmi, S. Jothishri, S. Bavankumar, B. Rajalingam, and R. Santhoshkumar, "Music genre classification using deep learning techniques," *Turkish Online Journal of Qualitative Inquiry*, vol. 12, no. 8, pp. 7293–7305, 8 Nov. 10, 2021.
- [217] B. S. Jensen, R. Troelsgaard, J. Larsen, and L. K. Hansen, "Towards a universal representation for audio information retrieval and analysis," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2013, pp. 3168–3172.
- [218] I.-Y. Jeong and K. Lee, "Learning temporal features using a deep neural network and its application to music genre classification," in *Proceedings of the 17th International Society for Music Information Retrieval Conference*, New York City, United States, 2016, pp. 434–440.
- [219] Z. Jiang and H. N. Huynh, "Unveiling music genre structure through common-interest communities," *Social Network Analysis and Mining*, vol. 12, no. 1, p. 35, Feb. 14, 2022.
- [220] C. Johnson-Roberson and E. Sudderth, "Content-based genre classification and sample recognition using topic models," M.S. thesis, Brown University, 2015.
- [221] A. Kamala and H. Hassani, "Kurdish music genre recognition using a CNN and DNN," *Engineering Proceedings*, vol. 31, no. 1, p. 64, 1 2022.
- [222] K. Kamtue, K. Euchukanonchai, D. Wanvarie, and N. Pratanwanich, "Lukthung Classification Using Neural Networks on Lyrics and Audios," in *2019 23rd International Computer Science and Engineering Conference (ICSEC)*, Oct. 2019, pp. 269–274.
- [223] E. Kanalici and G. Bilgin, "Music genre classification via sequential wavelet scattering feature learning," in *Knowledge Science, Engineering and Management*, C. Douligeris, D. Karagiannis, and D. Apostolou, Eds., ser. Lecture Notes in Computer Science, Cham: Springer International Publishing, 2019, pp. 365–372.

- [224] K. Kang and F. Lin. “Computing Class Hierarchies from Classifiers.” arXiv: 2112.01187 [cs]. (Dec. 2, 2021), [Online]. Available: <http://arxiv.org/abs/2112.01187> (visited on 03/24/2023), preprint.
- [225] D. Kania, P. Kania, and T. Łukaszewicz, “Trajectory of fifths in music data mining,” *IEEE Access*, vol. 9, pp. 8751–8761, 2021.
- [226] G. Karamanolakis, E. Iosif, A. Zlatintsi, A. Pikrakis, and A. Potamianos, “Audio-based distributional semantic models for music auto-tagging and similarity measurement,” in *Multi-Learn 2017 Workshop at the 25th European Signal Processing Conference*, Kos Island, Greece, 2017.
- [227] N. Karunakaran and A. Arya, “A scalable hybrid classifier for music genre classification using machine learning concepts and spark,” in *2018 International Conference on Intelligent Autonomous Systems (ICoIAS)*, Mar. 2018, pp. 128–135.
- [228] E. Karystinaios, C. Guichaoua, M. Andreatta, L. Bigo, and I. Bloch, “Music genre descriptor for classification based on tonnetz trajectories,” in *Actes Musical et Environnements Informatiques : Actes Des Journées d’Informatique Musicale 2020 (JIM 2020)*, 2021.
- [229] C. Kaur and R. Kumar, “Study and analysis of feature based automatic music genre classification using gaussian mixture model,” in *2017 International Conference on Inventive Computing and Informatics (ICICI)*, Nov. 2017, pp. 465–468.
- [230] A. J. E. Kell, D. L. K. Yamins, E. N. Shook, S. V. Norman-Haignere, and J. H. McDermott, “A Task-Optimized Neural Network Replicates Human Auditory Behavior, Predicts Brain Responses, and Reveals a Cortical Processing Hierarchy,” *Neuron*, vol. 98, no. 3, pp. 630–644.e16, May 2, 2018.
- [231] C. Kereliuk, B. L. Sturm, and J. Larsen, “Deep learning, audio adversaries, and music content analysis,” in *2015 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2015.
- [232] C. Kereliuk, B. L. Sturm, and J. Larsen, “Deep learning and music adversaries,” *IEEE Transactions on Multimedia*, vol. 17, no. 11, pp. 2059–2071, Sep. 2015.
- [233] Y. Khasgiwala and J. Taylor, “Vision transformer for music genre classification using mel-frequency cepstrum coefficient,” in *2021 IEEE 4th International Conference on Computing, Power and Communication Technologies (GUCON)*, Sep. 2021, pp. 1–5.
- [234] Y. Kikuchi and N. Aoki, “A study on automatic music genre classification based on the summarization of music data,” in *2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIIC)*, Feb. 2020, pp. 705–708.
- [235] J. Kim, M. Won, X. Serra, and C. C. S. Liem, “Transfer Learning of Artist Group Factors to Musical Genre Classification,” in *Companion of the The Web Conference 2018 on The Web Conference 2018 - WWW ’18*, 2018, pp. 1929–1934. arXiv: 1805.02043 [cs, eess, stat].
- [236] J. Kim, J. Urbano, C. C. S. Liem, and A. Hanjalic, “One deep music representation to rule them all? A comparative analysis of different representation learning strategies,” *Neural Computing and Applications*, vol. 32, no. 4, pp. 1067–1093, Feb. 1, 2020.
- [237] J. Kim and C. C. S. Liem, “The power of deep without going deep? A study of HDPGMM music representation learning,” in *Proceedings of the 23rd International Society for Music Information Retrieval Conference*, Bengaluru, India: ISMIR, Dec. 4, 2022, pp. 116–124.
- [238] M. A. Kızrak, K. S. Bayram, and B. Bolat, “Classification of Classic Turkish Music Makams,” in *2014 IEEE International Symposium on Innovations in Intelligent Systems and Applications (INISTA) Proceedings*, Jun. 2014, pp. 394–397.
- [239] M. Kleć and D. Koržinek, “Unsupervised feature pre-training of the scattering wavelet transform for musical genre recognition,” in *International Workshop on Innovations in Information and Communication Science and Technology*, vol. 18, Jan. 1, 2014, pp. 133–139.
- [240] M. Kleć and D. Korzinek, “Pre-trained deep neural network using sparse autoencoders and scattering wavelet transform for musical genre recognition,” *Computer Science*, vol. 16, no. 2, pp. 133–144, 2015.
- [241] M. Kleć, “Multi-instrumental deep learning for automatic genre recognition,” in *Recent Developments in Intelligent Information and Database Systems*, Springer, 2016, pp. 53–61.
- [242] M. Kobayakawa, M. Hoshi, and K. Yuzawa, “Music genre classification of MPEG AAC audio data,” in *2014 IEEE International Symposium on Multimedia*, Dec. 2014, pp. 347–352.
- [243] T. Kobayashi, A. Kubota, and Y. Suzuki, “Audio feature extraction based on sub-band signal correlations for music genre classification,” in *2018 IEEE International Symposium on Multimedia (ISM)*, Dec. 2018, pp. 180–181.
- [244] A. L. Koerich, “Improving the Reliability of Music Genre Classification using Rejection and Verification,” in *Proceedings of the 14th International Society for Music Information Retrieval Conference*, Curitiba, Brazil: ISMIR, Nov. 4, 2013, pp. 511–516.

- [245] S. Koparde, V. R. Bhadgaonkar, K. N. Patil, G. N. Basutkar, and D. D. Gayke, "A survey on music genre classification using machine learning," *International Research Journal of Engineering and Technology*, vol. 08, no. 03, pp. 640–644, 2021.
- [246] G. Korvel and B. Kostek, "Discovering rule-based learning systems for the purpose of music analysis," in *Proceedings of Meetings on Acoustics*, vol. 39, Acoustical Society of America, Dec. 2, 2019, p. 035 004.
- [247] B. Kostek, "Music Information Retrieval in Music Repositories," in *Rough Sets and Intelligent Systems - Professor Zdzislaw Pawlak in Memoriam: Volume 1*, ser. Intelligent Systems Reference Library, A. Skowron and Z. Suraj, Eds., Berlin, Heidelberg: Springer, 2013, pp. 463–489.
- [248] B. Kostek, P. Hoffmann, A. Kaczmarek, and P. Spaleniak, "Creating a Reliable Music Discovery and Recommendation System," in *Intelligent Tools for Building a Scientific Information Platform: From Research to Implementation*, ser. Studies in Computational Intelligence, R. Bembeniak, Ł. Skonieczny, H. Rybiński, M. Kryszkiewicz, and M. Niezgódka, Eds., Cham: Springer International Publishing, 2014, pp. 107–130.
- [249] D. Kostrzewa, R. Brzeski, and M. Kubanski, "The classification of music by the genre using the KNN classifier," in *Beyond Databases, Architectures and Structures. Facing the Challenges of Data Proliferation and Growing Variety*, S. Kozielski, D. Mrozek, P. Kasprowski, B. Małysiak-Mrozek, and D. Kostrzewa, Eds., ser. Communications in Computer and Information Science, Cham: Springer International Publishing, 2018, pp. 233–242.
- [250] D. Kostrzewa, P. Kaminski, and R. Brzeski, "Music genre classification: Looking for the perfect network," in *Computational Science –ICCS 2021*, M. Paszynski, D. Kranzlmüller, V. V. Krzhizhanovskaya, J. J. Dongarra, and P. M. A. Sloot, Eds., ser. Lecture Notes in Computer Science, Cham: Springer International Publishing, 2021, pp. 55–67.
- [251] N. Kothari and P. Kumar, "Literature survey for music genre classification using neural network," *International Research Journal of Engineering and Technology*, vol. 9, pp. 691–695, 2022.
- [252] A. Kotsifakos, E. E. Kotsifakos, P. Papapetrou, and V. Athitsos, "Genre classification of symbolic music with SMBGT," in *Proceedings of the 6th International Conference on Pervasive Technologies Related to Assistive Environments*, ser. PE-TRA '13, New York, NY, USA: Association for Computing Machinery, May 29, 2013, pp. 1–7.
- [253] G. Kour and N. Mehan, "Music genre classification using MFCC, SVM and BPNN," *International Journal of Computer Applications*, vol. 112, no. 6, pp. 12–14, Feb. 18, 2015.
- [254] A. Koutras, "Song emotion recognition using music genre information," in *Speech and Computer*, A. Karpov, R. Potapova, and I. Mporas, Eds., ser. Lecture Notes in Computer Science, Cham: Springer International Publishing, 2017, pp. 669–679.
- [255] D. Kowald, E. Lex, and M. Schedl, "Utilizing human memory processes to model genre preferences for personalized music recommendations," in *IUI '20: Proceedings of the 25th International Conference on Intelligent User Interfaces Companion*, ACM, Mar. 24, 2020, pp. 19–20. arXiv: 2003.10699 [cs].
- [256] P. P. Kuksa. "Efficient multivariate sequence classification." arXiv: 1409.8211 [cs]. (Sep. 30, 2014), [Online]. Available: <http://arxiv.org/abs/1409.8211> (visited on 02/19/2024), preprint.
- [257] D. P. Kumar, B. J. Sowmya, Chetan, and K. G. Srinivasa, "A comparative study of classifiers for music genre classification based on feature extractors," in *2016 IEEE Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER)*, Aug. 2016, pp. 190–194.
- [258] A. Kumar, B. K. S. Siva, G. S. Reddy, and M. R. Rashmi, "Music genre classification," *International Research Journal of Engineering and Technology*, vol. 4, no. 4, pp. 3412–3414, 2017.
- [259] B. Kumaraswamy and P. G. Poonacha, "Deep convolutional neural network for musical genre classification via new self adaptive sea lion optimization," *Applied Soft Computing*, vol. 108, p. 107446, Sep. 1, 2021.
- [260] B. Kumaraswamy, "Optimized deep learning for genre classification via improved moth flame algorithm," *Multimedia Tools and Applications*, vol. 81, no. 12, pp. 17071–17093, May 1, 2022.
- [261] N. Kumari, T. Shukla, K. S. Swati, and K. Balachandra, "Music genre classification for indian music genres," *International Journal for Research in Applied Science and Engineering Technology*, vol. 9, no. 8, pp. 1756–1762, 2021.
- [262] B. Lansdown, "Machine Learning for Music Genre Classification," M.S. thesis, University of Birmingham, Sep. 1, 2019.
- [263] D. S. Lau and R. Ajoodha, "Music genre classification: A comparative study between deep learning and traditional machine learning approaches," in *Proceedings of Sixth International Congress on Information and Communication Technology*, X.-S. Yang, S. Sherratt, N. Dey, and A. Joshi, Eds.,

- ser. Lecture Notes in Networks and Systems, Singapore: Springer, 2022, pp. 239–247.
- [264] K. Leartpantulak and Y. Kitjaidure, “Music genre classification of audio signals using particle swarm optimization and stacking ensemble,” in *2019 7th International Electrical Engineering Congress (iEECON)*, Mar. 2019, pp. 1–4.
- [265] C.-H. Lee, H.-S. Lin, and L.-H. Chen, “Music classification using the bag of words model of modulation spectral features,” in *2015 15th International Symposium on Communications and Information Technologies (ISCIT)*, Oct. 2015, pp. 121–124.
- [266] J. Lee, S. Shin, D. Jang, S.-J. Jang, and K. Yoon, “Music recommendation system based on usage history and automatic genre classification,” in *2015 IEEE International Conference on Consumer Electronics (ICCE)*, Jan. 2015, pp. 134–135.
- [267] J. Lee and J. Nam, “Multi-level and multi-scale feature aggregation using pretrained convolutional neural networks for music auto-tagging,” *IEEE Signal Processing Letters*, vol. 24, no. 8, pp. 1208–1212, Aug. 2017.
- [268] S. Lee, J. Lee, and K. lee, “Content-based feature exploration for transparent music recommendation using self-attentive genre classification,” in *Proceedings of the Late-Breaking Results Track Part of the Twelfth ACM Conference on Recommender Systems (RecSys’18)*, arXiv, Sep. 3, 2018. arXiv: 1808.10600 [cs, eess].
- [269] J. Lee, K. Yoon, D. Jang, S.-J. Jang, S. Shin, and J.-H. Kim, “Music recommendation system based on genre distance and user preference classification,” *Journal of Theoretical & Applied Information Technology*, vol. 96, no. 5, 2018.
- [270] J. Lee, M. Lee, D. Jang, and K. Yoon, “Korean traditional music genre classification using sample and MIDI phrases,” *KSII Transactions on Internet and Information Systems*, vol. 12, no. 4, pp. 1869–1886, 2018.
- [271] J. Lee, J. Park, and J. Nam, “Representation Learning of Music Using Artist, Album, and Track Information,” presented at the Machine Learning for Music Discovery Workshop at ICML 2019, arXiv, Jun. 27, 2019. arXiv: 1906.11783 [cs, eess].
- [272] J. Lee, N. J. Bryan, J. Salamon, Z. Jin, and J. Nam, “Metric learning vs classification for disentangled music representation learning,” in *Proceedings of the 21st International Society for Music Information Retrieval Conference*, Montreal, Canada: ISMIR, Oct. 11, 2020, pp. 439–445.
- [273] A. Lefavre and J. Z. Zhang, “Music genre classification: Genre-specific characterization and pairwise evaluation,” in *Proceedings of the Audio Mostly 2018 on Sound in Immersion and Emotion*, ser. AM’18, New York, NY, USA: Association for Computing Machinery, Sep. 12, 2018, pp. 1–4.
- [274] M. Leimeister, D. Gaertner, and C. Dittmar, “Rhythmic Classification of Electronic Dance Music,” presented at the Audio Engineering Society Conference: 53rd International Conference: Semantic Audio, Audio Engineering Society, Jan. 27, 2014.
- [275] M. T. Leleuly and P. H. Gunawan, “Analysis of feature correlation for music genre classification,” in *2020 8th International Conference on Information and Communication Technology (ICoICT)*, Jun. 2020, pp. 1–4.
- [276] X. Li, “HouseX: A Fine-grained House Music Dataset and its Potential in the Music Industry,” presented at the Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC) 2022, Chiang Mai, Thailand: arXiv, Oct. 11, 2022. arXiv: 2207.11690 [cs, eess].
- [277] Y. Liang, Y. Zhou, T. Wan, and X. Shu, “Deep neural networks with depthwise separable convolution for music genre classification,” in *2019 IEEE 2nd International Conference on Information Communication and Signal Processing (ICICSP)*, Sep. 2019, pp. 267–270.
- [278] B. Liang and M. Gu, “Music genre classification using transfer learning,” in *2020 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, Aug. 2020, pp. 392–393.
- [279] C. Lin, W. Chen, C. Qiu, Y. Wu, S. Krishnan, and Q. Zou, “LibD3C: Ensemble classifiers with a clustering and dynamic selection strategy,” *Neurocomputing*, Advances in Pattern Recognition Applications and Methods, vol. 123, pp. 424–435, Jan. 10, 2014.
- [280] J. Liu, C. Wang, and L. Zha, “A middle-level learning feature interaction method with deep learning for multi-feature music genre classification,” *Electronics*, vol. 10, no. 18, p. 2206, 18 Jan. 2021.
- [281] C. Liu, L. Feng, G. Liu, H. Wang, and S. Liu, “Bottom-up broadcast neural network for music genre classification,” *Multimedia Tools and Applications*, vol. 80, no. 5, pp. 7313–7331, Feb. 1, 2021.
- [282] X. Liu, S. Song, M. Zhang, and Y. Huang, “MATT. A Multiple-instance Attention Mechanism for Long-tail Music Genre Classification,” in *2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Oct. 2022, pp. 782–787.

- [283] K. Liu, J. DeMori, and K. Abayomi. "Open set recognition for music genre classification." arXiv: 2209.07548 [eess, math]. (Sep. 15, 2022), [Online]. Available: <http://arxiv.org/abs/2209.07548> (visited on 03/03/2023), preprint.
- [284] Y.-L. Lo, C.-Y. Chiu, and T.-W. Chang, "Discovering Similar Music for Alpha Wave Music," in *Mobile and Wireless Technologies 2017*, K. J. Kim and N. Joukov, Eds., ser. Lecture Notes in Electrical Engineering, Singapore: Springer, 2018, pp. 571–580.
- [285] M. Long, L. Hu, and F. Jin, "Analysis of main characteristics of music genre based on PCA algorithm," in *2021 IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE)*, Mar. 2021, pp. 101–105.
- [286] Y.-C. Lu, C.-W. Wu, C.-T. Lu, and A. Lerch, "An unsupervised approach to anomaly detection in music datasets," in *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '16, New York, NY, USA: Association for Computing Machinery, Jul. 7, 2016, pp. 749–752.
- [287] Y.-C. Lu, C.-W. Wu, A. Lerch, and C.-T. Lu, "Automatic outlier detection in music genre datasets.," in *Proceedings of the 17th International Society for Music Information Retrieval Conference*, 2016, pp. 101–107.
- [288] H. Lukashovich, "Confidence Measures in Automatic Music Classification," in *Data Analysis, Machine Learning and Knowledge Discovery*, M. Spiliopoulou, L. Schmidt-Thieme, and R. Janning, Eds., ser. Studies in Classification, Data Analysis, and Knowledge Organization, Cham: Springer International Publishing, 2014, pp. 397–405.
- [289] A. Lykartsis, "Evaluation of accent-based rhythmic descriptors for genre classification of musical signals," M.S. thesis, Technische Universität Berlin, Berlin, 2014.
- [290] A. Lykartsis, C.-W. Wu, and A. Lerch, "Beat histogram features from NMF-based novelty functions for music classification.," in *Proceedings of the 16th International Society for Music Information Retrieval Conference*, Málaga, Spain: ISMIR, Oct. 2015, pp. 434–440.
- [291] Z. Ma, "Comparison between machine learning models and neural networks on music genre classification," in *2022 3rd International Conference on Computer Vision, Image and Deep Learning & International Conference on Computer Engineering and Applications (CVIDL & ICCEA)*, May 2022, pp. 189–194.
- [292] V. Macharla and P. Radha Krishna, "Music genre classification using neural networks with data augmentation," *Journal of Innovation Sciences and Sustainable Technologies*, vol. 1, no. 1, pp. 21–37, Jan. 12, 2021.
- [293] P. Mandal, I. Nath, N. Gupta, M. K. Jha, D. G. Ganguly, and S. Pal, "Automatic music genre detection using artificial neural networks," in *Intelligent Computing in Engineering*, V. K. Solanki, M. K. Hoang, Z. (Lu, and P. K. Pattnaik, Eds., ser. Advances in Intelligent Systems and Computing, Singapore: Springer, 2020, pp. 17–24.
- [294] K. Markov and T. Matsui, "Music genre classification using gaussian process models," in *2013 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, Sep. 2013, pp. 1–6.
- [295] K. Markov and T. Matsui, "High level feature extraction for the self-taught learning algorithm," *EURASIP Journal on Audio Speech and Music Processing*, vol. 2013, no. 1, p. 6, Apr. 9, 2013.
- [296] K. Markov and T. Matsui, "Music Genre and Emotion Recognition Using Gaussian Processes," *IEEE Access*, vol. 2, pp. 688–697, 2014.
- [297] G. C. Marques, "Machine Learning Techniques for Music Information Retrieval," Ph.D. dissertation, Universidade de Lisboa (Portugal), Portugal, 2014, 211 pp.
- [298] L. Maršík, J. Pokorný, and M. Ilčík, "Improving Music Classification Using Harmonic Complexity," in *Proceedings of the 14th Conference Information Technologies-Applications and Theory*, Prague, 2014.
- [299] J. Martel, T. Nakashika, C. Garcia, and K. Idrissi, "A Combination of Hand-Crafted and Hierarchical High-Level Learnt Feature Extraction for Music Genre Classification," in *Artificial Neural Networks and Machine Learning –ICANN 2013*, V. Mladenov, P. Koprinkova-Hristova, G. Palm, A. E. P. Villa, B. Appollini, and N. Kasabov, Eds., ser. Lecture Notes in Computer Science, Berlin, Heidelberg: Springer, 2013, pp. 397–404.
- [300] M. Matocha and S. K. Zieliński, "Music genre recognition using convolutional neural networks," *Advances in Computer Science Research*, vol. 14, pp. 125–142, 2018.
- [301] M. Mauch, R. M. MacCallum, M. Levy, and A. M. Leroi, "The evolution of popular music: USA 1960–2010," *Royal Society Open Science*, vol. 2, no. 5, p. 150 081, May 2015.
- [302] M. C. Mccallum, F. Korzeniowski, S. Oramas, F. Gouyon, and A. Ehmann, "Supervised and Unsupervised Learning of Audio Representations for Music Understanding," in *Proceedings of the 23rd International Society for Music Information Retrieval Conference*, Bengaluru, India: ISMIR, Dec. 4, 2022, pp. 256–263.

- [303] C. McKay, J. Cumming, and I. Fujinaga, “JSYMBOLIC 2.2: Extracting Features from Symbolic Music for use in Musicological and MIR Research,” in *Proceedings of the 19th International Society for Music Information Retrieval Conference*, Paris, France: ISMIR, Sep. 23, 2018, pp. 348–354.
- [304] F. Medhat, D. Chesmore, and J. Robinson, “Music genre classification using masked conditional neural networks,” in *Neural Information Processing*, D. Liu, S. Xie, Y. Li, D. Zhao, and E.-S. M. El-Alfy, Eds., ser. Lecture Notes in Computer Science, Cham: Springer International Publishing, 2017, pp. 470–481.
- [305] F. Medhat, D. Chesmore, and J. Robinson, “Masked conditional neural networks for audio classification,” in *Artificial Neural Networks and Machine Learning –ICANN 2017*, A. Lintas, S. Rovetta, P. F. Verschure, and A. E. Villa, Eds., ser. Lecture Notes in Computer Science, Cham: Springer International Publishing, 2017, pp. 349–358.
- [306] F. Medhat, D. Chesmore, and J. Robinson, “Automatic classification of music genre using masked conditional neural networks,” in *2017 IEEE International Conference on Data Mining (ICDM)*, Nov. 2017, pp. 979–984.
- [307] J. Mehta, D. Gandhi, G. Thakur, and P. Kanani, “Music genre classification using transfer learning on log-based MEL spectrogram,” in *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*, Apr. 2021, pp. 1101–1107.
- [308] J. Mendes, “Deep learning techniques for music genre classification and building a music recommendation system,” M.S. thesis, Dublin, National College of Ireland, 2020, 22 pp.
- [309] R. Mignot and G. Peeters, “An Analysis of the Effect of Data Augmentation Methods: Experiments for a Musical Genre Classification Task,” *Transactions of the International Society for Music Information Retrieval*, vol. 2, no. 1, pp. 97–110, Dec. 18, 2019.
- [310] J. Mitra and D. Saha. “An Efficient Feature Selection in Classification of Audio Files.” arXiv: 1404.1491 [cs]. (Mar. 24, 2014), [Online]. Available: <http://arxiv.org/abs/1404.1491> (visited on 02/19/2024), preprint.
- [311] K. S. Mounika, S. Deyaradevi, K. Swetha, and V. Vanitha, “Music genre classification using deep learning,” in *2021 International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA)*, Oct. 2021, pp. 1–7.
- [312] S. F. Mughal, S. Aamir, S. A. Sahto, and A. Samad, “Urdu music genre classification using convolution neural networks,” in *2022 International Conference on Emerging Trends in Smart Technologies (ICETST)*, Sep. 2022, pp. 1–6.
- [313] G. Mujtaba, S. Kim, E. Park, S. Kim, J. Ryu, and E.-S. Ryu, “Client-driven animated keyframe generation system using music analysis,” in *Proceedings of the Korean Society of Broadcast Engineers Conference*, The Korean Institute of Broadcast and Media Engineers, 2019, pp. 173–175.
- [314] S. Muñoz-Romero, J. A. García, and V. Gómez-Verdejo, “Nonnegative OPLS for Supervised Design of Filter Banks: Application to Image and Audio Feature Extraction,” *IEEE Transactions on Multimedia*, vol. 20, no. 7, pp. 1751–1766, Jul. 2018. arXiv: 2112.12280 [cs, eess].
- [315] B. Muraier and G. Specht, “Detecting music genre using extreme gradient boosting,” in *Companion Proceedings of the The Web Conference 2018*, ser. WWW ’18, Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, Apr. 23, 2018, pp. 1923–1927.
- [316] F. Nahar, K. Agres, B. BT, and D. Herremans, “A dataset and classification model for Malay, Hindi, Tamil and Chinese music,” in *Proceedings of the 13th International Workshop on Machine Learning and Music at ECML-PKDD 2020*, arXiv, Sep. 15, 2020. arXiv: 2009.04459 [cs, eess].
- [317] T. Nakai, N. Koide-Majima, and S. Nishimoto, “Encoding and decoding of music-genre representations in the human brain,” in *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Oct. 2018, pp. 584–589.
- [318] H. Nakamura, H.-H. Huang, and K. Kawagoe, “Detecting Musical Genre Borders for Multi-label Genre Classification,” in *2013 IEEE International Symposium on Multimedia*, Dec. 2013, pp. 532–533.
- [319] L. Nanni, Y. Costa, and S. Brahmam, “Set of texture descriptors for music genre classification,” presented at the 22nd International Conference in Central European Computer Graphics, Visualization and Computer Vision, Václav Skala - UNION Agency, 2014, pp. 145–152.
- [320] L. Nanni, Y. M. G. Costa, A. Lumini, M. Y. Kim, and S. R. Baek, “Combining visual and acoustic features for music genre classification,” *Expert Systems with Applications*, vol. 45, pp. 108–117, Mar. 1, 2016.
- [321] L. Nanni, Y. M. G. Costa, D. R. Lucio, C. N. Silla, and S. Brahmam, “Combining visual and acoustic features for audio classification tasks,” *Pattern Recognition Letters*, vol. 88, pp. 49–56, Mar. 1, 2017.

- [322] L. Nanni, Y. M. G. Costa, R. L. Aguiar, C. N. Silla, and S. Brahmam, "Ensemble of deep learning, visual and acoustic features for music genre classification," *Journal of New Music Research*, vol. 47, no. 4, pp. 383–397, Aug. 8, 2018.
- [323] N. Narkhede, S. Mathur, and A. Bhaskar, "Machine learning techniques for music genre classification," in *Information and Communication Technology for Competitive Strategies (ICTCS 2020)*, A. Joshi, M. Mahmud, R. G. Ragel, and N. V. Thakur, Eds., ser. Lecture Notes in Networks and Systems, Singapore: Springer, 2022, pp. 155–161.
- [324] N. Narkhede, S. Mathur, and A. Bhaskar, "Automatic classification of music genre using SVM," in *Computer Networks and Inventive Communication Technologies*, S. Smys, R. Bestak, R. Palanisamy, and I. Kotuliak, Eds., ser. Lecture Notes on Data Engineering and Communications Technologies, Singapore: Springer, 2022, pp. 439–449.
- [325] A. Nasridinov and Y.-H. Park, "A Study on Music Genre Recognition and Classification Techniques," *International Journal of Multimedia and Ubiquitous Engineering*, vol. 9, no. 4, pp. 31–42, Apr. 30, 2014.
- [326] N. Ndou, R. Ajoodha, and A. Jadhav, "Music genre classification: A review of deep-learning and traditional machine-learning approaches," in *2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS)*, Apr. 2021, pp. 1–6.
- [327] W. W. Y. Ng, W. Zeng, and T. Wang, "Multi-level local feature coding fusion for music genre recognition," *IEEE Access*, vol. 8, pp. 152 713–152 727, 2020.
- [328] Q. H. Nguyen *et al.*, "Music genre classification using residual attention network," in *2019 International Conference on System Science and Engineering (ICSSE)*, Jul. 2019, pp. 115–119.
- [329] K. Nie, "Inaccurate Prediction or Genre Evolution? Rethinking Genre Classification," in *Proceedings of the 23rd International Society for Music Information Retrieval Conference*, Bengaluru, India: ISMIR, Dec. 4, 2022, pp. 329–336.
- [330] M. R. Nirmal and S. Mohan B S, "Music genre classification using spectrograms," in *2020 International Conference on Power, Instrumentation, Control and Computing (PICC)*, Dec. 2020, pp. 1–5.
- [331] T. Nkambule and R. Ajoodha, "Classification of music by genre using probabilistic models and deep learning models," in *Proceedings of Sixth International Congress on Information and Communication Technology*, X.-S. Yang, S. Sherratt, N. Dey, and A. Joshi, Eds., ser. Lecture Notes in Networks and Systems, Singapore: Springer, 2022, pp. 185–193.
- [332] S. Ntalampiras, "Directed acyclic graphs for content based sound, musical genre, and speech emotion classification," *Journal of New Music Research*, vol. 43, no. 2, pp. 173–182, Apr. 3, 2014.
- [333] C. J. O'Brien, "Supervised feature learning via sparse coding for music information retrieval," M.S. thesis, Georgia Tech, Apr. 24, 2015.
- [334] S. Oramas, L. Espinosa-Anke, A. Lawlor, X. Serra, and H. Saggion, "Exploring customer reviews for music genre classification and evolutionary studies," in *Proceedings of the 17th International Society for Music Information Retrieval Conference*, 2016, pp. 150–156.
- [335] S. Oramas, O. Nieto, F. Barbieri, and X. Serra, "Multi-label music genre classification from audio, text, and images using deep features," in *Proceedings of the 18th International Society for Music Information Retrieval Conference*, Suzhou, China: ISMIR, Jul. 16, 2017, pp. 23–30. arXiv: 1707.04916 [cs].
- [336] S. Oramas, "Knowledge extraction and representation learning for music recommendation and classification," Ph.D. dissertation, Universitat Pompeu Fabra, Barcelona, Spain, Nov. 14, 2017.
- [337] S. Oramas, F. Barbieri, O. Nieto, and X. Serra, "Multimodal Deep Learning for Music Genre Classification," *Transactions of the International Society for Music Information Retrieval*, vol. 1, no. 1, pp. 4–21, 1 Sep. 4, 2018.
- [338] R. Ozakar and E. Gedikli, "Music genre classification using novel song structure derived features," in *2020 5th International Conference on Computer Science and Engineering (UBMK)*, Sep. 2020, pp. 117–120.
- [339] T. Özseven and B. E. Özseven, "A content analysis of the research approaches in music genre recognition," in *2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, Jun. 2022, pp. 1–13.
- [340] H. Palmason, B. Þ. Jónsson, L. Amsaleg, M. Schedl, and P. Knees, "On competitiveness of nearest-neighbor-based music classification: A methodological critique," in *Similarity Search and Applications*, C. Beecks, F. Borutta, P. Kröger, and T. Seidl, Eds., ser. Lecture Notes in Computer Science, Cham: Springer International Publishing, 2017, pp. 275–283.
- [341] H. Palmason, B. Þ. Jónsson, M. Schedl, and P. Knees, "Music genre classification revisited: An in-depth examination guided by music experts," in *Proceedings of the International Symposium on Computer Music Multidisciplinary Research*, M. Aramaki, M. E. P. Davies, R. Kronland-Martinet, and S. Ystad, Eds., ser. Lecture Notes in Computer Science, Cham: Springer International Publishing, 2018, pp. 49–62.

- [342] Y. Panagakis and C. Kotropoulos, "Music classification by low-rank semantic mappings," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2013, no. 1, p. 13, Jun. 24, 2013.
- [343] Y. Panagakis, C. L. Kotropoulos, and G. R. Arce, "Music genre classification via joint sparse low-rank representation of audio features," *IEEE/ACM Transactions on Audio Speech and Language Processing*, vol. 22, no. 12, pp. 1905–1917, Dec. 2014.
- [344] M. M. Panchwagh and V. D. Katkar, "Music genre classification using data mining algorithm," in *2016 Conference on Advances in Signal Processing (CASP)*, Jun. 2016, pp. 49–53.
- [345] A. Pandey and I. Dutta, "Bundeli Folk-Song Genre Classification with kNN and SVM," in *Proceedings of the 11th International Conference on Natural Language Processing*, Goa, India, 2014, pp. 133–138.
- [346] Y. R. Pandeya, J. You, B. Bhattarai, and J. Lee, "Multi-modal, multi-task and multi-label for music genre classification and emotion regression," in *2021 International Conference on Information and Communication Technology Convergence (ICTC)*, Oct. 2021, pp. 1042–1045.
- [347] S. Panwar, A. Das, M. Roopaei, and P. Rad, "A deep learning approach for mapping music genres," in *2017 12th System of Systems Engineering Conference (SoSE)*, Jun. 2017, pp. 1–5.
- [348] C. Papaioannou, I. Valiantzas, T. Giannakopoulos, M. Kaliakatsos-Papakostas, and A. Potamianos, "A Dataset for Greek Traditional and Folk Music: Lyra," in *Proceedings of the 23rd International Society for Music Information Retrieval Conference*, Bengaluru, India: ISMIR, Dec. 4, 2022, pp. 377–383.
- [349] J. Park, J. Lee, J. Park, J.-W. Ha, and J. Nam, "Representation learning of music using artist labels," in *Proceedings of the 19th International Society for Music Information Retrieval Conference*, ISMIR, 2018, pp. 717–724.
- [350] S. Park, I. Kim, and K. Ahn, "A stochastic process for music: The example of K-pop music," *Journal of Physics Conference Series*, vol. 2287, no. 1, p. 012 010, Jun. 2022.
- [351] A. R. S. Parmezan, D. F. Silva, and G. E. Batista, "A combination of local approaches for hierarchical music genre classification," in *Proceedings of the 21st International Society for Music Information Retrieval Conference*, Montreal, Canada: ISMIR, 2020, pp. 740–747.
- [352] N. M. Patil and M. U. Nemade, "Music genre classification using MFCC, K-NN and SVM classifier," *International Journal of Computer Engineering in Research Trends*, vol. 4, no. 2, pp. 43–47, 2017.
- [353] V. Pavan and R. Dhanalakshmi, "Analysis of Audio Data and Prediction of the Genre using Novel Random Forest and Decision Tree," in *2022 4th International Conference on Inventive Research in Computing Applications (ICIRCA)*, Sep. 2022, pp. 1773–1777.
- [354] G. Peeters, "Rhythm classification using spectral rhythm patterns," in *Proceedings of the 6th International Conference on Music Information Retrieval*, 2005, pp. 644–647.
- [355] I. A. Pegoraro Santana *et al.*, "Music4all: A new music database and its applications," in *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*, Jul. 2020, pp. 399–404.
- [356] R. Peiris and L. Jayaratne, "Supervised learning approach for classification of Sri Lankan music based on music structure similarity," in *Proceedings of Ninth Annual International Conference on Computer Games, Multimedia and Allied Technology CGAT 2016*, Singapore, 2016, pp. 84–90.
- [357] R. Peiris and L. Jayaratne, "Musical genre classification of recorded songs based on music structure similarity," *European Journal of Computer Science and Information Technology*, vol. 4, no. 5, pp. 70–88, 2016.
- [358] N. Pelchat and C. M. Gelowitz, "Neural network music genre classification," *Canadian Journal of Electrical and Computer Engineering*, vol. 43, no. 3, pp. 170–173, 2020.
- [359] R. M. Pereira and C. N. Silla, "Using simplified chords sequences to classify songs genres," in *2017 IEEE International Conference on Multimedia and Expo (ICME)*, Jul. 2017, pp. 1446–1451.
- [360] R. M. Pereira, Y. M. G. Costa, R. L. Aguiar, A. S. Britto, L. E. S. Oliveira, and C. N. Silla, "Representation learning vs. Handcrafted features for music genre classification," in *2019 International Joint Conference on Neural Networks (IJCNN)*, Jul. 2019, pp. 1–8.
- [361] L. V. d. A. S. Pereira and T. F. Tavares, "An interplay between genre and emotion prediction in music: A study in the Emotify dataset," in *Anais Do Simpósio Brasileiro de Computação Musical (SBCM)*, SBC, Oct. 24, 2021, pp. 25–29.
- [362] H. C. Piccoli, C. N. Silla, P. J. P. de Léon, and A. Pertusa, "An evaluation of symbolic feature sets and their combination for music genre classification," in *2013 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Oct. 2013, pp. 1901–1905.
- [363] M. H. Pimenta-Zanon, G. M. Bressan, and F. M. Lopes. "Complex Network-Based Approach for Feature Extraction and Classification of Musical Genres." arXiv: 2110.04654 [cs, eess]. (Oct. 9, 2021), [Online]. Available: <http://>

- arxiv.org/abs/2110.04654 (visited on 03/24/2023), preprint.
- [364] J. Pons and X. Serra, "Randomly Weighted CNNs for (Music) Audio Classification," in *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, United Kingdom: IEEE, May 2019, pp. 336–340.
- [365] R. Popovici and R. Andonie, "Music genre classification with self-organizing maps and edit distance," in *2015 International Joint Conference on Neural Networks (IJCNN)*, Jul. 2015, pp. 1–7.
- [366] S. Poria, A. Gelbukh, A. Hussain, S. Bandyopadhyay, and N. Howard, "Music Genre Classification: A Semi-supervised Approach," in *Pattern Recognition*, J. A. Carrasco-Ochoa, J. F. Martínez-Trinidad, J. S. Rodríguez, and G. S. di Baja, Eds., ser. Lecture Notes in Computer Science, Berlin, Heidelberg: Springer, 2013, pp. 254–263.
- [367] A. Porter, D. Bogdanov, and X. Serra, "Mining metadata from the web for AcousticBrainz," in *Proceedings of the 3rd International Workshop on Digital Libraries for Musicology*, ser. DLfM 2016, New York, NY, USA: Association for Computing Machinery, Aug. 12, 2016, pp. 53–56.
- [368] N. Prabhu, A. Asnodkar, and R. Kenkre, "Music Genre Classification using Improved Artificial Neural Network with Fixed Size Momentum," *International Journal of Computer Applications*, vol. 101, no. 14, pp. 25–30, Sep. 18, 2014.
- [369] N. Prabhu, A. Asnodkar, and R. Kenkre, "Multi-class Support Class Support Vector Machine for Music Genre Classification," *International Journal of Computer Applications*, vol. 107, no. 19, pp. 15–17, Dec. 18, 2014.
- [370] N. R. Prabhu, J. Andro-Vasko, D. Bein, and W. Bein, "Music genre classification using data mining and machine learning," in *Information Technology - New Generations*, S. Latifi, Ed., ser. Advances in Intelligent Systems and Computing, Cham: Springer International Publishing, 2018, pp. 397–403.
- [371] R. Prajwal, S. Sharma, P. Naik, and S. Mk, "Music genre classification using machine learning," *International Research Journal of Modernization in Engineering Technology and Science*, vol. 3, no. 7, pp. 953–957, 2021.
- [372] P. R. M. Prasetyaa and G. A. V. M. Giria, "Comparison of use of music content (tempo) and user context (mood) features on classification of music genre," *JELIKU (Jurnal Elektronik Ilmu Komputer Udayana)*, vol. 2301, p. 5373, 2019.
- [373] V. Prashanthi, S. Kanakala, V. Akila, and A. Harshavardhan, "Music genre categorization using machine learning algorithms," in *2021 International Conference on Computational Intelligence and Computing Applications (ICCICA)*, Nov. 2021, pp. 1–4.
- [374] F. Prezja, "Developing and testing sub-band spectral features in music genre and music mood machine learning," M.S. thesis, University of Jyväskylä, Finland, 2018.
- [375] M. Prockup, A. F. Ehmann, F. Gouyon, E. M. Schmidt, Ö. Celma, and Y. E. Kim, "Modeling Genre with the Music Genome Project: Comparing Human-Labeled Attributes and Audio Features.," in *Proceedings of the 16th International Society for Music Information Retrieval Conference*, Málaga, Spain: ISMIR, Oct. 26, 2015, pp. 31–37.
- [376] L. K. Puppala, S. S. R. Muvva, S. R. Chinige, and P. Rajendran, "A novel music genre classification using convolutional neural network," in *2021 6th International Conference on Communication and Electronics Systems (ICCES)*, Jul. 2021, pp. 1246–1249.
- [377] N. Purnama, "Music genre recommendations based on spectrogram analysis using convolutional neural network algorithm with RESNET-50 and VGG-16 architecture," *JISA (Jurnal Informatika dan Sains)*, vol. 5, no. 1, pp. 69–74, 1 Jun. 20, 2022.
- [378] Z. Qi, M. Rahouti, M. A. Jasim, and N. Siasi, "Music genre classification and feature comparison using ML," in *2022 7th International Conference on Machine Learning Technologies (ICMLT)*, ser. ICMLT 2022, New York, NY, USA: Association for Computing Machinery, Jun. 10, 2022, pp. 42–50.
- [379] Z. Qin, W. Liu, and T. Wan, "A Bag-of-Tones Model with MFCC Features for Musical Genre Classification," in *Advanced Data Mining and Applications*, H. Motoda, Z. Wu, L. Cao, O. Zaiane, M. Yao, and W. Wang, Eds., ser. Lecture Notes in Computer Science, Berlin, Heidelberg: Springer, 2013, pp. 564–575.
- [380] C. Qin, H. Yang, W. Liu, S. Ding, and Y. Geng, "Music genre trend prediction based on spatial-temporal music influence and euclidean similarity," in *2021 36th Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, May 2021, pp. 406–411.
- [381] L. Qiu, S. Li, and Y. Sung, "3D-DCDAE: Unsupervised music latent representations learning method based on a deep 3D convolutional denoising autoencoder for music genre classification," *Mathematics*, vol. 9, no. 18, p. 2274, 18 Jan. 2021.

- [382] L. Qiu, S. Li, and Y. Sung, “DBTMPE: Deep bidirectional transformers-based masked predictive encoder approach for music genre classification,” *Mathematics*, vol. 9, no. 5, p. 530, 5 Jan. 2021.
- [383] R. J. M. Quinto, R. O. Atienza, and N. M. C. Tiglaio, “Jazz music sub-genre classification using deep learning,” in *TENCON 2017 - 2017 IEEE Region 10 Conference*, Nov. 2017, pp. 3111–3116.
- [384] S. A. Raczyński and E. Vincent, “Genre-based music language modeling with latent hierarchical pitman-yor process allocation,” *IEEE/ACM Transactions on Audio Speech and Language Processing*, vol. 22, no. 3, pp. 672–681, Mar. 2014.
- [385] Q. G. Rafi, M. Noman, S. Z. Proadhan, S. Alam, and D. Nandi, “Comparative analysis of three improved deep learning architectures for music genre classification,” *International Journal of Information Technology and Computer Science*, vol. 13, no. 2, pp. 1–14, 2021.
- [386] T. Raissi, A. Tibo, and P. Bientinesi, “Extended pipeline for content-based feature engineering in music genre recognition,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2018, pp. 2661–2665.
- [387] H. Raj, A. K. Dubey, V. Deep, and Anuranjana, “Music genre classification using machine learning,” in *Advances in Interdisciplinary Engineering*, N. Kumar, S. Tibor, R. Sindhvani, J. Lee, and P. Srivastava, Eds., ser. Lecture Notes in Mechanical Engineering, Singapore: Springer, 2021, pp. 763–774.
- [388] R. Rajan and H. A. Murthy, “Music genre classification by fusion of modified group delay and melodic features,” in *2017 Twenty-third National Conference on Communications (NCC)*, Mar. 2017, pp. 1–6.
- [389] A. R. Rajanna, K. Aryafar, A. Shokoufandeh, and R. Ptucha, “Deep neural networks: A case study for music genre classification,” in *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, Dec. 2015, pp. 655–660.
- [390] B. Rajesh and D. G. Bhalke, “Automatic genre classification of Indian Tamil and western music using fractional MFCC,” *International Journal of Speech Technology*, vol. 19, no. 3, pp. 551–563, Sep. 1, 2016.
- [391] A. Ramanathan, P. Srivastava, and R. Jeya, “Machine learning in music genre classification,” *Turkish Online Journal of Qualitative Inquiry*, vol. 12, no. 3, pp. 2494–2510, 3 Jul. 1, 2021.
- [392] P. Rameshkumar, M. Monisha, B. Santhi, and T. Vigneshwaran, “Robust feature selection method for music classification,” in *2014 International Conference on Computer Communication and Informatics*, Jan. 2014, pp. 1–6.
- [393] J. Ramírez and M. J. Flores, “Machine learning for music genre: Multifaceted review and experimentation with audioset,” *Journal of Intelligent Information Systems*, vol. 55, no. 3, pp. 469–499, Dec. 1, 2020.
- [394] F. Raposo, R. Ribeiro, and D. M. de Matos, “On the Application of Generic Summarization Algorithms to Music,” *IEEE Signal Processing Letters*, vol. 22, no. 1, pp. 26–30, Jan. 2015.
- [395] F. Raposo, R. Ribeiro, and D. Martins de Matos, “Using Generic Summarization to Improve Music Information Retrieval Tasks,” *IEEE/ACM Transactions on Audio Speech and Language Processing*, vol. 24, no. 6, pp. 1119–1128, Jun. 2016.
- [396] M. Raval, P. Dave, and R. Dattani, “Music genre classification using neural networks,” *International Journal of Advanced Research in Computer Science*, vol. 12, no. 5, 2021.
- [397] T. Ren, F. Wang, and H. Wang, “A sequential naive bayes method for music genre classification based on transitional information from pitch and beat,” *Statistics and Its Interface*, vol. 13, no. 3, pp. 361–371, 2020.
- [398] S. S. Renteria Aguilar, L. Llano, and J. Cantú-Ortiz, “Data-driven techniques for music genre recognition,” in *Computer Science & Information Technology*, D. C. Wyld and D. Nagamalai, Eds., Academy and Industry Research Collaboration Center (AIRCC), Jul. 11, 2020.
- [399] F. Rodrigues, F. Pereira, and B. Ribeiro, “Learning from multiple annotators: Distinguishing good from random labelers,” *Pattern Recognition Letters*, vol. 34, no. 12, pp. 1428–1436, Sep. 1, 2013.
- [400] F. Rodríguez-Algarra, B. L. Sturm, and H. Maruri-Aguilar, “Analysing scattering-based music content analysis systems: Where’s the music?” In *Proceedings of the 17th International Society for Music Information Retrieval Conference*, ISMIR, 2016, pp. 344–350.
- [401] F. Rodríguez-Algarra, B. L. Sturm, and S. Dixon, “Characterising confounding effects in music classification experiments through interventions,” *Transactions of the International Society for Music Information Retrieval*, vol. 2, no. 1, pp. 52–66, 2019.
- [402] L. Rompré, I. Biskri, and J.-G. Meunier, “Using association rules mining for retrieving genre-specific music files,” in *The Thirtieth International Flairs Conference*, 2017.
- [403] A. Rosner, M. Michalak, and B. Kostek, “A study on influence of normalization methods on music genre classification results employing kNN algorithms,” *Studia Informatica Pomerania*, vol. 34, pp. 411–423, 2013.

- [404] A. Rosner, F. Weninger, B. Schuller, M. Michalak, and B. Kostek, "Influence of low-level features extracted from rhythmic and harmonic sections on music genre classification," in *Man-Machine Interactions 3*, D. A. Gruca, T. Czachórski, and S. Kozielski, Eds., ser. Advances in Intelligent Systems and Computing, Cham: Springer International Publishing, 2014, pp. 467–473.
- [405] A. Rosner, B. Schuller, and B. Kostek, "Classification of music genres based on music separation into harmonic and drum components," *Archives of Acoustics*, vol. 39, no. 4, pp. 629–638, 4 Dec. 8, 2014.
- [406] A. Rosner and B. Kostek, "Musical instrument separation applied to music genre classification," in *Foundations of Intelligent Systems*, F. Esposito, O. Pivert, M.-S. Hacid, Z. W. Rás, and S. Ferilli, Eds., ser. Lecture Notes in Computer Science, Cham: Springer International Publishing, 2015, pp. 420–430.
- [407] A. Rosner and B. Kostek, "Automatic music genre classification based on musical instrument track separation," *Journal of Intelligent Information Systems*, vol. 50, no. 2, pp. 363–384, Apr. 1, 2018.
- [408] S. Saju, R. Rajan, and A. R. Jayan, "Music genre classification using spatan 6 FPGA and TMS320C6713 DSK," in *2017 International Conference on Signal Processing and Communication (ICSPC)*, Jul. 2017, pp. 196–200.
- [409] A. E. Coca Salazar, "Hierarchical mining with complex networks for music genre classification," *Digital Signal Processing*, vol. 127, p. 103 559, Jul. 1, 2022.
- [410] A. Saravanou, F. Tomasi, R. Mehrotra, and M. Lalmas, "Multi-Task Learning of Graph-based Inductive Representations of Music Content," in *Proceedings of the 22nd International Society for Music Information Retrieval Conference*, Online: ISMIR, Nov. 7, 2021, pp. 602–609.
- [411] R. Sarkar and S. K. Saha, "Music genre classification using EMD and pitch based feature," in *2015 Eighth International Conference on Advances in Pattern Recognition (ICAPR)*, Jan. 2015, pp. 1–6.
- [412] R. Sarkar, N. Biswas, and S. Chakraborty, "Music genre classification using frequency domain features," in *2018 Fifth International Conference on Emerging Applications of Information Technology (EAIT)*, Jan. 2018, pp. 1–4.
- [413] M. Sathyamurthy, X. Dong, and M. P. Kumar, "Geometric deep learning for music genre classification," in *Proceedings of the 13th International Workshop on Machine Learning and Music*, 2020, pp. 28–31.
- [414] C. Savard, E. H. Bugbee, M. R. McGuirl, and K. M. Kinnaird, "SuPP & MaPP: Adaptable structure-based representations for MIR tasks," in *Proceedings of the 21st International Society for Music Information Retrieval Conference*, Montreal, Canada: ISMIR, Oct. 11, 2020, pp. 335–342.
- [415] Y. Sazaki and A. Aramadhan, "Rock Genre Classification using K-Nearest Neighbor," in *Proceeding of The 1st International Conference on Computer Science and Engineering*, 2014, pp. 81–83.
- [416] S. Scardapane, D. Comminiello, M. Scarpiniti, and A. Uncini, "Music classification using extreme learning machines," in *2013 8th International Symposium on Image and Signal Processing and Analysis (ISPA)*, Sep. 2013, pp. 377–381.
- [417] S. Scardapane, R. Fierimonte, D. Wang, M. Panella, and A. Uncini, "Distributed music classification using Random Vector Functional-Link nets," in *2015 International Joint Conference on Neural Networks (IJCNN)*, Jul. 2015, pp. 1–8.
- [418] M. Scarpiniti, S. Scardapane, D. Comminiello, and A. Uncini, "Music genre classification using stacked auto-encoders," in *Neural Approaches to Dynamics of Signal Exchanges*, ser. Smart Innovation, Systems and Technologies, A. Esposito, M. Faundez-Zanuy, F. C. Morabito, and E. Pasero, Eds., Singapore: Springer, 2020, pp. 11–19.
- [419] M. Schedl and C. Bauer, "Online music listening culture of kids and adolescents: Listening analysis and music recommendation tailored to the young," in *RecSys '17: Proceedings of the Eleventh ACM Conference on Recommender Systems*, arXiv, Dec. 24, 2019, pp. 376–377. arXiv: 1912.11564 [cs].
- [420] A. Schindler and A. Rauber, "An audio-visual approach to music genre classification through affective color features," in *Advances in Information Retrieval*, A. Hanbury, G. Kazai, A. Rauber, and N. Fuhr, Eds., ser. Lecture Notes in Computer Science, Cham: Springer International Publishing, 2015, pp. 61–67.
- [421] A. Schindler and A. Rauber, "Harnessing music-related visual stereotypes for music information retrieval," *ACM Transactions on Intelligent Systems and Technology*, vol. 8, no. 2, 20:1–20:21, Oct. 25, 2016.
- [422] A. Schindler, T. Lidy, and A. Rauber, "Comparing shallow versus deep neural network architectures for automatic music genre classification," in *Forum Media Technology*, 2016, pp. 17–21.
- [423] D. Schnitzer, A. Flexer, M. Schedl, and G. Widmer, "Local and Global Scaling Reduce Hubs in Space," *Journal of Machine Learning Research*, vol. 13, no. 92, pp. 2871–2902, 2012.

- [424] H. Schreiber, "Improving genre annotations for the million song dataset.," in *Proceedings of the 16th International Society for Music Information Retrieval Conference*, Málaga, Spain, 2015, pp. 241–247.
- [425] H. Schreiber, "Genre Ontology Learning: Comparing Curated with Crowd-Sourced Ontologies.," in *Proceedings of the 17th International Society for Music Information Retrieval Conference*, New York City, United States: ISMIR, Aug. 7, 2016, pp. 400–406.
- [426] B. Schuller, F. Eyben, and G. Rigoll, "Tango or waltz?: Putting ballroom dance style into tempo detection," *EURASIP Journal on Audio Speech and Music Processing*, vol. 2008, no. 1, pp. 1–12, 1 Dec. 2008.
- [427] A. Sen, "Automatic Music Clustering using Audio Attributes," *International Journal of Computer Science Engineering*, vol. 3, no. 6, pp. 307–312, 2014.
- [428] C. Senac, T. Pellegrini, F. Mouret, and J. Piquier, "Music feature maps with convolutional neural networks for music genre classification," in *Proceedings of the 15th International Workshop on Content-Based Multimedia Indexing*, ser. CBMI '17, New York, NY, USA: Association for Computing Machinery, Jun. 19, 2017, pp. 1–5.
- [429] M. Sen Sarma and A. Das, "BMGC: A deep learning approach to classify bengali music genres," in *Proceedings of the 4th International Conference on Networking, Information Systems & Security*, ser. NISS2021, New York, NY, USA: Association for Computing Machinery, Nov. 26, 2021, pp. 1–6.
- [430] J. S. Seo, "A Musical Genre Classification Method Based on the Octave-Band Order Statistics," *The Journal of the Acoustical Society of Korea*, vol. 33, no. 1, pp. 81–86, 2014.
- [431] M. Serwach and B. Stasiak, "GA-based parameterization and feature selection for automatic music genre recognition," in *2016 17th International Conference Computational Problems of Electrical Engineering (CPEE)*, Sep. 2016, pp. 1–5.
- [432] D. R. I. M. Setiadi *et al.*, "Comparison of SVM, KNN, and NB classifier for genre music classification based on metadata," in *2020 International Seminar on Application for Technology of Information and Communication (iSemantic)*, Sep. 2020, pp. 12–16.
- [433] D. R. I. M. Setiadi *et al.*, "Effect of feature selection on the accuracy of music genre classification using SVM classifier," in *2020 International Seminar on Application for Technology of Information and Communication (iSemantic)*, Sep. 2020, pp. 7–11.
- [434] K. Seyerlehner, M. Schedl, R. Sonnleitner, D. Hauger, and B. Ionescu, "From Improved Auto-Taggers to Improved Music Similarity Measures," in *Adaptive Multimedia Retrieval: Semantics, Context, and Adaptation*, A. Nürnberger, S. Stober, B. Larsen, and M. Detyniecki, Eds., ser. Lecture Notes in Computer Science, Cham: Springer International Publishing, 2014, pp. 193–202.
- [435] D. Shah, C. Sachdev, and B. Shah, "Classification of music genre using neural networks with cross-entropy optimization and soft-max output," *International Journal of Computer Applications*, vol. 119, no. 12, 2015.
- [436] M. Shah, N. Pujara, K. Mangaroliya, L. Gohil, T. Vyas, and S. Degadwala, "Music genre classification using deep learning," in *2022 6th International Conference on Computing Methodologies and Communication (ICCMC)*, Mar. 2022, pp. 974–978.
- [437] T. Shaikh and A. Jadhav, "Music genre classification using neural network," in *ITM Web of Conferences*, vol. 44, EDP Sciences, 2022, p. 03 016.
- [438] A. Shakya, B. Gurung, M. S. Thapa, M. Rai, and B. Joshi, "Music classification based on genre and mood," in *Computational Intelligence, Communications, and Business Analytics*, J. K. Mandal, P. Dutta, and S. Mukhopadhyay, Eds., ser. Communications in Computer and Information Science, Singapore: Springer, 2017, pp. 168–183.
- [439] S. Sharma, P. Fulzele, and I. Sreedevi, "Novel hybrid model for music genre classification based on support vector machine," in *2018 IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE)*, Apr. 2018, pp. 395–400.
- [440] H. L. Shashirekha, "Using MFCC Features for the Classification of Monophonic Music," in *International Conference on Information and Communication Technologies (ICICT- 2014)*, 2014, pp. 5–9.
- [441] M. Sheikh Fathollahi and F. Razzazi, "Music similarity measurement and recommendation system using convolutional neural networks," *International Journal of Multimedia Information Retrieval*, vol. 10, no. 1, pp. 43–53, Mar. 1, 2021.
- [442] L. Shi, C. Li, and L. Tian, "Music genre classification based on chroma features and deep learning," in *2019 Tenth International Conference on Intelligent Control and Information Processing (ICI-CIP)*, Dec. 2019, pp. 81–86.
- [443] S.-H. Shin, H.-W. Yun, W.-J. Jang, and H. Park, "Extraction of acoustic features based on auditory spike code and its application to music genre classification," *IET Signal Processing*, vol. 13, no. 2, pp. 230–234, 2019.

- [444] S. S. Shinde and D. S. L. Nalbalwar, "Feature Extraction and Wavelet Analysis in Musical Genre Categorization," *International Journal of Scientific Engineering and Technology Research*, vol. 3, no. 18, pp. 3759–3763, 2014.
- [445] A. Sibi, R. Singh, K. Anurag, A. Choudhary, A. Prakash Agrawal, and G. Raj, "Music genre classification using light gradient boosting machine: A pilot study," in *Machine Intelligence and Data Science Applications*, V. Skala, T. P. Singh, T. Choudhury, R. Tomar, and Md. Abul Bashar, Eds., ser. Lecture Notes on Data Engineering and Communications Technologies, Singapore: Springer Nature, 2022, pp. 733–748.
- [446] I. Siddavatam, A. Dalvi, D. Gupta, Z. Farooqui, and M. Chouhan, "Multi genre music classification and conversion system," *International Journal of Information Engineering and Electronic Business*, vol. 12, no. 1, p. 30, 2020.
- [447] S. Sigtia and S. Dixon, "Improved music feature learning with deep neural networks," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, pp. 6959–6963.
- [448] D. F. Silva, R. G. Rossi, S. O. Rezende, and G. E. D. A. P. A. Batista, "Music classification by transductive learning using bipartite heterogeneous networks," in *Proceedings of the 15th International Society for Music Information Retrieval Conference*, H.-M. Wang, Y.-H. Yang, and J. H. Lee, Eds., 2014, pp. 113–118.
- [449] A. C. M. da Silva, M. A. N. Coelho, and R. F. Neto, "A Music Classification model based on metric learning applied to MP3 audio files," *Expert Systems with Applications*, vol. 144, p. 113 071, Apr. 15, 2020.
- [450] D. F. Silva, A. C. M. da Silva, L. F. Ortolan, and R. M. Maracacini, "On generalist and domain-specific music classification models and their impacts on brazilian music genre recognition," in *Anais Do XVIII Simpósio Brasileiro de Computação Musical*, SBC, 2021, pp. 60–67.
- [451] D. Silva, M. Silva, R. S. Filho, and A. Silva, "On the fusion of multiple audio representations for music genre classification," in *Anais Do XVIII Simpósio Brasileiro de Computação Musical*, Porto Alegre, RS, Brasil: SBC, 2021, pp. 37–44.
- [452] E. F. Simas Filho, E. A. Borges Jr, and A. C. Fernandes Jr, "Genre classification for brazilian music using independent and discriminant features," *Journal of Communication and Information Systems*, vol. 33, no. 1, 2018.
- [453] A. J. Sinclair, "Predicting music genre preferences based on online comments," M.S. thesis, California Polytechnic State University, San Luis Obispo, California, Jun. 1, 2014.
- [454] A. Singh and S. Ramanna, "Application of tolerance near sets to audio signal classification," in *Advances in Feature Selection for Data and Pattern Recognition*, ser. Intelligent Systems Reference Library, U. Stańczyk, B. Zielosko, and L. C. Jain, Eds., Cham: Springer International Publishing, 2018, pp. 241–266.
- [455] J. Singh and V. K. Bohat, "Neural network model for recommending music based on music genres," in *2021 International Conference on Computer Communication and Informatics (ICCCI)*, Jan. 2021, pp. 1–6.
- [456] Y. Singh and A. Biswas, "Robustness of musical features on deep learning models for music genre classification," *Expert Systems with Applications*, vol. 199, p. 116 879, Aug. 1, 2022.
- [457] P. Siva Sankalp, T. Baruah, S. Tiwari, and S. Sankar Ganesh, "Intelligent classification of electronic music," in *2014 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, Dec. 2014, pp. 000 031–000 035.
- [458] M. Skowron, F. Lemmerich, B. Ferwerda, and M. Schedl, "Predicting genre preferences from cultural and socio-economic factors for music retrieval," in *Advances in Information Retrieval*, J. M. Jose *et al.*, Eds., ser. Lecture Notes in Computer Science, Cham: Springer International Publishing, 2017, pp. 561–567.
- [459] L. Soboh, I. Elkabani, and Z. Osman, "Arabic cultural style based music classification," in *2017 International Conference on New Trends in Computing Sciences (ICTCS)*, Oct. 2017, pp. 6–11.
- [460] S. I. Sohel, C. Mondol, H. S. Ayon, U. T. Islam, and M. K. Morol, "Music suggestions from determining the atmosphere of images," in *2021 24th International Conference on Computer and Information Technology (ICCIT)*, Dec. 2021, pp. 1–7.
- [461] G. Song, Z. Wang, F. Han, and S. Ding, "Transfer learning for music genre classification," in *Intelligence Science I*, Z. Shi, B. Goertzel, and J. Feng, Eds., ser. IFIP Advances in Information and Communication Technology, Cham: Springer International Publishing, 2017, pp. 183–190.
- [462] A. Soriano, F. Paulovich, L. G. Nonato, and M. C. F. Oliveira, "Visualization of Music Collections Based on Structural Content Similarity," in *2014 27th SIBGRAPI Conference on Graphics, Patterns and Images*, Brazil: IEEE, Aug. 2014, pp. 25–32.
- [463] M. Srinivas, D. Roy, and C. K. Mohan, "Music genre classification using on-line dictionary learning," in *2014 International Joint Conference on Neural Networks (IJCNN)*, Jul. 2014, pp. 1937–1941.

- [464] S. Sruthi and S. Sridhar, "Music genre predictor based classification of audio files with low level feature of frequency and time domain using support vector machine over K-means clustering algorithm," in *2022 Sixth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, Nov. 2022, pp. 268–273.
- [465] S. Stern, "Analysis of music genre clustering algorithms," M.S. thesis, The University of Wisconsin - Milwaukee, United States – Wisconsin, 2021, 25 pp.
- [466] W. Stokowiec, "A comparative study on music genre classification algorithms," in *Machine Intelligence and Big Data in Industry*, ser. Studies in Big Data, D. Ryżko, P. Gawrysiak, M. Kryszkiewicz, and H. Rybiński, Eds., Cham: Springer International Publishing, 2016, pp. 123–132.
- [467] B. L. Sturm. "The GTZAN dataset: Its contents, its faults, their effects on evaluation, and its future use." arXiv: 1306.1461. (2013), [Online]. Available: <http://arxiv.org/abs/1306.1461>, preprint.
- [468] B. L. Sturm and F. Gouyon, "Revisiting inter-genre similarity," *IEEE Signal Processing Letters*, vol. 20, no. 11, pp. 1050–1053, Nov. 2013.
- [469] B. L. Sturm, "Evaluating music emotion recognition: Lessons from music genre recognition?" In *Proceedings of the 2013 IEEE International Conference on Multimedia and Expo (ICME)*, 2013, pp. 1–6.
- [470] B. L. Sturm, "A simple method to determine if a music information retrieval system is a "horse";" *IEEE Transactions on Multimedia*, vol. 16, no. 6, pp. 1636–1644, 2014.
- [471] B. L. Sturm, "The state of the art ten years after a state of the art: Future research in music information retrieval," *Journal of New Music Research*, vol. 43, no. 2, pp. 147–172, 2014.
- [472] B. L. Sturm, C. Kereliuk, and A. Pikrakis, "A closer look at deep learning neural networks with low-level spectral periodicity features," in *Proceedings of the 4th International Workshop on Cognitive Information Processing*, Copenhagen, Denmark, 2014, pp. 1–6.
- [473] B. L. Sturm and N. Collins, "The kiki-bouba challenge: Algorithmic composition for content-based MIR research & development," in *Proceedings of the 15th International Society for Music Information Retrieval Conference*, 2014, pp. 21–26.
- [474] B. L. Sturm, "A survey of evaluation in music genre recognition," in *Adaptive Multimedia Retrieval: Semantics, Context, and Adaptation*, A. Nürnberger, S. Stober, B. Larsen, and M. Detryniecki, Eds., vol. LNCS 8382, Springer International Publishing, Oct. 2014, pp. 29–66.
- [475] B. L. Sturm, C. Kereliuk, and J. Larsen, "¿El Caballo Viejo? Latin genre recognition with deep learning and spectral periodicity," in *Proceedings of the International Conference on Mathematics and Computation in Music*, 2015, pp. 335–346.
- [476] B. L. Sturm, "The "Horse"inside: Seeking causes behind the behaviors of music content analysis systems," *Computers in Entertainment*, vol. 14, no. 2, 2016.
- [477] B. L. Sturm, "Faults in the latin music database and with its use," in *Extended Abstracts for the Late-Breaking Demo Session of the 16th International Society for Music Information Retrieval Conference*, 2015.
- [478] B. L. Sturm, "Revisiting priorities: Improving MIR evaluation practices," in *Proceedings of the 17th International Society for Music Information Retrieval Conference*, New York City, United States: ISMIR, 2016, pp. 488–494.
- [479] L. Su, C.-C. M. Yeh, J.-Y. Liu, J.-C. Wang, and Y.-H. Yang, "A systematic evaluation of the bag-of-frames representation for music information retrieval," *IEEE Transactions on Multimedia*, vol. 16, no. 5, pp. 1188–1200, Aug. 2014.
- [480] M. Suero, C. P. Gassen, D. Mitic, N. Xiong, and M. Leon, "A deep neural network model for music genre recognition," in *Advances in Natural Computation, Fuzzy Systems and Knowledge Discovery*, Y. Liu, L. Wang, L. Zhao, and Z. Yu, Eds., ser. Advances in Intelligent Systems and Computing, Cham: Springer International Publishing, 2020, pp. 377–384.
- [481] S. Sugianto and S. Suyanto, "Voting-based music genre classification using melspectrogram and convolutional neural network," in *2019 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, Dec. 2019, pp. 330–333.
- [482] G. Sun, "Research on architecture for long-tailed genre computer intelligent classification with music information retrieval and deep learning," *Journal of Physics Conference Series*, vol. 2033, no. 1, p. 012 008, Sep. 2021.
- [483] E. N. Tamatjita and A. W. Mahastama, "Comparison of music genre classification using nearest centroid classifier and k-nearest neighbours," in *2016 International Conference on Information Management and Technology (ICIMTech)*, Nov. 2016, pp. 118–123.
- [484] C. P. Tang, K. L. Chui, Y. K. Yu, Z. Zeng, and K. H. Wong, "Music genre classification using a hierarchical long short term memory (LSTM) model," in *Third International Workshop on Pattern Recognition*, vol. 10828, SPIE, Jul. 26, 2018, pp. 334–340.

- [485] H. Tang, Y. Zhang, and Q. Zhang, “The Use of Deep Learning-Based Intelligent Music Signal Identification and Generation Technology in National Music Teaching,” *Frontiers in Psychology*, vol. 13, p. 762402, 2022. PMID: 35814087.
- [486] T. F. Tavares and J. H. Foleiss, “Automatic music genre classification in small and ethnic datasets,” in *International Symposium on Computer Music Multidisciplinary Research*, M. Aramaki, M. E. P. Davies, R. Kronland-Martinet, and S. Ystad, Eds., ser. Lecture Notes in Computer Science, Cham: Springer International Publishing, 2018, pp. 35–48.
- [487] R. Thiruvengatanadhan, “Music genre classification using mfcc and aann,” *International Research Journal of Engineering and Technology*, vol. 5, no. 10, pp. 1064–1066, 2018.
- [488] R. Thiruvengatanadhan, “Music genre classification using GMM,” *International Research Journal of Engineering and Technology*, vol. 5, no. 10, pp. 2395–0056, 2018.
- [489] N. Tokui. “Can GAN originate new electronic dance music genres? – Generating novel rhythm patterns using GAN with Genre Ambiguity Loss.” arXiv: 2011.13062 [cs]. (Nov. 25, 2020), [Online]. Available: <http://arxiv.org/abs/2011.13062> (visited on 03/24/2023), preprint.
- [490] T. Toshniwal, P. Tandon, and N. P., “Music genre recognition using short time fourier transform and CNN,” in *2022 International Conference on Computer Communication and Informatics (ICCCI)*, Jan. 2022, pp. 1–4.
- [491] A. Tsaptsinos, “Lyrics-Based Music Genre Classification Using a Hierarchical Attention Network,” in *Proceedings of the 18th International Society for Music Information Retrieval Conference*, Suzhou, China: ISMIR, Oct. 23, 2017, pp. 694–701.
- [492] S. Tsuchida, S. Fukayama, M. Hamasaki, and M. Goto, “AIST Dance Video Database: Multi-Genre, Multi-Dancer, and Multi-Camera Database for Dance Information Processing,” in *Proceedings of the 20th International Society for Music Information Retrieval Conference*, Delft, The Netherlands: ISMIR, Nov. 4, 2019, pp. 501–510.
- [493] G. Tzanetakis, “Chapter 26 - Music Mining,” in *Academic Press Library in Signal Processing: Volume 1 Signal Processing Theory and Machine Learning*, ser. Academic Press Library in Signal Processing: Volume 1, P. S. R. Diniz, J. A. K. Suykens, R. Chellappa, and S. Theodoridis, Eds., vol. 1, Elsevier, Jan. 1, 2014, pp. 1453–1492.
- [494] C. L. R. S. Ueno and D. Furtado Silva, “On combining diverse models for lyrics-based music genre classification,” in *2019 8th Brazilian Conference on Intelligent Systems (BRACIS)*, Oct. 2019, pp. 138–143.
- [495] A. S. Ulaganathan and S. Ramanna, “Granular methods in automatic music genre classification: A case study,” *Journal of Intelligent Information Systems*, vol. 52, no. 1, pp. 85–105, Feb. 1, 2019.
- [496] J. Urbano, D. Bogdanov, P. Herrera Boyer, E. Gómez Gutiérrez, and X. Serra, “What is the effect of audio quality on the robustness of MFCCs and chroma features?” In *Proceedings of the 15th International Society for Music Information Retrieval Conference*, International Society for Music Information Retrieval (ISMIR), 2014, pp. 573–578.
- [497] V. D. Valerio, R. M. Pereira, Y. M. Costa, D. Bertoini, and C. N. Silla Jr, “A resampling approach for imbalance on music genre classification using spectrograms,” in *The Thirty-First International Flairs Conference*, 2018.
- [498] J. Valverde-Rebaza, A. Soriano, L. Berton, M. C. Ferreira de Oliveira, and A. De Andrade Lopes, “Music Genre Classification Using Traditional and Relational Approaches,” in *2014 Brazilian Conference on Intelligent Systems*, Oct. 2014, pp. 259–264.
- [499] A. van den Oord, S. Dieleman, and B. Schrauwen, “Transfer learning by supervised pre-training for audio-based music classification,” in *Proceedings of the 15th International Society for Music Information Retrieval Conference*, 2014, pp. 29–34.
- [500] I. Vatulkin, M. Preuß, and G. Rudolph, “Training set reduction based on 2-gram feature statistics for music genre recognition,” Technische Universität Dortmund, Faculty of Computer Science, Algorithm ... , Algorithm Engineering Report TR13-2001, 2013.
- [501] I. Vatulkin, “Improving supervised music classification by means of multi-objective evolutionary feature selection,” Ph.D. dissertation, TU Dortmund, Jun. 20, 2013.
- [502] I. Vatulkin, G. Rötter, and C. Weihs, “Music Genre Prediction by Low-Level and High-Level Characteristics,” in *Data Analysis, Machine Learning and Knowledge Discovery*, ser. Studies in Classification, Data Analysis, and Knowledge Organization, M. Spiliopoulou, L. Schmidt-Thieme, and R. Janning, Eds., Cham: Springer International Publishing, 2014, pp. 427–434.
- [503] I. Vatulkin and G. Rudolph, “Interpretable music categorisation based on fuzzy rules and high-level audio features,” in *Data Science, Learning by Latent Structures, and Knowledge Discovery*, B. Lausen, S. Krolak-Schwerdt, and M. Böhmer, Eds., ser. Studies in Classification, Data Analysis,

- and Knowledge Organization, Berlin, Heidelberg: Springer, 2015, pp. 423–432.
- [504] I. Vatolkin, G. Rudolph, and C. Weihs, “Evaluation of album effect for feature selection in music genre recognition,” in *Proceedings of the 16th International Society for Music Information Retrieval Conference*, Málaga, Spain, 2015, pp. 169–175.
- [505] I. Vatolkin, G. Rudolph, and C. Weihs, “Interpretability of Music Classification as a Criterion for Evolutionary Multi-objective Feature Selection,” in *Evolutionary and Biologically Inspired Music, Sound, Art and Design*, C. Johnson, A. Carballal, and J. Correia, Eds., ser. Lecture Notes in Computer Science, Cham: Springer International Publishing, 2015, pp. 236–248.
- [506] I. Vatolkin and C. McKay, “Stability of Symbolic Feature Group Importance in the Context of Multi-Modal Music Classification,” in *Proceedings of the 23rd International Society for Music Information Retrieval Conference*, Bengaluru, India: ISMIR, Dec. 4, 2022, pp. 469–476.
- [507] I. Vatolkin and C. McKay, “Multi-Objective Investigation of Six Feature Source Types for Multi-Modal Music Classification,” *Transactions of the International Society for Music Information Retrieval*, vol. 5, no. 1, pp. 1–19, 1 Jan. 24, 2022.
- [508] T. G. Videira, B. Pennycook, and J. M. Rosa, “Formalizing fado: a contribution to automatic song-making,” *Journal of Creative Music Systems*, vol. 1, no. 2, 2 Mar. 1, 2017.
- [509] S. Vishnupriya and K. Meenakshi, “Automatic music genre classification using convolution neural network,” in *2018 International Conference on Computer Communication and Informatics (ICCCI)*, Jan. 2018, pp. 1–4.
- [510] L. Wadhwa and P. Mukherjee, “Music genre classification using multi-modal deep learning based fusion,” in *2021 Grace Hopper Celebration India (GHCI)*, Feb. 2021, pp. 1–5.
- [511] Z. Wang, J. Xia, and B. Luo, “The Analysis and Comparison of Vital Acoustic Features in Content-Based Classification of Music Genre,” in *2013 International Conference on Information Technology and Applications*, Nov. 2013, pp. 404–408.
- [512] Y. Wang, X. Chen, and P. J. Ramadge, “Sparse representation classification via sequential Lasso screening,” in *2013 IEEE Global Conference on Signal and Information Processing*, Dec. 2013, pp. 1001–1004.
- [513] J. Wang, C. Wang, J. Wei, and J. Dang, “Chinese opera genre classification based on multi-feature fusion and extreme learning machine,” in *2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (AP-SIPA)*, Dec. 2015, pp. 811–814.
- [514] Z. Wang, S. Muknahallipatna, M. Fan, A. Okray, and C. Lan, “Music Classification using an Improved CRNN with Multi-Directional Spatial Dependencies in Both Time and Frequency Dimensions,” in *2019 International Joint Conference on Neural Networks (IJCNN)*, Budapest, Hungary: IEEE, Jul. 2019, pp. 1–8.
- [515] L. Wang, H. Zhu, X. Zhang, S. Li, and W. Li, “Transfer learning for music classification and regression tasks using artist tags,” in *Proceedings of the 7th Conference on Sound and Music Technology (CSMT)*, H. Li, S. Li, L. Ma, C. Fang, and Y. Zhu, Eds., ser. Lecture Notes in Electrical Engineering, Singapore: Springer, 2020, pp. 81–89.
- [516] G. Wassi, S. Iloga, O. Romain, and B. Granado, “FPGA-based real-time MFCC extraction for automatic audio indexing on FM broadcast data,” in *2015 Conference on Design and Architectures for Signal and Image Processing (DASIP)*, Sep. 2015, pp. 1–6.
- [517] C. Weiss, M. Mauch, and S. Dixon, “Timbre-invariant audio features for style analysis of classical music,” in *11th Sound and Music Computing Conference and 40th International Computer Music Conference (SMC/ICMC2014)*, Athens, Greece: Zenodo, Sep. 14, 2014.
- [518] C. Weiß, “Computational methods for tonality-based style analysis of classical music audio recordings,” Ph.D. dissertation, Technische Universität Ilmenau, Aug. 25, 2017.
- [519] F. W. Wibowo and Wihayati, “Detection of indonesian dangdut music genre with foreign music genres through features classification using deep learning,” in *2021 International Seminar on Machine Learning, Optimization, and Data Science (ISMODE)*, Jan. 2022, pp. 313–318.
- [520] B. Wilkes, I. Vatolkin, and H. Müller, “Statistical and visual analysis of audio, text, and image features for multi-modal music genre recognition,” *Entropy*, vol. 23, no. 11, p. 1502, 11 Nov. 2021.
- [521] R. Wongso and D. D. Santika, “Automatic music genre classification using dual tree complex wavelet transform and support vector machine,” *Journal of Theoretical & Applied Information Technology*, vol. 63, no. 1, pp. 61–68, 2014.
- [522] H. Q. Wu and M. Zhang, “Gabor-LBP Features and Combined Classifiers for Music Genre Classification,” *Advanced Materials Research*, vol. 756–759, pp. 4407–4411, 2013.
- [523] M.-J. Wu and J.-S. R. Jang, “Combining acoustic and multilevel visual features for music genre classification,” *ACM Transactions on Multimedia Computing Communications and Applications*, vol. 12, no. 1, 10:1–10:17, Aug. 24, 2015.

- [524] M. Wu and Y. Wang, "A feature selection algorithm of music genre classification based on ReliefF and SFS," in *2015 IEEE/ACIS 14th International Conference on Computer and Information Science (ICIS)*, Jun. 2015, pp. 539–544.
- [525] W. Wu, F. Han, G. Song, and Z. Wang, "Music genre classification using independent recurrent neural network," in *2018 Chinese Automation Congress (CAC)*, Nov. 2018, pp. 192–195.
- [526] M. Wu and X. Liu, "A double weighted KNN algorithm and its application in the music genre classification," in *2019 6th International Conference on Dependable Systems and Their Applications (DSA)*, Jan. 2020, pp. 335–340.
- [527] B. D. Wundervald and W. M. Zeviani. "Machine learning and chord based feature engineering for genre prediction in popular brazilian music." arXiv: 1902.03283 [cs, eess, stat]. (Feb. 8, 2019), [Online]. Available: <http://arxiv.org/abs/1902.03283> (visited on 03/09/2023), preprint.
- [528] B. Wundervald, "Feature Engineering for Genre Characterization in Brazilian Music," in *Proceedings of the 13th International Workshop on Machine Learning and Music*, 2020, pp. 60–64.
- [529] Y. Xiao, Q. Zhang, M. Wu, and D. Kailing, "Application of multilevel local feature coding in music genre recognition," *Mathematical Problems in Engineering*, vol. 2022, e3627831, Mar. 22, 2022.
- [530] Y. Xu and W. Zhou, "A deep music genres classification model based on CNN with squeeze & excitation block," in *2020 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, Dec. 2020, pp. 332–338.
- [531] Z. Xu, "Construction of intelligent recognition and learning education platform of national music genre under deep learning," *Frontiers in Psychology*, vol. 13, p. 843427, May 26, 2022. pmid: 35693513.
- [532] K. Xu, M. A. Alif, and G. He, "A novel music genre classification algorithm based on continuous wavelet transform and convolution neural network," in *Proceedings of the 2021 5th International Conference on Electronic Information Technology and Computer Engineering*, ser. EITCE 2021, New York, NY, USA: Association for Computing Machinery, Dec. 31, 2022, pp. 1269–1273.
- [533] Z. Xu *et al.*, "Research on music genre classification based on residual network," in *Mobile Multimedia Communications*, Y. Chenggang, W. Honggang, and L. Yun, Eds., ser. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, Cham: Springer Nature Switzerland, 2022, pp. 209–223.
- [534] Y.-H. Yang, "Towards real-time music auto-tagging using sparse features," in *2013 IEEE International Conference on Multimedia and Expo (ICME)*, Jul. 2013, pp. 1–6.
- [535] J. Yang, "Lyric-based music genre classification," M.S. thesis, 2018.
- [536] H. Yang and W.-Q. Zhang, "Music genre classification using duplicated convolutional layers in neural networks.," in *Interspeech*, 2019, pp. 3382–3386.
- [537] R. Yang, L. Feng, H. Wang, J. Yao, and S. Luo, "Parallel recurrent convolutional neural networks-based music genre classification method for mobile devices," *IEEE Access*, vol. 8, pp. 19629–19637, 2020.
- [538] T. Ye, S. Si, J. Wang, N. Cheng, and J. Xiao, "Uncertainty Calibration for Deep Audio Classifiers," in *Interspeech*, Jun. 27, 2022. arXiv: 2206.13071 [cs, eess].
- [539] C.-C. M. Yeh, L. Su, and Y.-H. Yang, "Dual-layer bag-of-frames model for music genre classification," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2013, pp. 246–250.
- [540] Y. Yi, K.-Y. Chen, and H.-Y. Gu, "Mixture of CNN experts from multiple acoustic feature domain for music genre classification," in *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, Nov. 2019, pp. 1250–1255.
- [541] Y. Yi, X. Zhu, Y. Yue, and W. Wang, "Music genre classification with LSTM based on time and frequency domain features," in *2021 IEEE 6th International Conference on Computer and Communication Systems (ICCCS)*, Apr. 2021, pp. 678–682.
- [542] J. Yoon, H. Lim, and D.-W. Kim, "Music genre classification using feature subset search," *International Journal of Machine Learning and Computing*, vol. 6, no. 2, p. 134, 2016.
- [543] Y. Yu, S. Luo, S. Liu, H. Qiao, Y. Liu, and L. Feng, "Deep attention based music genre classification," *Neurocomputing*, vol. 372, pp. 84–91, Jan. 8, 2020.
- [544] C. Yuan *et al.*, "Exploiting heterogeneous artist and listener preference graph for music genre classification," in *Proceedings of the 28th ACM International Conference on Multimedia*, ser. MM '20, New York, NY, USA: Association for Computing Machinery, Oct. 12, 2020, pp. 3532–3540.
- [545] M. Zanoni, "Content-based macro-descriptors for music classification and multimedia information retrieval," Ph.D. dissertation, Politecnico Milano, Milan, Italy, Jan. 1, 2013.

- [546] C. Zhang, G. Evangelopoulos, S. Voinea, L. Rosasco, and T. Poggio, "A deep representation for invariance and music classification," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, pp. 6984–6988.
- [547] P. Zhang *et al.*, "A deep neural network for modeling music," in *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, ser. ICMR '15, New York, NY, USA: Association for Computing Machinery, Jun. 22, 2015, pp. 379–386.
- [548] W. Zhang, W. Lei, X. Xu, and X. Xing, "Improved music genre classification with convolutional neural networks," in *Interspeech*, ISCA, Sep. 8, 2016, pp. 3304–3308.
- [549] X. Zhang, T. Ren, L. Wang, and H. Xu. "Music Influence Modeling Based on Directed Network Model." arXiv: 2204.03588 [stat]. (Apr. 7, 2022), [Online]. Available: <http://arxiv.org/abs/2204.03588> (visited on 03/24/2023), preprint.
- [550] Y. Zhang, Z. Zhou, and M. Sun, "Influence of musical elements on the perception of 'chinese style' in music," *Cognitive Computation and Systems*, vol. 4, no. 2, pp. 147–164, 2022.
- [551] R. Zhang, X. Zhou, and J. Song, "Music and musician influence, similarity measure, and music genre division based on social network analysis," in *2nd International Conference on Artificial Intelligence, Automation, and High-Performance Computing (AIAHPC 2022)*, vol. 12348, SPIE, Nov. 10, 2022, pp. 107–116.
- [552] W. Zhang, "Music genre classification based on deep learning," *Mobile Information Systems*, vol. 2022, e2376888, Aug. 21, 2022.
- [553] Y. Zhao and J. Guo, "MusiCoder: A universal music-acoustic encoder based on transformer," in *MultiMedia Modeling*, J. Lokoč *et al.*, Eds., ser. Lecture Notes in Computer Science, Cham: Springer International Publishing, 2021, pp. 417–429.
- [554] H. Zhao, C. Zhang, B. Zhu, Z. Ma, and K. Zhang, "S3T: Self-supervised pre-training with swin transformer for music classification," in *2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2022, pp. 606–610.
- [555] E. Zheng, M. Moh, and T.-S. Moh, "Music genre classification: A N-gram based musicological approach," in *2017 IEEE 7th International Advance Computing Conference (IACC)*, Jan. 2017, pp. 671–677.
- [556] Z. Zheng, "The classification of music and art genres under the visual threshold of deep learning," *Computational Intelligence and Neuroscience*, vol. 2022, p. 4439738, May 18, 2022. pmid: 35634048.
- [557] H. Zhu, Y. Niu, D. Fu, and H. Wang, "MusicBERT: A self-supervised learning of music representation," in *Proceedings of the 29th ACM International Conference on Multimedia*, ser. MM '21, New York, NY, USA: Association for Computing Machinery, Oct. 17, 2021, pp. 3955–3963.
- [558] Y. Zhuang, Y. Chen, and J. Zheng, "Music genre classification with transformer classifier," in *Proceedings of the 2020 4th International Conference on Digital Signal Processing*, ser. IC DSP 2020, New York, NY, USA: Association for Computing Machinery, Sep. 10, 2020, pp. 155–159.
- [559] A. Zlatintsi and P. Maragos, "Comparison of different representations based on nonlinear features for music genre classification," in *2014 22nd European Signal Processing Conference*, Sep. 2014, pp. 1547–1551.
- [560] A. Zuhair and H. Hassani. "Comparing the accuracy of deep neural networks (DNN) and convolutional neural network (CNN) in music genre recognition (MGR): Experiments on kurkish music." arXiv: 2111.11063 [cs, eess]. (Nov. 22, 2021), [Online]. Available: <http://arxiv.org/abs/2111.11063> (visited on 03/03/2023), preprint.
- [561] B. L. Sturm, "Two systems for automatic music genre recognition: What are they really recognizing?" In *Proc. ACM MIRUM Workshop*, Nov. 2012, pp. 69–74.
- [562] B. L. Sturm, "Classification accuracy is not enough: On the evaluation of music genre recognition systems," *J. Intell. Info. Systems*, vol. 41, no. 3, pp. 371–406, 2013.
- [563] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Trans. Speech Audio Process.*, vol. 10, no. 5, pp. 293–302, Jul. 2002.
- [564] B. L. Sturm, "An analysis of the GTZAN music genre dataset," in *Proc. ACM MIRUM Workshop*, Nara, Japan, Nov. 2012, pp. 7–12.
- [565] F. R. Algarra and B. L. Sturm, "Re-evaluating the scattering transform," in *Proc. ISMIR (Late breaking demo)*, 2015.
- [566] J. Urbano, M. Schedl, and X. Serra, "Evaluation in music information retrieval," *Journal of Intelligent Information Systems*, vol. 41, no. 3, pp. 345–369, 2013.
- [567] A. Flexer, "Statistical evaluation of music information retrieval experiments," *J. New Music Research*, vol. 35, no. 2, pp. 113–120, 2006.

- [568] A. Flexer, “A closer look on artist filters for musical genre classification,” in *Proc. ISMIR*, Sep. 2007, pp. 341–344.
- [569] A. Flexer and D. Schnitzer, “Album and artist effects for audio similarity at the scale of the web,” in *Proc. SMC*, Jul. 2009, pp. 59–64.
- [570] A. Flexer, D. Schnitzer, M. Gasser, and T. Pohle, “Combining features reduces hubness in audio similarity,” in *Proc. Int. Symp. Music Info. Retrieval*, 2010, pp. 171–176.
- [571] E. Pampalk, A. Flexer, and G. Widmer, “Improvements of audio-based music similarity and genre classification,” in *Proc. Int. Soc. Music Info. Retrieval*, Sep. 2005, pp. 628–233.
- [572] G. Peeters, J. Urbano, and G. J. F. Jones, “Notes from the ISMIR 2012 late-breaking session on evaluation in music information retrieval,” in *Proc. ISMIR*, 2012.
- [573] M. Schedl, A. Flexer, and J. Urbano, “The neglected user in music information retrieval research,” *J. Intell. Info. Systems*, vol. 41, no. 3, pp. 523–539, 2013.
- [574] J. Urbano, B. McFee, J. S. Downie, and M. Schedl, “How significant is statistically significant? the case of audio music similarity and retrieval,” in *Proc. ISMIR*, 2012, pp. 181–186.
- [575] G. C. Bowker and S. L. Star, *Sorting Things out: Classification and Its Consequences*. The MIT Press, 1999.
- [576] D. Brackett, *Interpreting Popular Music*, First printing of this edition. Berkeley: University of California Press, Oct. 25, 2000, 280 pp.
- [577] D. Brackett, *Categorizing Sound: Genre and Twentieth-Century Popular Music*. University of California Press, Jul. 2016, 376 pp.
- [578] P. Coulangeon and I. Roharik, “Testing the “Omnivore/Univore” Hypothesis in a Cross-National Perspective. On the Social Meaning of Eclectism in Musical Tastes,” presented at the The Summer Meeting of the ISA RC28, UCLA, Aug. 19, 2005.
- [579] P. Coulangeon, “Social Stratification of Musical Tastes : Questioning the Cultural Legitimacy Model,” *Revue française de sociologie*, vol. 46, no. 5, pp. 123–154, 2005.
- [580] M. A. Coutinho and F. Miranda, “To describe genres:: Problems and strategies,” in *Genre in a Changing World*, C. Bazerman, A. Bonini, and D. Figueiredo, Eds., The WAC Clearinghouse, 2009, pp. 35–55.
- [581] P. DiMaggio, “Classification in art,” *American Sociological Review*, vol. 52, no. 4, pp. 440–455, 1987. JSTOR: 2095290.
- [582] E. Drott, “The End(s) of Genre,” *Journal of Music Theory*, vol. 57, no. 1, pp. 1–45, Apr. 1, 2013.
- [583] F. Fabbri, “A theory of musical genres: Two applications,” in *Proc. Int. Conf. Popular Music Studies*, 1980.
- [584] F. Fabbri, “Browsing musical spaces: Categories and the musical mind,” in *Proc. Int. Association for the Study of Popular Music*, 1999.
- [585] S. Frith, *Music For Pleasure: Essays on the Sociology of Pop*, First Edition. Cambridge: Polity Press, Sep. 22, 1988, 232 pp.
- [586] J. Frow, *Genre*. New York, NY, USA: Routledge, 2005.
- [587] J. C. Lena and R. A. Peterson, “Classification as culture: Types and trajectories of music genres,” *American Sociological Review*, vol. 73, no. 5, pp. 697–718, 2008. eprint: <https://doi.org/10.1177/000312240807300501>.
- [588] J. C. Lena, *Banding Together: How Communities Create Genres in Popular Music*. Princeton University Press, 2012. JSTOR: j.ctt7rrzb.
- [589] O. Lizardo, “The mutual specification of genres and audiences: Reflective two-mode centralities in person-to-culture data,” *Poetics*, vol. 68, pp. 52–71, Jun. 1, 2018.
- [590] K. McLeod, “Genres, Subgenres, Sub-Subgenres and More: Musical and Social Differentiation Within Electronic/Dance Music Communities,” *Journal of Popular Music Studies*, vol. 13, no. 1, pp. 59–75, 2001.
- [591] L. B. Meyer, *Emotion and Meaning in Music*. Chicago, IL: University of Chicago Press, Feb. 1961, 315 pp.
- [592] A. F. Moore, “Categorical conventions in music discourse: Style and genre,” *Music & Letters*, vol. 82, no. 3, pp. 432–442, 2001.
- [593] K. Negus, *Producing Pop: Culture and Conflict in the Popular Music Industry*. London: Hodder Education, Jan. 7, 1993, 192 pp.
- [594] K. Negus, *Popular Music in Theory: An Introduction*, 1st edition. Cambridge: Polity, Nov. 30, 1996, 255 pp.
- [595] K. Negus, “From creator to data: The post-record music industry and the digital conglomerates,” *Media, Culture & Society*, vol. 41, no. 3, pp. 367–384, Apr. 2019.
- [596] W. G. Roy, ““Race records” and “hillbilly music” : Institutional origins of racial categories in the American commercial recording industry,” *Poetics, Music in Society: The Sociological Agenda*, vol. 32, no. 3, pp. 265–279, Jun. 1, 2004.
- [597] W. G. Roy and T. J. Dowd, “What Is Sociological about Music?” *Annual Review of Sociology*, vol. 36, pp. 183–203, Volume 36, 2010 Aug. 11, 2010.

- [598] P. Tagg, “Analysing popular music: Theory, method and practice,” *Popular Music*, vol. 2, pp. 37–67, 1982. JSTOR: 852975.
- [599] N. Scaringella, G. Zoia, and D. Mlynek, “Automatic genre classification of music content: A survey,” *IEEE Signal Process. Mag.*, vol. 23, no. 2, pp. 133–141, Mar. 2006.
- [600] T. Bertin-Mahieux, D. P. Ellis, B. Whitman, and P. Lamere, “The million song dataset,” in *Proc. ISMIR*, 2011, pp. 591–596.
- [601] F. Rodríguez-Algarra, B. L. Sturm, and S. Dixon, “Characterising confounding effects in music classification experiments through interventions,” *Trans. Int. Soc. Music Information Retrieval*, vol. 2, no. 1, pp. 52–66, 2019.
- [602] S. Oramas, F. Barieri, O. Nieto, and X. Serra, “Multimodal deep learning for music genre classification,” *Trans. ISMIR*, 2018.
- [603] B. Matityaho and M. Furst, “Neural network based model for classification of music type,” in *Proc. Conv. Electrical and Elect. Eng. in Israel*, Mar. 1995, pp. 1–5.
- [604] J.-J. Aucouturier and E. Pampalk, “Introduction – from genres to tags: A little epistemology of music information retrieval research,” *J. New Music Research*, vol. 37, no. 2, pp. 87–92, 2008.
- [605] K. Byun and M. Y. Kim, “Musical Genre Classification System based on Multiple-Octave Bands,” *Journal of the Institute of Electronics and Information Engineers*, vol. 50, no. 12, pp. 238–244, 2013.
- [606] H. C. Chang and C. K. Yang, “Automatic MIDI Genre Conversion,” *Applied Mechanics and Materials*, vol. 284–287, pp. 3040–3043, 2013.
- [607] C.-H. Chuan, “Audio Classification and Retrieval Using Wavelets and Gaussian Mixture Models,” *International Journal of Multimedia Data Engineering and Management (IJMDEM)*, vol. 4, no. 1, pp. 1–20, Jan. 2013.
- [608] B. Kostek and A. Kaczmarek, “Music Recommendation Based on Multidimensional Description and Similarity Measures,” *Fundamenta Informaticae*, vol. 127, no. 1-4, pp. 325–340, Jan. 2013.
- [609] T. F. B. M. de Matos, “Métodos Estatísticos de Classificação de Géneros Musicais,” M.S. thesis, Lisbon, Jul. 2013.
- [610] P. Hoffmann and B. Kostek, “Subjective perception of music genres in the field of music information retrieval systems,” in *15th International Symposium on New Trends in Audio and Video*, 2014.
- [611] D. Jang, S. Shin, J. Lee, S.-J. Jang, and T.-B. Lim, “Feature reduction based on distance metric learning for musical genre classification,” in *Proceedings of the Korean Society of Broadcast Engineers Conference*, The Korean Institute of Broadcast and Media Engineers, 2014, pp. 3–4.
- [612] R. Miki, W. Kameyama, and M. Suganuma, “A Consideration on Music Classification using Background Activity of Brain,” *IEICE Technical Report*, vol. 114, no. 68, pp. 211–216, May 2014.
- [613] B. K. Baniya and 이준환, “Label prediction of the unlabeled mood of a music genre using semi-supervised learning,” *차세대컨버전스정보서비스기술논문지*, vol. 4, no. 2, pp. 51–64, 2015.
- [614] P. Hoffmann and B. Kostek, “Music genre classification applied to bass enhancement for mobile technology,” *Elektronika : konstrukcje, technologie, zastosowania*, vol. 56, no. 4, pp. 14–19, 2015.
- [615] K. C. Silva Paulo, R. D. Solgon Bassi, A. L. Delorme, R. C. [Guido, I. N. da Silva, and Anonymous, “Music genre classification based on para-consistency,” in *2nd International Conference On Advanced Education Technology And Management Science (Aetms 2014)*, Destech Publications, Inc, Jan. 2015, p. 427.
- [616] S. U. N. Hui, X. U. Jieping, and L. I. U. Binbin, “Music genre classification based on multiple kernel learning and support vector machine,” *Journal of Computer Applications*, vol. 35, no. 6, p. 1753, Jun. 2015.
- [617] S. Kim, D. Kim, and B. Suh, “Music genre classification using multimodal deep learning,” in *Proceedings of HCI Korea*, ser. HCIK ’16, Seoul, KOR: Hanbit Media, Inc., Jan. 2016, pp. 389–395.
- [618] Ö. Çoban and G. T. Özzyer, “Music genre classification from turkish lyrics,” in *2016 24th Signal Processing and Communication Application Conference (SIU)*, May 2016, pp. 101–104.
- [619] R. H. D. Zottesso, Y. M. G. Costa, and D. Bertolini, “Music genre classification using visual features with feature selection,” in *2016 35th International Conference of the Chilean Computer Science Society (SCCC)*, Oct. 2016, pp. 1–6.
- [620] W.-J. Jang, H.-W. Yun, S.-H. Shin, H.-J. Cho, W. Jang, and H. Park, “Music genre classification using spikegram and deep neural network,” *Journal of Broadcast Engineering*, vol. 22, no. 6, pp. 693–701, 2017.
- [621] K. Açıcı, T. Aşuroğlu, and H. Oğul, “Information retrieval in metal music sub-genres,” in *2017 25th Signal Processing and Communications Applications Conference (SIU)*, May 2017, pp. 1–4.
- [622] Ö. Çoban and I. Karabey, “Music genre classification with word and document vectors,” in *2017 25th Signal Processing and Communications Applications Conference (SIU)*, May 2017, pp. 1–4.
- [623] A. Karatana and O. Yildiz, “Music genre classification with machine learning techniques,” in *2017 25th Signal Processing and Communications Applications Conference (SIU)*, May 2017, pp. 1–4.

- [624] A. Azcarraga and F. K. Flores, “A study on self-organizing maps and K-means clustering on a music genre dataset,” in *Theory and Practice of Computation*, World Scientific, Oct. 2017, pp. 219–234.
- [625] V. S. González, “The impact of temporal features in music genre recognition,” M.S. thesis, TU Dublin, Jan. 2019.
- [626] Y. Hao, “Music genre classification and transfer based on MusicXML with high-level features and chord vectors,” M.S. thesis, ResearchSpace@Auckland, 2019.
- [627] C. Dabas, A. Agarwal, N. Gupta, V. Jain, and S. Pathak, “Machine learning evaluation for music genre classification of audio signals,” *International Journal of Grid and High Performance Computing (IJGHPC)*, vol. 12, no. 3, pp. 57–67, 2020.
- [628] M. Anand, V. Vijayalakshmi, and S. Vimal, “Music genre classification with deep learning,” *Solid State Technology*, vol. 63, no. 6, pp. 14 730–14 734, Dec. 2020.
- [629] A. Shreyash, S. P. Dhanure, P. P. Rathod, C. Ashay, and G. Pritesh, “Identification of music genre using convolutional neural network,” *NEW ARCH-INTERNATIONAL JOURNAL OF CONTEMPORARY ARCHITECTURE*, vol. 8, no. 2, pp. 2103–2109, Oct. 2021.
- [630] A. K. Mishra, D. K. Singh, and A. Khare, “Music genre detection using deep learning models,” *i-Manager’s Journal on Information Technology*, vol. 11, no. 2, p. 10, 2022.
- [631] J. Singh, “An efficient deep neural network model for music classification,” *International Journal of Web Science*, vol. 3, no. 3, pp. 236–248, Jan. 2022.
- [632] D. V. Subbaiah, N. N. Jyothi, K. Lokesh, K. S. Anusha, and K. Saikumar, “Instinctive music genre detection and categorization of audio data using machine learning,” *Harbin Gongye Daxue Xuebao/Journal of Harbin Institute of Technology*, vol. 54, no. 4, pp. 354–356, 2022.
- [633] A. Porter, D. Bogdanov, R. Kaye, R. Tsukanov, and X. Serra, “Acousticbrainz: A community platform for gathering music information obtained from audio,” in *Proc. ISMIR*, 2015, pp. 786–792.
- [634] P. Cano *et al.*, “Ismir 2004 audio description contest,” Barcelona: Universitat Pompeu Fabra, Music technology Group, Tech. Rep. MTG-TR-2006-02, 2006.
- [635] K. Benzi, M. Defferrard, P. Vandergheynst, and X. Bresson, “FMA: A dataset for music analysis,” *arXiv*, vol. 1612.01840, 2016.
- [636] C. N. Silla, A. L. Koerich, and C. A. A. Kaestner, “The Latin music database,” in *Proc. ISMIR*, 2008, pp. 451–456.
- [637] S. Dixon, F. Gouyon, and G. Widmer, “Towards characterisation of music via rhythmic patterns,” in *Proc. ISMIR*, 2004, pp. 509–517.
- [638] B. L. T. Sturm and A. Flexer, “A review of validity and its relationship to music information research,” in *Proc. Int. Symp. Music Info. Retrieval*, 2023.
- [639] R. Huang, A. Holzapfel, B. L. T. Sturm, and A.-K. Kaila, “Beyond diverse datasets: Responsible mir, interdisciplinarity, and the fractured worlds of music,” *Trans. Int. Soc. Music Information Retrieval*, vol. 6, no. 1, pp. 43–59, 2023.
- [640] J. Priem, H. Piwowar, and R. Orr, *Openalex: A fully-open index of scholarly works, authors, venues, institutions, and concepts*, 2022. arXiv: 2205.01833 [cs.DL].
- [641] J. Rodu and M. Baiocchi, “When black box algorithms are (not) appropriate,” *Observational Studies*, vol. 9, no. 2, pp. 79–101, 2023.
- [642] D. Raji, E. Denton, E. M. Bender, A. Hanna, and A. Paullada, “AI and the Everything in the Whole Wide World Benchmark,” *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, vol. 1, Dec. 6, 2021.
- [643] R. McElreath, *Statistical Rethinking: A Bayesian Course with Examples in R and Stan*. CRC press, 2020.
- [644] A. Ferraro, G. Ferreira, F. Diaz, and G. Born, “Measuring Commonality in Recommendation of Cultural Content: Recommender Systems to Enhance Cultural Citizenship,” in *Proceedings of the 16th ACM Conference on Recommender Systems*, ser. RecSys ’22, New York, NY, USA: Association for Computing Machinery, Sep. 13, 2022, pp. 567–572.
- [645] A. Holzapfel, B. L. Sturm, and M. Coeckelbergh, “Ethical dimensions of music information retrieval technology,” *Trans. Int. Soc. Music Information Retrieval*, vol. 1, no. 1, pp. 44–55, 2018.
- [646] M. Youngblood, K. Baraghith, and P. E. Savage, “Phylogenetic reconstruction of the cultural evolution of electronic music via dynamic community detection (1975–1999),” *Evolution and Human Behavior*, vol. 42, no. 6, pp. 573–582, Nov. 1, 2021.
- [647] J. A. Hockman, “An Ethnographic and Technological Study of Breakbeats in Hardcore, Jungle and Drum & Bass,” Ph.D. dissertation, McGill University (Canada), Canada – Quebec, CA, 2013, 544 pp.
- [648] A. Srinivasamurthy, A. Holzapfel, K. K. Ganguli, and X. Serra, “Aspects of Tempo and Rhythmic Elaboration in Hindustani Music: A Corpus Study,” *Frontiers in Digital Humanities*, vol. 4, Oct. 31, 2017.
- [649] K. Frieler, “A feature history of jazz improvisation,” in *Jazz @ 100*, ser. Darmstadt Studies in Jazz Research, W. Knauer, Ed., vol. 15, Hofheim: Wolke Verlag, 2018, pp. 67–90.

- [650] W. R. Shadish, T. D. Cook, and D. T. Campbell, *Experimental and Quasi-experimental Designs for Generalised Causal Inference*. Houghton Mifflin Company, 2001.

CONTENT-BASED CONTROLS FOR MUSIC LARGE LANGUAGE MODELING

Liwei Lin^{1,2} Gus Xia^{1,2} Junyan Jiang^{1,2} Yixiao Zhang³

¹ Music X Lab, New York University Shanghai

² Mohamed bin Zayed University of Artificial Intelligence

³ C4DM, Queen Mary University of London

ABSTRACT

Recent years have witnessed a rapid growth of large-scale language models in the domain of music audio. Such models enable end-to-end generation of higher-quality music, and some allow conditioned generation using text descriptions. However, the control power of text controls on music is intrinsically limited, as they can only describe music *indirectly* through meta-data (such as singers and instruments) or high-level representations (such as genre and emotion). *We aim to further equip the models with direct and content-based controls on innate music languages* such as pitch, chords and drum track. To this end, we contribute *Coco-Mulla*, a **content-based control** method for **music large language modeling**. It uses a parameter-efficient fine-tuning (PEFT) method tailored for Transformer-based audio models. Experiments show that our approach achieves high-quality music generation with **low-resource** semi-supervised learning. We fine-tune the model with less than 4% of the original parameters on a small dataset with fewer than 300 songs. Moreover, our approach enables effective content-based controls. We illustrate its controllability via chord and rhythm conditions, two of the most salient features of pop music. Furthermore, we show that by combining content-based controls and text descriptions, our system achieves flexible music variation generation and arrangement. Our source codes and demos are available online ^{1 2}.

1. INTRODUCTION

Controllable music generation encompasses the creation of music under various controls, such as musical or textual descriptions [1–4]. It enables amateur users to create customized music and helps professional musicians explore new ideas for composition and arrangement. Re-

markable advancements have been made in this field in recent years, particularly in text-to-music generation [3–6]. These cross-modality models are trained on extensive sets of parallel text-audio data pairs, utilizing pre-trained large language models [7] or multi-modal embeddings [8, 9] to establish a mapping between natural language and music. They enable high-level controls such as mood and tempo by incorporating them into a text prompt.

However, not all music information can be expressed via text. Existing models cannot yet apply effective controls on intricate musical languages (e.g., chord progressions) or directly refer to musical contents from other audio recordings. The ability to accommodate such *content-based* controls is crucial for tasks such as music editing, music variation generation, and arrangement. For example, in AI-assistant composition systems, the generative models are required to compose based on a rough idea like motifs or counter-melodies; in music re-instrumentation, we tend to keep the underlying harmony unchanged and reinterpret the song with new instruments.

Historically speaking, the shift from text-based (and metadata-based) models to content-based models has once happened in the realm of music information retrieval (MIR), which dramatically improved the model performances and also expanded the boundary of music understanding. In a similar fashion, we aim to push the boundary of music audio generation domain by further equipping off-shelf generative models with content-based controls.

A simple approach to incorporating content-based control is to train a separate generative model conditioned on the provided control content [4, 10]. For example, both the MusicGen and MusicLM systems offer two versions of model: a vanilla text-to-music version, and a melody-conditioned version for accompaniment generation. The main issue with this simple conditioning approach lies in the high training cost, which is at the same scale as the base model in terms of both computational resources and training data. Moreover, each conditioned model can only deal with *one type* of content-based control input. This rigid setting is not practical in real music production and arrangement scenarios where multiple control contents are often required for satisfactory results.

To solve the aforementioned problems, we contribute a unified approach to incorporating different content-based controls with music-audio generative models. Particularly,

¹ <https://github.com/Kikyo-16/coco-mulla-repo>.

² <https://kikyo-16.github.io/coco-mulla/>.



we see the immense potential of large-scale pre-trained models in their semantic-level understanding of music and therefore propose a novel parameter-efficient fine-tuning condition adaptor based on llama adaptor [11].

The current design of the adaptor integrates the joint embeddings of symbolic music chords and piano roll and acoustic drum tracks with the pre-trained MusicGen model. In theory, the adaptor can perform on joint embeddings of any combination of content-based controls and can be integrated with any Transformer-based generative model. In short, the main contribution of this work is as follows:

- **A unified approach on content-based controls:** Our model enables chord and drum pattern controls via acoustic hints, achieving an arbitrary combination of textual, harmonic, and rhythmic description for the controlled generation process.
- **Low-resource fine-tuning on pseudo-labeled datasets:** We provide a method to fine-tune a large auto-regressive audio generative model with a small-size, pseudo-labeled dataset in which all the pseudo labels are extracted using existing MIR tools. We fine-tune MusicGen [4], an excellent text-to-music model, on 4% trainable parameters of the original model with a training set of fewer than 300 songs without text or other annotations.
- **Flexible variation generation and arrangement:** Our model achieves flexible variation generation and arrangement of the given polyphonic piano roll by combining text prompts and content-based controls. This enables numerous downstream music-editing applications.

2. RELATED WORK

We review two realms of related works: 1) large-language models (LLMs) for music audio and 2) existing methods of parameter-efficient fine-tuning.

2.1 Music Audio Generation

Music audio generation necessitates extensive contextual modeling to account for the intricate structure of musical language. Recent large-scale music audio generative models, encompassing auto-regressive and diffusion-based approaches, have made remarkable strides in capturing such a long-term structure while introducing cross-modality conditions. For example, Jukebox [1] leverages VQ-VAE [12] and transformer decoders to achieve lyrics- and genre-based generation; Diffusion-based Moûsai [3] adopts the pre-trained frozen T5 encoder [7] to summarize text conditions; auto-regressive MusicGen [4] realizes monophonic melody and text controls by assembling EnCodec [13], T5 encoder, and an acoustic transformer decoder. Specifically, MusicGen is the first text- and melody-conditioned model, limited to a monophonic melody condition, and it does not accommodate drum tracks. Additionally, a contemporaneous work, Music ControNet [14], shares a similar goal

with ours but is based on a diffusion model rather than on pretrained large language models (LLMs).

2.2 Parameter-Efficient Fine-Tuning

Parameter-efficient fine-tuning (PEFT) methods adapt pre-trained language models (PLMs) without fine-tuning all model parameters [15, 16], significantly reducing computational and storage expenses. [17] and [18] tailor PLMs for specific tasks by appending task-specific prefixes to input sequences. [19] employs low-rank adaption (LoRA) to fine-tune pre-trained linear transformations in PLMs. [11] and [20] propose LLaMA-Adapter, adjusting attention outputs using prompt adaptors and zero gate scalars. LLaMA-Adapter also introduces a multi-modal conditional variant by incorporating *global* image representations into prompt adaptors. In this study, we present a novel PEFT method, inspired by LLaMA-Adapter, designed to fine-tune large-scale models while accommodating external *sequential* multi-modal conditions.

3. BASE MODEL

In this work, we choose MusicGen [4], an excellent Transformer-based music audio language model, as our base model. MusicGen offers two variants: a text-only model and a melody-based model. The melody-based model conditions its generation on the dominant time-frequency bin of the audio chromagram and text prompts, limiting it to monophonic conditioning and preventing it from incorporating rhythmic drum patterns. In this study, we adopt the text-only MusicGen as our base model, augmenting it with content-based controls via the proposed PEFT method.

The largest text-only MusicGen consists of 3 components: a pre-trained EnCodec, a pre-trained T5 encoder, and an acoustic transformer decoder. The transformer decoder comprises $N = 48$ layers, each including a causal self-attention block and a cross-attention block to handle condition text prompts.

MusicGen tokenizes audio signals using EnCodec [13], a Residual Vector Quantization (RVQ) [21] auto-encoder, compressing signals of sample rate $S_r = 32000$ into discrete codes of a low frame rate $f_s = 50$. The acoustic transformer decoder takes these tokens as its input.

4. METHODOLOGY

Our approach consists of 2 components: 1) a joint embedding encoder to integrate content-based controls, and 2) a condition adaptor to fine-tune MusicGen by incorporating the learned joint embeddings. To maintain the ability of the vanilla MusicGen to associate text with music, we train the adaptor for only the self-attention blocks of the acoustic transformer decoder. During training, all the parameters of MusicGen are frozen.

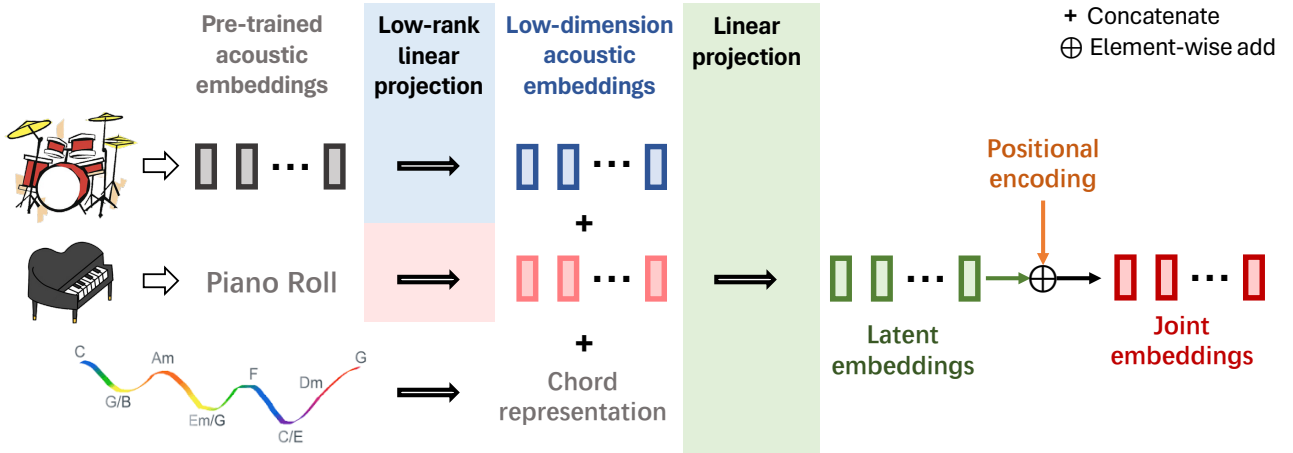


Figure 1. The joint embedding module. We randomly mask acoustic or piano roll embedding with probability r during training.

Chord Name	Pitch	Root	Bass	Pitch Indices
C:maj	C, E, G	C (0)	C (0)	0, 4, 7
C:maj/E	C, E, G	C (0)	E (4)	0, 4, 7
D:min	D, F, A	D (2)	D (2)	0, 3, 7

Table 1. Symbolic chord data structure.

4.1 Joint Symbolic and Acoustic Embedding

To incorporate the desired chord progression and to borrow musical content from another audio recording simultaneously, we design a joint symbolic and acoustic embedding. For precise alignment of generated music with acoustic hints, we use a frame-wise representation for both symbolic and acoustic data.

4.1.1 Symbolic Chord and MIDI Representation

We describe a chord as a combination of a root pitch class, the bass pitch class, and the chroma representation of the chord quality. As shown in Table 1, we represent a chord as $\{\text{root}, \text{bass}, \mathbf{m}\}$ (root, bass $\in \{0, 1, \dots, 11\}$), with $\mathbf{m} \in \mathbb{R}^{12}$ being a multi-hot positional vector indicating the active pitches in the octave starting from the root note. Define $\mathbf{c}_i \in \mathbb{R}^{12+12+12+1}$ to represent the i^{th} -frame chord:

$$\mathbf{c}_i = \begin{cases} [e(\text{root}); e(\text{bass}); \mathbf{m}; 0], & \text{if } i^{\text{th}} \text{ frame has a chord} \\ [0; 0; 0; 1], & \text{otherwise} \end{cases}, \quad (1)$$

where e is a function from an index $j \in \{0, 1, \dots, 11\}$ to its one-hot vector $e(j) \in \mathbb{R}^{12}$.

We represent MIDI using the piano roll format. Assume $\mathbf{p}_i \in \{0, 1\}^{128}$ is the i^{th} frame in the piano roll indicating the presence of each pitch. We further compress the sparse piano roll into a low-dimension MIDI representation \mathbf{p}'_i using a trainable matrix $\mathbf{W}_p \in \mathbb{R}^{d_1 \times 128}$:

$$\mathbf{p}'_i = \mathbf{W}_p^T \mathbf{p}_i \in \mathbb{R}^{d_1}. \quad (2)$$

Throughout the training process, we use pseudo chord annotations obtained through a chord recognition model from [22] and MIDI annotations via an automatic music transcription model from [23].

4.1.2 Acoustic Representation

We convert the separated drum stem to discrete codes using EnCodec [13]. Instead of directly modeling these discrete codes, we pass the i^{th} -frame codes through the frozen input embedding layer of MusicGen transformer decoder to obtain a pre-trained acoustic embedding $\mathbf{h}_i \in \mathbb{R}^{2048}$. Such a continuous pre-trained representation is more robust since it can address the issue of utilizing discrete codes not present in the training data during the inference stage.

To mitigate overfitting and reduce training complexity, we employ a trainable low-rank matrix $\mathbf{W}_a \in \mathbb{R}^{2048 \times d_2}$ to map \mathbf{h}_i into a lower-dimensional space:

$$\mathbf{h}'_i = \mathbf{W}_a^T \mathbf{h}_i \in \mathbb{R}^{d_2}. \quad (3)$$

We set $d_2 = d_1 = 12$ in our experiments.

4.1.3 Masking Scheme and Positional Encoding

During training, we randomly mask MIDI and acoustic representation with a probability r independently:

$$\mathbf{z}_i^p = \begin{cases} \mathbf{p}'_i, & \text{if not masked} \\ \mathbf{s}_i^p, & \text{otherwise} \end{cases}, \quad (4)$$

$$\mathbf{z}_i^a = \begin{cases} \mathbf{h}'_i, & \text{if not masked} \\ \mathbf{s}_i^a, & \text{otherwise} \end{cases}. \quad (5)$$

Here, \mathbf{s}_i^p and \mathbf{s}_i^a are learnable masked embeddings. The masking strategy trains the model to follow an arbitrary combination of conditional tracks during inference. We set $r = 0.4$ in our experiments.

We introduce a learnable matrix $\mathbf{W}_e \in \mathbb{R}^{(d_1+d_2+37) \times d}$ to incorporate the above embeddings and a learnable positional embedding $\mathbf{z}_i^{\text{pos}} \in \mathbb{R}^{d_1+d_2+37}$ to facilitate sequen-

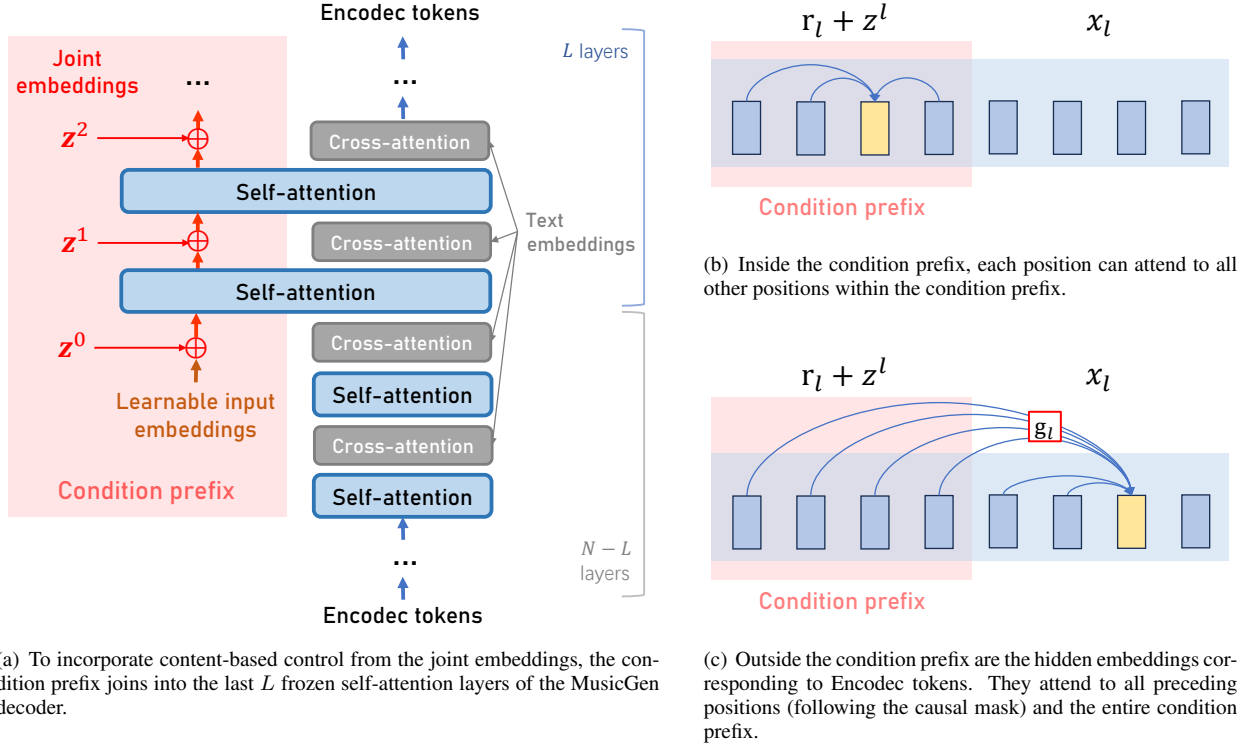


Figure 2. Condition adaptor. The condition prefix is injected to the self-attention mechanism of the MusicGen transformer decoder. All transformation matrices in MusicGen are frozen. Only the input embeddings, joint embedding encoders, and the gate factors are trainable.

tial modeling. The final joint symbolic and acoustic embedding is as follows:

$$z_i = \mathbf{W}_e^T([c_i; z_i^p; z_i^a] + z_i^{\text{pos}}) \in \mathbb{R}^d. \quad (6)$$

Finally, let T be the total number of frames. The complete sequential joint embedding is then:

$$z = \{z_1, z_2, \dots, z_T\} \in \mathbb{R}^{T \times d}. \quad (7)$$

4.2 Condition Adaptor

To plug the joint embeddings into MusicGen, we present a novel condition adaptor that can take time-varying sequential conditions. In the vanilla Transformer, each self-attention layer operates on a sequence of T hidden embeddings corresponding to T frames of Encodec tokens. In the proposed condition adaptor, as shown in Fig 2, for the last L layers of the MusicGen decoder, we expand the sequence of hidden embeddings to $2T$, where T new positions take on the task of incorporating and processing condition-related information. We call the newly introduced positions the *condition prefix*.

Specifically, we insert a sequence of learnable input embeddings into the $(N - L + 1)^{\text{th}}$ MusicGen transformer decoder layer, initiating the condition prefix. Inside the condition prefix, we pass the hidden states only through self-attention layers, skipping cross-attention layers.

Let $\mathbf{H}_l^p \in \mathbb{R}^{T \times d}$ ($N - L + 1 \leq l \leq N$) represent the output of the l^{th} -layer attention layer for the condition

prefix. \mathbf{H}_0^p is a sequence of learnable input embeddings. We compute the condition prefix as follows:

$$\mathbf{Q}_l^p, \mathbf{K}_l^p, \mathbf{V}_l^p = \text{QKV-projector}(\mathbf{H}_l^p + \mathbf{Z}_l), \quad (8)$$

$$\mathbf{H}_{l+1}^p = \text{Self-Attention}(\mathbf{Q}_l^p, \mathbf{K}_l^p, \mathbf{V}_l^p), \quad (9)$$

where the sequential joint embeddings \mathbf{Z}_l is defined in Eq (7). Note that we learn distinct joint embeddings for each decoder layer. Since the adaptor aims at capturing the long-term contextual information of the sequential joint embeddings, it *does not* employ a causal attention mask for the condition prefix. Additionally, the condition prefix does not attend to the Encodec token frames, as shown in Fig 2(b).

For the non-prefix part, hidden states are passed through both self-attention and cross-attention layers. Let $\mathbf{H}_l \in \mathbb{R}^{T \times d}$ ($1 \leq l \leq N$) represent the output of the l^{th} attention layer for the encoded tokens. we compute vanilla attention output \mathbf{S}_l as follows:

$$\mathbf{Q}_l, \mathbf{K}_l, \mathbf{V}_l = \text{QKV-projector}(\mathbf{H}_l), \quad (10)$$

$$\mathbf{S}_l = \text{Self-Attention}(\mathbf{Q}_l, \mathbf{K}_l, \mathbf{V}_l). \quad (11)$$

To incorporate condition information, in the last L layers, we compute cross attention \mathbf{S}'_l between \mathbf{Q}_l and $\{\mathbf{K}_l^p, \mathbf{V}_l^p\}$. We leverage Self-Attention layers to compute \mathbf{S}'_l rather than Cross-Attention layers since the controls are closer to the audio modality than the textual modality. To

	Chord _{rec} ↑	Chord _{rec} [*] ↑	Beat _{F₁} ↑	CLAP _{scr} ↑	FAD _{vgg} [*] ↓	FAD _{vgg} ↓
Chord-only	0.412	0.195	-	0.401	6.209	6.695
MIDI-only	0.649	0.406	-	0.381	7.105	7.094
Drums-only	0.530	0.267	0.856	0.360	3.845	4.933
Full	0.791	0.524	0.864	0.351	3.697	4.370
MusicGen	-	-	-	0.441	6.434	6.847
Oracle	0.885	0.695	0.898	-	-	-

Table 2. The performance of the model with $L = 48$ on RWC-POP-100 subset. Oracle scores gauge the performance of the chord recognition model [22] and beat tracking model [24] on the ground-truth audio.

L	Total	Trainable	CLAP _{scr} [*] ↑		Chord _{rec} ↑	
			Chord-only	Full	Chord-only	Full
12	3.29B	0.87%	0.428	0.371	0.239	0.672
24	3.31B	1.66%	0.408	0.358	0.397	0.747
36	3.33B	2.44%	0.396	0.344	0.410	0.772
48	3.36B	3.20%	0.401	0.351	0.412	0.791

Table 3. The performance of models with different L values under the chord-only condition.

make {query, key, value} more compatible, we use the fusion of Q_l and Q_l^p as the query instead of a single Q_l :

$$S'_l = \text{Self-Attention}(Q_l + Q_l^p, K_l^p, V_l^p). \quad (12)$$

Following this, we combine S_l and S'_l using a zero-initialized learnable gating factor g_l . Additionally, to maintain the text controllability of the model, we then compute the cross attention between textual embedding and them:

$$H_{l+1} = \text{Cross-Attention}(S_l + g_l \cdot S'_l, \text{text}). \quad (13)$$

All the layers in MusicGen are frozen, including the QKV-projector, Self-Attention, and Cross-Attention layers. Hence, the total trainable parameters only comprise H_0^p , W_p , W_a , W_e , z^{pos} , and g_l . Moreover, the proposed adaptor can learn in a semi-supervised manner using pseudo-separated tracks, pseudo MIDI, and pseudo chord labels. Additionally, during training, each music piece is assigned to a vague text description randomly sampled from a small set of predefined phrases, eliminating the requirement for text-audio data pairs.

5. EXPERIMENT

5.1 Datasets

The training dataset consists of 299 *unannotated instrumental* songs. We collect 150 of them from an open-source dataset MUSDB18 [25] and download the remain 149 songs from the internet. The latter subset predominantly consists of Pop songs, with a limited data of other genres such as Jazz and Rock. We omit the silent start and end segments of each training song, yielding 17.12 hours of audio. We employ Demucs to extract drum tracks, a chord recognition model [22], an automatic transcription

model [26], and a beat tracking model [24] to generate pseudo chord, MIDI, and beat labels.

The test set comprises 50 songs with chord, beat, and MIDI annotations from RWC-POP-100 [27]. Vocals are excluded by a music source separation model Demucs [28].

5.2 Training Configuration

We train the proposed model using 4 RTX8000s with an initial learning rate of $2e-3$ and a batch of 24 20-second samples for 10 epochs. We set the warm-up epoch to 2 and update the model using a cross-entropy reconstruction loss. During training, for each audio sample, we simply sample a text prompt from a text description set for each music segment: {*melodic music, catchy song, a song, music tracks*}.

5.3 Evaluation

We separate each audio clip in the test set into 4 stems using Demucs and discard the vocal track. As shown in Table 2, “*Chord-only*” signifies no drums and MIDI controls, while “*Full*” indicates both drums and MIDI controls. Within each group, we generate 16 20-second audio samples per given chord progression, employing various text prompts while keeping the same chord unchanged. In total, we have 4 test groups, each with 800 generated audio samples.

We report weighted recall score for chord accuracy, standard F-measure for rhythm control, CLAP [29] score for text control evaluation, and Fréchet Audio Distance (FAD) [30] for audio quality measure. As depicted in Table 2, Chord_{rec} represents chord root accuracy, and Chord_{rec}^{*} assesses full chord accuracy. FAD_{vgg}^{*} quantifies the audio dissimilarity between generated samples

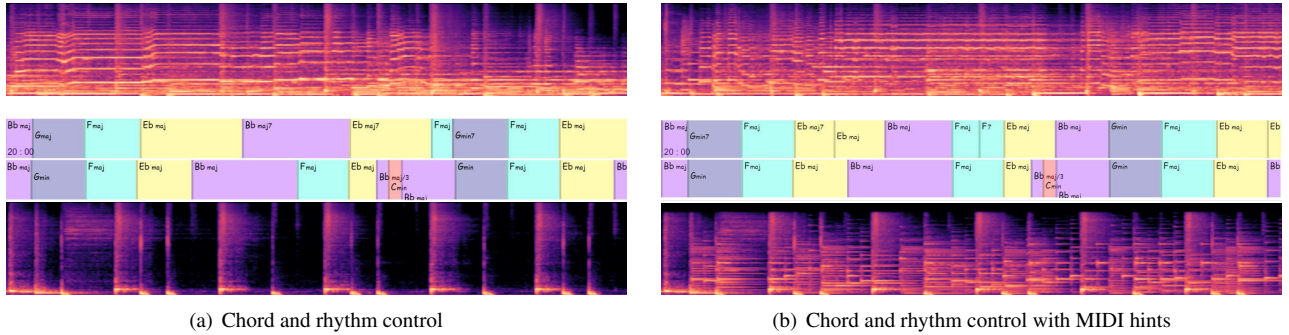


Figure 3. Comparison of generated samples and groundtruth. The top two rows are generated samples, while the bottom rows are reference soundtracks. The text prompt is “*lazy jazz composition features a captivating saxophone solo that effortlessly melds with piano chords, skillfully weaving its way through the melody with languid grace. Instruments: saxophone, piano, drums*”.

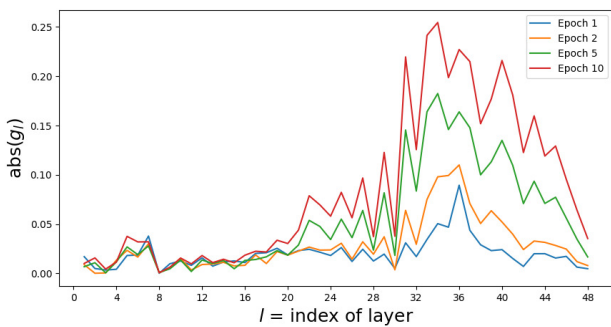


Figure 4. The variation of $|g_l|$ during training.

and groundtruth audio from the RWC-POP-100 subset, whereas $FAD_{V_{gg}}$ assesses this dissimilarity using the remaining 50 audios within RWC-POP-100.

5.4 Results

As illustrated in Table 2 and Figure 3, our model excels in chord and rhythm control while maintaining the text-conditioned ability, even though we do not train the model with real text annotations. We do not report chord and beat accuracy for the baseline model, as it cannot use these controls. Code details and more demos are publicly available online.

5.4.1 Low-resource Fine-tuning

During fine-tuning, our observations suggest that our model works better with a smaller training dataset characterized by high-quality audio fidelity than a larger one featuring pseudo-separated instrumental ground truth. As indicated in Table 3, we discern a trade-off between controllability and semantic correlation. As the number of trainable layers increases, the model achieves a simultaneous improvement in chord recall score while witnessing a reduction in $CLAP_{src}$. In addition, as shown in Fig 4, as the layers go deeper, the absolute value of gate factor g_l increases. It demonstrates the proposed adapter primarily affects the topmost layers of the decoder transformer, implying that the lower layers are likely responsible for mod-

eling high-level semantics, while the upper layers shape the finer details of the content.

5.4.2 Chord and Rhythm Control

As shown in Table 2 and Fig 3(a), the chord control capability of our model strengthens as MIDI hints are provided. Additionally, the results highlight the model’s adeptness at rhythm control when a drum track is included. However, in cases of semantic conflict between the content-based condition and the text prompt, we observe that the model tends to prioritize the former, leading to music that aligns with the drum pattern rather than the prompt.

5.4.3 Variation Generation and Arrangement

As depicted in Fig 3(b), our model, with the assistance of MIDI hints, can produce variations by integrating musical elements from conditioned MIDI tracks, such as motifs and walking bass. Furthermore, we have observed instances where the generated audio aligns with the original main or counter melodies. This facilitates idea-driven variation generation and semantic-based arrangement within the provided polyphonic piano roll.

6. CONCLUSION

We present a content-based music generative model, achieved by fine-tuning a pre-trained Transformer-based audio language model using the proposed condition adaptor. Our experimental results substantiate its proficiency in seamlessly integrating chord progressions, rhythm patterns, MIDI, and text prompts into the generated music. Our work bridges the gap of direct control via musical elements and audio conditions in the music audio generation field. Furthermore, the proposed condition adaptor facilitates efficient low-resource fine-tuning, even with a relatively small unannotated training set. Nonetheless, results generated with conflicting audio, MIDI, and text prompts may lack musicality and may not fully meet semantic control expectations. In the future, we aim to explore further enhancements in the areas of harmonic direct control and content-based generation.

7. ACKNOWLEDGMENTS

This work was supported in part through the NYU and NYUSH IT High Performance Computing resources, services, and staff expertise.

8. REFERENCES

- [1] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, and I. Sutskever, “Jukebox: A generative model for music,” *arXiv preprint arXiv:2005.00341*, 2020.
- [2] H. Liu, Z. Chen, Y. Yuan, X. Mei, X. Liu, D. Mandic, W. Wang, and M. D. Plumbley, “Audioldm: Text-to-audio generation with latent diffusion models,” *arXiv preprint arXiv:2301.12503*, 2023.
- [3] F. Schneider, Z. Jin, and B. Schölkopf, “Moûsai: Text-to-music generation with long-context latent diffusion,” *arXiv preprint arXiv:2301.11757*, 2023.
- [4] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Défossez, “Simple and controllable music generation,” *arXiv preprint arXiv:2306.05284*, 2023.
- [5] Q. Huang, D. S. Park, T. Wang, T. I. Denk, A. Ly, N. Chen, Z. Zhang, Z. Zhang, J. Yu, C. Frank *et al.*, “Noise2music: Text-conditioned music generation with diffusion models,” *arXiv preprint arXiv:2302.03917*, 2023.
- [6] A. Agostinelli, T. I. Denk, Z. Borsos, J. Engel, M. Verzetti, A. Caillon, Q. Huang, A. Jansen, A. Roberts, M. Tagliasacchi *et al.*, “MusicLM: Generating music from text,” *arXiv preprint arXiv:2301.11325*, 2023.
- [7] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, “Exploring the limits of transfer learning with a unified text-to-text transformer,” *The Journal of Machine Learning Research*, vol. 21, no. 1, pp. 5485–5551, 2020.
- [8] A. Guzhov, F. Raue, J. Hees, and A. Dengel, “Audioclip: Extending clip to image, text and audio,” in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 976–980.
- [9] Q. Huang, A. Jansen, J. Lee, R. Ganti, J. Y. Li, and D. P. W. Ellis, “Mulan: A joint embedding of music audio and natural language,” in *Proceedings of the 23rd International Society for Music Information Retrieval Conference*, P. Rao, H. A. Murthy, A. Srinivasamurthy, R. M. Bittner, R. C. Repetto, M. Goto, X. Serra, and M. Miron, Eds., 2022, pp. 559–566.
- [10] C. Donahue, A. Caillon, A. Roberts, E. Manilow, P. Esling, A. Agostinelli, M. Verzetti, I. Simon, O. Pietquin, N. Zeghidour *et al.*, “Singsong: Generating musical accompaniments from singing,” *arXiv preprint arXiv:2301.12662*, 2023.
- [11] R. Zhang, J. Han, A. Zhou, X. Hu, S. Yan, P. Lu, H. Li, P. Gao, and Y. Qiao, “Llama-adapter: Efficient fine-tuning of language models with zero-init attention,” *arXiv preprint arXiv:2303.16199*, 2023.
- [12] A. Van Den Oord, O. Vinyals *et al.*, “Neural discrete representation learning,” *Advances in neural information processing systems*, vol. 30, 2017.
- [13] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, “High fidelity neural audio compression,” *arXiv preprint arXiv:2210.13438*, 2022.
- [14] S.-L. Wu, C. Donahue, S. Watanabe, and N. J. Bryan, “Music controlnet: Multiple time-varying controls for music generation,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 32, pp. 2692–2703, 2024.
- [15] J. D. M.-W. C. Kenton and L. K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” in *Proceedings of NAACL-HLT*, 2019, pp. 4171–4186.
- [16] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever *et al.*, “Language models are unsupervised multitask learners,” *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.
- [17] X. L. Li and P. Liang, “Prefix-tuning: Optimizing continuous prompts for generation,” in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2021, pp. 4582–4597.
- [18] X. Liu, K. Ji, Y. Fu, W. Tam, Z. Du, Z. Yang, and J. Tang, “P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks,” in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2022, pp. 61–68.
- [19] E. J. Hu, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen *et al.*, “Lora: Low-rank adaptation of large language models,” in *International Conference on Learning Representations*, 2021.
- [20] P. Gao, J. Han, R. Zhang, Z. Lin, S. Geng, A. Zhou, W. Zhang, P. Lu, C. He, X. Yue *et al.*, “Llama-adapter v2: Parameter-efficient visual instruction model,” *arXiv preprint arXiv:2304.15010*, 2023.
- [21] N. Zeghidour, A. Luebs, A. Omran, J. Skoglund, and M. Tagliasacchi, “Soundstream: An end-to-end neural audio codec,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 495–507, 2021.
- [22] J. Jiang, K. Chen, W. Li, and G. Xia, “Large-vocabulary chord transcription via chord structure decomposition,” in *Proceedings of the 20th International Society for Music Information Retrieval Conference*,

- A. Flexer, G. Peeters, J. Urbano, and A. Volk, Eds., 2019, pp. 644–651.
- [23] Y.-T. Wu, Y.-J. Luo, T.-P. Chen, I. Wei, J.-Y. Hsu, Y.-C. Chuang, L. Su *et al.*, “Omnizart: A general toolbox for automatic music transcription,” *The Journal of Open Source Software*, vol. 6, no. 68, p. 3391, 2021.
- [24] S. Böck, F. Korzeniowski, J. Schlüter, F. Krebs, and G. Widmer, “Madmom: A new python audio and music signal processing library,” in *Proceedings of the 24th ACM international conference on Multimedia*, 2016, pp. 1174–1178.
- [25] Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimilakis, and R. Bittner, “Musdb18-a corpus for music separation,” 2017.
- [26] Y.-T. Wu, Y.-J. Luo, T.-P. Chen, I.-C. Wei, J.-Y. Hsu, Y.-C. Chuang, and L. Su, “Omnizart: A general toolbox for automatic music transcription,” *Journal of Open Source Software*, vol. 6, no. 68, p. 3391, 2021. [Online]. Available: <https://doi.org/10.21105/joss.03391>
- [27] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, “Rwc music database: Popular, classical and jazz music databases,” in *Proceedings of the 3rd International Society for Music Information Retrieval Conference*, 2002.
- [28] A. Défossez, N. Usunier, L. Bottou, and F. Bach, “Demucs: Deep extractor for music sources with extra unlabeled data remixed,” *arXiv preprint arXiv:1909.01174*, 2019.
- [29] Y. Wu, K. Chen, T. Zhang, Y. Hui, T. Berg-Kirkpatrick, and S. Dubnov, “Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation,” in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.
- [30] K. Kilgour, M. Zuluaga, D. Roblek, and M. Sharifi, “Fréchet audio distance: A metric for evaluating music enhancement algorithms,” *arXiv preprint arXiv:1812.08466*, 2018.

EXPLORING THE INNER MECHANISMS OF LARGE GENERATIVE MUSIC MODELS

Marcel A. Vélez Vásquez*

Charlotte Pouw*
Willem Zuidema

John Ashley Burgoyne

Institute for Logic, Language and Computation, University of Amsterdam

{m.a.velezvasquez,c.m.pouw,j.a.burgoyne,w.h.zuidema}@uva.nl

ABSTRACT

Generative models are starting to become very good at generating realistic text, images, and even music. Identifying how exactly these models conceptualize data has become crucial. To date, however, interpretability research has mainly focused on the text and image domain, leaving a gap in the music domain. In this paper, we investigate the transferability of straightforward text-oriented interpretability techniques to the music domain. Specifically, we examine the usability of these techniques for analyzing how the generative music model MusicGen constructs representations of human-interpretable musicological concepts. Using the DecoderLens, we gain insight into how the model gradually composes these concepts, and using interchange interventions, we observe the contributions of individual model components in generating the sound of specific instruments and genres. We also encounter several shortcomings of the interpretability techniques for the music domain, which underscore the complexity of music and need for proper audio-oriented adaptation. Our research marks an initial step toward understanding generative music models, *fundamentally*, paving the way for future advancements in controlling music generation.

1. INTRODUCTION

Generative AI systems for music have become mainstream in the past year, and have become a popular application for consumers, an eye-catching product for AI engineers and companies, and a key research topic for researchers. The most successful of these systems are built on top of recent advances in deep learning for text and audio encoding, and add a large *text-to-music* model, using the Transformer-architecture [1], to allow users to generate music from a text and/or audio prompt [2–5].

These systems are typically trained end-to-end, and present us with the infamous *black box problem*: it is ex-

*These authors contributed equally to this work.

tremely difficult to understand what is happening in the billions of mathematical operations between input and generated output. This severely limits the ability of users to influence the generated output (other than by just trying a different prompt), of companies to trace an individual output to examples from the training set (and give credit where credit is due), of engineers to diagnose shortcomings and improve the system (other than by retraining on a better dataset or bigger model) and of music researchers to relate the behavior of these models to the large body of existing theoretical and empirical work on how music works.

‘Opening the black box’ of generative music models is therefore a key new area of research. In this paper, we build on advances with interpretability techniques for generative text models. Although there are many important differences between text and music (including their discrete versus continuous nature, and the temporal resolution needed to build good models), we find that those techniques can be adapted to the music domain and indeed give us insights into the inner mechanisms. We focus on one representative, open-source generative music model, MusicGen [5], and on two representative human-interpretable concepts: *musical instrument* and *genre*. We ask: can we localize and manipulate those concepts in MusicGen? We report success on these tasks, and discuss in the final parts of the paper how these initial steps might be extended to the full toolbox needed to successfully address the negative consequences of the black box problem.

2. RELATED WORK

Interpretability research is relatively sparse in the music domain. Previous work has analyzed neural models trained on symbolic music representations (MIDI) using probing classifiers [6, 7], visual inspection of the embedding space [8], listenable explanations for classification models [9, 10], or post-hoc explanations in the form of highlighted parts of a piano roll [11]. To the best of our knowledge, no previous work has tried to interpret generative music models trained on raw audio data.

In the text domain, the Transformer architecture is dominating the field [1]. Recent advancements in interpretability methods are built on a key characteristic of Transformer models: their use of residual connections across layers. Typically, each layer of the Transformer contains an attention component and a Multilayer Perceptron (MLP), both



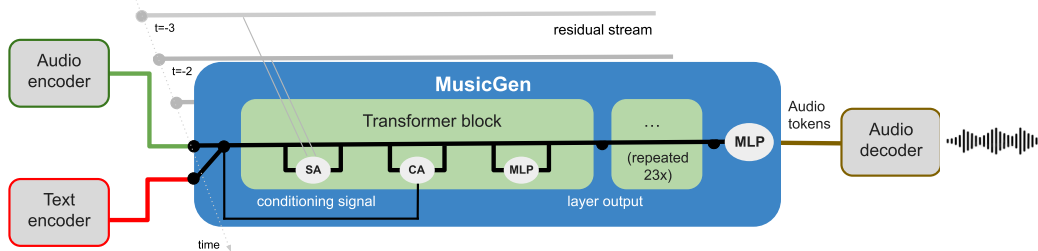


Figure 1: Architecture of MusicGen. SA = self-attention, CA = cross-attention, MLP = Multi-Layer Perceptron. The self-attention allows for communication between audio tokens; the cross-attention allows for communication between the audio tokens and the conditioning signal (consisting of an encoded text prompt and optionally an encoded melody prompt).

of which interact with the residual stream. This arrangement ensures that information in the residual stream remains accessible throughout all layers, facilitating the development of stable representations.

One method that exploits this characteristic is the **DecoderLens** [12]. It is an adaptation of the LogitLens [13], which was designed to interpret intermediate representations of decoder-only Transformer models. It applies the unembedding matrix to intermediate layer outputs to obtain a logit distribution over the vocabulary for each intermediate layer. The DecoderLens was designed to interpret the intermediate representations of encoder-decoder models. It applies the decoder to intermediate encoder outputs, providing insight in what information that can be decoded from earlier layers. In the original DecoderLens study, the authors use the DecoderLens to analyze how encoder-decoder models build meaningful representations for tasks like machine translation and question answering.

Additionally, **interchange interventions** have been used to identify model components that are causally involved in specific behavior, such as *greater-than* reasoning [14], pronoun resolution [15], and gender bias [16, 17]. By systematically altering model inputs or components and observing resultant changes in behavior, researchers have gained valuable insights into the underlying mechanisms driving model performance and decision making.

3. EXPERIMENTAL SETUP

We conduct two interpretability experiments, one using the DecoderLens and one using interchange interventions, for interpreting the inner workings of a popular generative music model, MusicGen [5].

MusicGen is an open-source, Transformer-based music generation model, built by researchers at Meta [5]. Its architecture is sketched in Figure 1. The model generates discrete audio tokens, optionally conditioned on a text prompt and/or a melody. Text prompts are first processed by a Transformer-based text encoder; music prompts by a 32-kHz EnCodec [18] tokenizer sampled at 50 Hz. MusicGen is autoregressive, and transforms its input over successive layers through multi-head *self-attention* (integrating information across timesteps) and MLP components, while incorporating information from the text and/or melody prompt through *cross-attention* to the respective

encoders. The generated audio tokens are then decoded into a waveform by an EnCodec decoder.

We analyze three model sizes: **MusicGen-small** (300M parameters, 1024 dimensions), **MusicGen-medium** (1.5B parameters, 1536 dimensions), and **MusicGen-large** (3.3B parameters, 2048 dimensions). We set the generation duration to 4 seconds and keep the rest of the MusicGen parameters at their default values.*

In all of our experiments, we only condition MusicGen on text prompts, not on melody prompts. We constructed the following template for our text prompts: *Compose a [MOOD] [GENRE] piece with a [INSTRUMENT] melody. Use a [TEMPO] tempo.* We only modify the components in between brackets and keep the remaining context fixed.

3.1 Experiment 1: DecoderLens

In our first experiment, we use the DecoderLens to globally examine how MusicGen builds up representations of musical concepts across its Transformer layers. This involves extracting intermediate representations from each layer and using the EnCodec decoder to map these to audio. We examine the representation of *musical instrument* and *genre*. We select four instruments and six genres (listed in Table 1) and construct 100 text prompts per category using our predefined template. We feed these prompts to MusicGen and use the DecoderLens to obtain 100 music outputs for each of the 24 Transformer layers.

We evaluate the **recognizability** of our selected musical concepts within the intermediate music outputs by employing an audio classifier that was among the top-ranking classifiers of the 2021 HEAR challenge [19]. This multi-label classifier, trained on AudioSet [20], provides a logit distribution across 527 audio classes, including both musical concepts and other sounds such as speech and environmental noises. We run each intermediate music output through the audio classifier and compute the **normalized discounted cumulative gain (NDCG)** [21], a metric commonly used in information retrieval to measure ranking quality. We consider this to be a proxy for the recognizability of a specific concept.

For each concept, we establish an ‘ideal ranking’ of the audio classes by assigning relevant labels (listed in Table

*For our code and listening examples, see our GitHub page: <https://github.com/Marcel-Velez/musicgen-mech-interp>

Category	Selection	Relevant Labels
Instrument	Guitar	Guitar, Acoustic Guitar, Electric Guitar, Bass Guitar, Plucked String Instrument
	Piano	Piano, Electric Piano, Keyboard (musical)
	Trumpet	Trumpet, Brass Instrument
	Violin	Violin/Fiddle, String Section, Bowed String Instrument
Genre	Classical	Classical Music
	Jazz	Jazz, Rhythm and Blues
	Pop	Pop Music
	Rock	Rock Music, Rock and Roll, Progressive Rock, Punk Rock
	EDM	Electronic Dance Music, Electronic Music, Techno, Drum and Bass, Dubstep, House Music
	Hip Hop	Hip Hop Music

Table 1: Relevant labels of the external audio classifier [19] for each category.

1) a relevance score of 1, while assigning all other labels a score of 0. This methodology facilitates a comparison between the predicted ranking generated by the audio classifier and our predefined ideal ranking. NDCG returns a high score when the relevant labels from our ideal ranking are ranked high by the audio classifier, with a score of 1.0 indicating a perfect predicted ranking.

3.2 Experiment 2: Interchange Interventions

In our second experiment, we use interchange interventions to identify the crucial model components responsible for generating specific musical instrument and genre sounds. The workflow for performing these interventions, which we apply for every permutation of two categories in Table 1, is as follows.

1. Construct two sets of text prompts: one for a concept such as *guitar* (henceforth the **original concept**), and one for a contrasting concept such as *piano* (henceforth the **desired concept**).
2. Run both sets of prompts through MusicGen and save the output of each individual component within the MusicGen Transformer (these model components are further explained in section 3.2.1). This leaves us with two activation caches: one for the original concept, and one for the desired concept.
3. While running MusicGen on the original concept prompts again, replace the output of a specific model component with the average output of that model component across all desired concept prompts. After the intervention, the forward pass continues as normal, but yields an **intervened music fragment**. Repeat this step for all model components.
4. To evaluate the effect of each individual intervention, run a classifier on the original and intervened music fragment, and assess how the odds for the original and desired concept labels changed (we use the same audio classifier that we used for our DecoderLens experiments). If the intervention was

effective, the odds for the original concept should have decreased, and the odds for the desired concept should have increased.

3.2.1 Intervention techniques

We explore two intervention techniques: **replace** and **adjust**. With the “replace” technique, we entirely substitute an activation from an individual *original concept* prompt with the average activation of 100 *desired concept* prompts. With the “adjust” technique, we first subtract the average activation of 100 *original concept* prompts from an individual *original concept* activation. Then, we add the average activation of 100 *desired concept* prompts to that result. The latter technique is inspired by the idea that, in language models, semantic properties of words can be adjusted by adding or subtracting specific word vectors, e.g., $king - man + woman = queen$ [22], or in our case, $music\ with\ guitar - guitar + piano = music\ with\ piano$.

We perform the interchange interventions across all 24 Transformer layers of MusicGen. Each layer consists of a self-attention block, a cross-attention block, and a Multi-Layer Perceptron (MLP). Thus, each intervention consists of swapping the output of one of these three components per layer individually. For a single text prompt, this adds up to 24 layers \times 3 layer components = 72 interventions.

3.2.2 Within-category vs. cross-category interventions

We investigate intervention effects on both **instrument** prompts and **genre** prompts. For each instrument and genre category listed in Table 1, we construct 100 text prompts based on the template outlined in Section 3. We then perform **within-** and **cross-category** interventions.

In within-category interventions, we interchange model activations between two sets of instrument prompts, or between two sets of genre prompts. For instance, we introduce *piano* activations during a forward pass intended for *guitar*, or we introduce *jazz* activations during a forward pass intended for *classical*.

In cross-category interventions, we interchange model activations between a set of instrument and a set of genre prompts. For example, we introduce *piano* activations during a *classical* forward pass, or we introduce *jazz* activations during a *guitar* forward pass.

3.2.3 Evaluating Intervention Effects

An ideal intervention removes the original concept and introduces the desired concept, but does not change anything about the rest of the music. We therefore evaluate the interventions along two axes: **intervention effectiveness** and **intervention precision**.

We evaluate intervention effectiveness using a metric that quantifies the impact on both the original and desired concept, inspired by the metric used in [23]. Specifically, we calculate:

$$\log \frac{\text{odds}(\text{original}_{\text{before}})}{\text{odds}(\text{original}_{\text{after}})} - \log \frac{\text{odds}(\text{desired}_{\text{before}})}{\text{odds}(\text{desired}_{\text{after}})} \quad (1)$$

A high score indicates that the odds of the original concept decreased as a result of the intervention, or that the odds

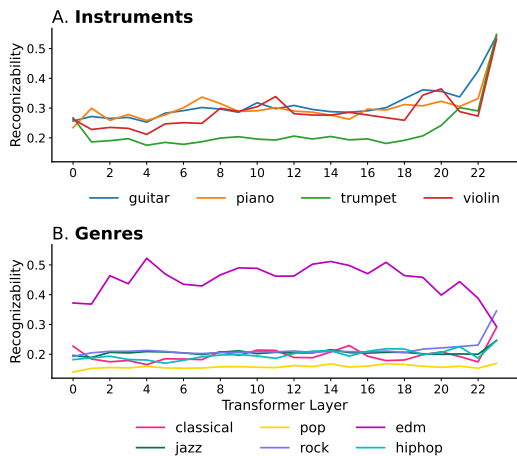


Figure 2: Results for Experiment 1 (**DecoderLens**): Average recognizability of instruments (A) and genres (B) across Transformer layers in MusicGen-small (as measured by the NDCG).

of the desired concept increased. We calculate odds by applying a softmax function over the logit distribution of our external audio classifier.*

We evaluate intervention precision using the **Kullback-Leibler (KL) divergence**, which quantifies the overall shift in softmax distribution across all audio labels. This metric gauges how much the intervened music fragment differs from the original. Ideally, our intervention only has an effect on the odds for the original and desired concept labels, leaving the odds for the others unchanged. This means that low KL scores are desirable, but for ease of interpretation, we reverse them to make higher scores better.

4. RESULTS

4.1 Results Experiment 1: DecoderLens

Figure 2 shows the average recognizability of our selected instruments and genres across the Transformer layers of MusicGen-small. For instruments, we observe relatively stable recognizability in layers 0-19, followed by a gradual increase in layers 20-23. This indicates that MusicGen gradually builds up the representation of individual musical instruments across layers, with the final layers playing a crucial role. The pattern for genres is different. Except for EDM, all genres exhibit the same recognizability across layers, with a slight increase in the final layer. In contrast, EDM exhibits high scores across layers, with a gradual decline in the final layers. This pattern could be attributed to the genre distribution in MusicGen’s training data: EDM is disproportionately represented [5], possibly leading to the model overfitting to EDM characteristics. It could be that most Transformer layers are tuned to generate music reminiscent of EDM, and only the final layer has learned to integrate genre information from the input text prompt.

An alternative explanation is that the DecoderLens is

*For simplicity, we only use one label for each concept in this analysis, i.e., the first label listed for each category in Table 1.

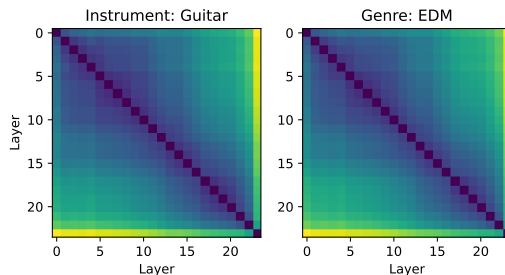


Figure 3: Self-similarity matrices (Euclidean distance) of intermediate layer outputs of the Transformer block within MusicGen-small, when conditioning the model on text prompts containing the instrument *guitar* (left) or containing the genre *EDM* (right). Both matrices are averaged over 100 prompts. A dark color indicates high similarity, a light color indicates low similarity.

currently not optimized for music. Upon listening to DecoderLens outputs, we noted instances of distortion and disorganization. While these outputs predominantly resemble the EDM genre when compared to other genres like *classical* or *jazz*, they may simply reflect artifacts of the EnCodec decoder, trained specifically for decoding representations from the final Transformer layer. The representations of earlier layers may be harder to decode, as they may follow a different representational distribution.

To explore this alternative hypothesis, we examined the similarity of intermediate layer outputs within the Transformer block of MusicGen-small. We re-ran the model with the same 100 text prompts for each concept and extracted the output of each intermediate Transformer layer. We averaged these layer outputs across time and then computed the Euclidean distance between all combinations of layers. Figure 3 displays the results for *guitar* and *EDM*, but similar patterns were observed for the other instruments and genres listed in Table 1. We indeed observe that the final layer (24) is highly dissimilar to the other layers.

Further exploration, possibly involving a “translation model” that maps intermediate layer outputs to final layer outputs [24, 25], could help to refine the DecoderLens for the music domain.

4.2 Results Experiment 2: Interchange Interventions

4.2.1 Intervention effects across model components

Figure 4A shows the average effect of intervening on different components (MLP, self-attention, cross-attention) across the Transformer layers of MusicGen-small, for both the “replace” technique (solid lines) and the “adjust” technique (dashed lines) (results for the medium and large model can be found in the Appendix). Starting with the “replace” technique, we observe a clear contrast between the MLP and the attention components: the MLP consistently shows positive intervention effects, whereas both self-attention and cross-attention predominantly show negative effects. With the “adjust” technique, intervening on the attention components results in positive scores, but they

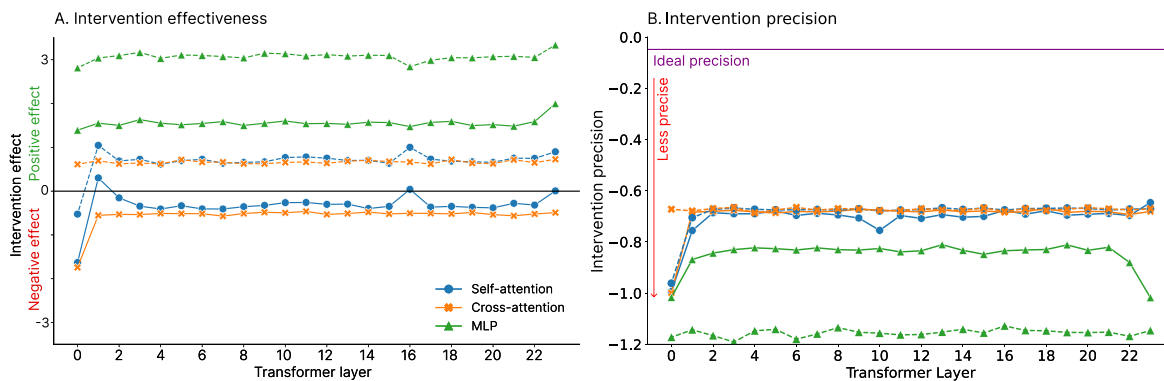


Figure 4: Results for Experiment 2 (**interchange interventions**) across model components and layers, for MusicGen-small only. Solid lines show the result for the “replace” technique, dashed lines show the results for the “adjust” technique. Figure A shows intervention effectiveness (as measured by our log odds ratio); Figure B shows intervention precision (as measured by the inversed KL-divergence). Higher scores are better for both metrics.

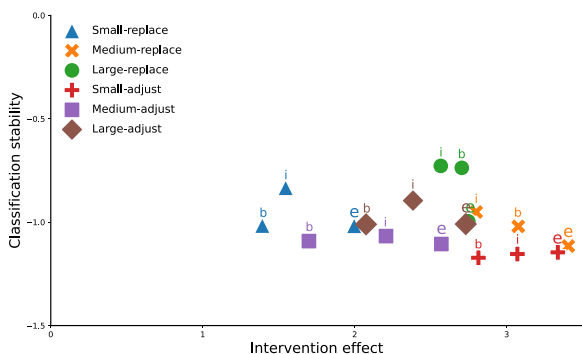


Figure 5: Intervention effect vs. intervention precision of the MLP across model sizes (small, medium, large) and intervention techniques (replace vs. adjust). The datapoints are labelled according to the layer where the intervention was performed (b = *beginning*, i = *intermediate*, e = *end*).

are still much lower than intervening on the MLP. This suggests that manipulating the sound of instruments and genres is achievable by intervening on the MLP output, but not to the same extent by intervening on the attention outputs.

Shifting focus to Figure 4B, we observe that interventions on all model components produce negative intervention precision scores. This means that all interventions induce some type of alteration to the audio output. To interpret the magnitude of these changes, we compare them to the “ideal” intervention precision score, where only the original concept and desired concept probabilities flip while everything else remains unchanged. We find that the actual stability scores are much lower than this ideal scenario, suggesting that the interventions are rather invasive and change the audio in a way that goes beyond merely flipping the original concept to the desired concept. A potential future approach could be to perform the interventions on specific frames rather than on the entire audio [26].

4.2.2 Dissecting intervention effects of the MLP

Figure 4A suggests that interventions on the MLP yield the desired alteration (reducing the original concept and

increasing the desired concept). We now analyze these effects in more detail. Figure 5 shows the relationship between intervention effectiveness and intervention precision for different model sizes (small, medium, large) and intervention techniques (replace vs. adjust) for the MLP only. For each combination of model size and intervention technique, we plot three scores: 1) the score for intervening on the **first layer**, 2) the average score for intervening on the **intermediate layers**, and 3) the score for intervening on the **final layer** (we average the intermediate layers since they showed very stable effects in Figure 4).

The effectiveness of intervention techniques seems to depend on model size. We see that the “adjust” technique performs best with the small model, while the “replace” technique shows better results with the medium and large models. We also notice that using the “replace” technique for the large model yields the highest intervention precision overall. This suggests that the large model might represent musical concepts in a more modular manner compared to the smaller ones; thus, we can more easily modify only a single concept without affecting other concepts. One possible explanation could be that in the smaller models, due to limited space, individual neurons represent multiple features simultaneously, a phenomenon known as *superposition* [27].

Finally, for all model sizes, we observe that intervening at the final layer produces the best results, followed by intervening at intermediate layers. Intervening at the first layer tends to be least effective. This pattern may be attributed to the model’s ability to “compensate” for interventions: when we intervene at early layers, the model still has plenty of opportunity to change the original or desired concept as it progresses through its forward pass.

4.2.3 Effect on original vs. desired concept

Figure 6 displays the intervention effect on the original and desired concept **separately**, as well as the **combined** score ($score_{original} + score_{desired}$) for all three model sizes. We also separately show the effects for different intervention types: **inter-category** (instrument-

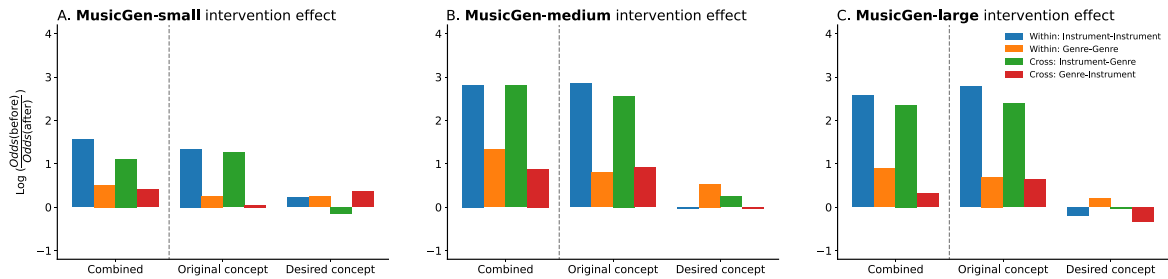


Figure 6: Combined and separated intervention effect of the **replace** technique on the original and desired concept for the MLP only, in the **small** (A), **medium** (B), and **large** (C) version of MusicGen. Bar colors indicate the intervention type (inter-category/cross-category).

instrument, genre-genre) and **cross-category** (instrument-genre, genre-instrument). Since intervention effects were similar across layers, each bar represents the average intervention effect across layers. We showcase the results for the “replace” technique here; the results for the “adjust” technique can be found in the Appendix.

When examining the scores for the original and desired concepts separately, a clear pattern emerges across all model sizes: the impact on the original concept is much bigger than on the desired concept. This suggests that interventions effectively reduce the original concept, but do not introduce the desired concept as effectively.

For the inter-category intervention effects (blue and orange bars), we observe that instrument–instrument interventions are much more effective than genre–genre interventions. This indicates that it is easier to manipulate the sound of individual instruments in the output than the sound of genres. This in turn suggests that instruments are represented in a more modular fashion than genres, which makes sense given the complex combination of features that are typically involved in a genre.

The pattern for cross-category interventions (green and red bars) is similar: instrument–genre interventions are more effective than genre–instrument interventions. Specifically, interventions inserting genre activations during a forward pass with an instrument prompt notably impact the instrument sound—but interventions inserting instrument activations during a forward pass with a genre prompt have a less pronounced effect on the genre. This supports the notion that genres are represented with multiple features, making them more resistant to manipulation compared to instruments’ more modular representation.

5. DISCUSSION

In this work, we explored the usability of text-oriented interpretability techniques for analyzing the representation of human-interpretable musicological concepts in MusicGen. We applied the DecoderLens for globally analyzing how the model conceptualizes musical instruments and genres across layers, and applied interchange interventions to dissect the role of individual layer components in generating specific instrument and genre sounds across several model sizes.

In our investigation, applying the DecoderLens to Mu-

sicGen revealed significant challenges in generating coherent audio from intermediate layers, a limitation underscored by the self-similarity matrix which showed that the last layer is vastly different from the rest of the model. Similarly, our attempts at interchange interventions, aimed at dissecting the influence of specific model components per layer on musical output, was fairly effective in removing existing musical concepts but was unsuccessful when it came to injecting new ones into the network, across all examined model sizes. These outcomes not only show the complexities inherent in interpreting generative music models but also underscore the need for music/audio specific intervention techniques.

In future work, we aim to adapt these interpretability techniques to be more suitable for audio. As for the DecoderLens, a single linear layer could be trained to map intermediate representations to final layer representations, possibly allowing for better decodability. Furthermore, we aim to explore different intervention techniques (e.g., intervening on specific frames instead of the entire sequence), which could contribute to less drastic alterations of the audio while still changing the desired concepts. These improvements may allow us to additionally explore other facets of music generation, such as tempo and rhythm.

5.1 Limitations

We used quantitative metrics based on a machine learning model to evaluate intervention effects, acknowledging that this approach introduces extra noise. However, it allowed us to investigate a larger parameter space compared to a user-listening study. For instance, we evaluated 100 prompts across multiple model components (100 * 3 components * 48 layers for the large model) for each permutation of concepts. Although human ratings from a listening study could provide many complementary insights, setting up such experiments is costly. Additionally, the authors themselves listened to several intervention results and noticed a lot of variation across samples, complicating the selection of a representative subset for a listening study. Therefore, we believe establishing robust quantitative results first is more practical. These results can inform the design of more focused and efficient listening studies, ensuring effective resource use and meaningful insights.

6. ACKNOWLEDGMENTS

We are grateful to Jamie Vlegels and Stefan Wijnja, whose research project inspired this work. This research was supported by the Dutch Research Council (NWO) as part of the project InDeep (NWA.1292.19.399).

7. REFERENCES

- [1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.
- [2] A. Agostinelli, T. Denk, Z. Borsos, J. Engel, M. Verzett, A. Caillon, Q. Huang, A. Jansen, A. Roberts, M. Tagliasacchi *et al.*, "Musiclm: Generating music from text," 2023.
- [3] Q. Huang, D. S. Park, T. Wang, T. I. Denk, A. Ly, N. Chen, Z. Zhang, Z. Zhang, J. Yu, C. Frank *et al.*, "Noise2music: Text-conditioned music generation with diffusion models," *arXiv preprint arXiv:2302.03917*, 2023.
- [4] F. Schneider, O. Kamal, Z. Jin, and B. Schölkopf, "Mousai: Text-to-music generation with long-context latent diffusion," *arXiv preprint arXiv:2301.11757*, 2023.
- [5] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Défossez, "Simple and controllable music generation," 2023.
- [6] M. Keller, G. Loiseau, and L. Bigo, "What musical knowledge does self-attention learn?" in *Workshop on NLP for Music and Spoken Audio (NLP4MuSA 2021)*, 2021.
- [7] S. Han, H. Ihm, and W. Lim, "Systematic analysis of music representations from bert," *arXiv preprint arXiv:2306.04628*, 2023.
- [8] N. Cosme-Clifford, J. Symons, K. Kapoor, and C. W. White, "Musicological interpretability in generative transformers," in *2023 4th International Symposium on the Internet of Sounds*. IEEE, 2023, pp. 1–9.
- [9] V. Haunschmid, E. Manilow, and G. Widmer, "audioLIME: Listenable Explanations Using Source Separation," 13th International Workshop on Machine Learning and Music, 2020.
- [10] S. Mishra, B. L. Sturm, and S. Dixon, "Local interpretable model-agnostic explanations for music content analysis." in *ISMIR*, vol. 53, 2017, pp. 537–543.
- [11] F. Foscari, K. Hoedt, V. Praher, A. Flexer, and G. Widmer, "Concept-based techniques for" musicologist-friendly" explanations in a deep music classifier," *arXiv preprint arXiv:2208.12485*, 2022.
- [12] A. Langedijk, H. Mohebbi, G. Sarti, W. Zuidema, and J. Jumelet, "Decoderlens: Layerwise interpretation of encoder-decoder transformers," *arXiv preprint arXiv:2310.03686*, 2023.
- [13] nostalgebraist, "Interpreting gpt: The logit lens." 2023. [Online]. Available: <https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens>
- [14] M. Hanna, O. Liu, and A. Variengien, "How does GPT-2 compute greater-than?: Interpreting mathematical abilities in a pre-trained language model," *arXiv preprint arXiv:2305.00586*, 2023.
- [15] T. Yamakoshi, J. McClelland, A. Goldberg, and R. Hawkins, "Causal interventions expose implicit situation models for commonsense language understanding," in *Findings of the Association for Computational Linguistics: ACL 2023*, A. Rogers, J. Boyd-Graber, and N. Okazaki, Eds. Toronto, Canada: Association for Computational Linguistics, Jul. 2023, pp. 13 265–13 293. [Online]. Available: <https://aclanthology.org/2023.findings-acl.839>
- [16] J. Vig, S. Gehrmann, Y. Belinkov, S. Qian, D. Nevo, Y. Singer, and S. Shieber, "Investigating gender bias in language models using causal mediation analysis," *Advances in neural information processing systems*, vol. 33, pp. 12 388–12 401, 2020.
- [17] A. Chintam, R. Beloch, W. Zuidema, M. Hanna, and O. van der Wal, "Identifying and adapting transformer-components responsible for gender bias in an English language model," *arXiv preprint arXiv:2310.12611*, 2023.
- [18] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, "High fidelity neural audio compression," 2022.
- [19] K. Koutini, J. Schlüter, H. Eghbal-Zadeh, and G. Widmer, "Efficient training of audio transformers with patchout," *arXiv preprint arXiv:2110.05069*, 2021.
- [20] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *Proc. IEEE ICASSP 2017*, New Orleans, LA, 2017.
- [21] K. Järvelin and J. Kekäläinen, "Cumulated gain-based evaluation of ir techniques," *ACM Transactions on Information Systems (TOIS)*, vol. 20, no. 4, pp. 422–446, 2002.
- [22] T. Mikolov, W.-t. Yih, and G. Zweig, "Linguistic regularities in continuous space word representations," in *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies*, 2013, pp. 746–751.

- [23] T. Yamakoshi, J. L. McClelland, A. E. Goldberg, and R. D. Hawkins, “Causal interventions expose implicit situation models for commonsense language understanding,” *arXiv preprint arXiv:2306.03882*, 2023.
- [24] N. Belrose, Z. Furman, L. Smith, D. Halawi, I. Ostrovsky, L. McKinney, S. Biderman, and J. Steinhardt, “Eliciting latent predictions from transformers with the tuned lens,” 2023.
- [25] A. Y. Din, T. Karidi, L. Choshen, and M. Geva, “Jump to conclusions: Short-cutting transformers with linear transformations,” *ArXiv*, vol. abs/2303.09435, 2023. [Online]. Available: <https://api.semanticscholar.org/CorpusID:257557722>
- [26] J. Koo, G. Wichern, F. G. Germain, S. Khurana, and J. L. Roux, “Smitin: Self-monitored inference-time intervention for generative music transformers,” *arXiv preprint arXiv:2404.02252*, 2024.
- [27] N. Elhage, T. Hume, C. Olsson, N. Schiefer, T. Henighan, S. Kravec, Z. Hatfield-Dodds, R. Lasenby, D. Drain, C. Chen *et al.*, “Toy models of superposition,” *arXiv preprint arXiv:2209.10652*, 2022.

QUANTITATIVE ANALYSIS OF MELODIC SIMILARITY IN MUSIC COPYRIGHT INFRINGEMENT CASES

Saeyul Park¹ Halla Kim¹ Jiye Jung²
Juyong Park¹ Jeounghoon Kim¹ Juhan Nam¹

¹Graduate School of Culture Technology, KAIST, Korea

²Heinrich Heine University Düsseldorf, Germany

{saeyul_park, kimhalla, juyongp, miru, juhannam}@kaist.ac.kr

jujiy100@uni-duesseldorf.de

ABSTRACT

This study aims to measure the similarity of melodies objectively using natural language processing (NLP) techniques. We utilize Mel2word which is a melody tokenization method based on byte-pair encoding to facilitate the semantic analysis of melodies. In addition, we apply two word weighting methods: the modified Tversky measure for word salience and the TF-IDF method for word importance and uniqueness, to better understand the characteristics of each melodic element. We validate our approach by comparing song vectors calculated from an average of Mel2Word vectors to the ground truth in 108 cases of music copyright infringement, sourced from an extensive review of legal documents from law archives. The results demonstrate that the proposed approach is more in accordance with court rulings and perceptual similarity.

1. INTRODUCTION

Since the landmark case of *Millett v. Snowden*¹ in 1844, music plagiarism has been a contentious issue for over a century. The term “plagiarism” refers to the subcategory of copyright infringement that involves the false designation of authorship and other unattributed uses of copyrighted material [1]. In determining plagiarism, courts have traditionally considered three major aspects of music infringement lawsuits: 1) copyright ownership, 2) accessibility, and 3) substantial similarity [2]. “substantial similarity”, which is the most crucial yet debatable factor, lacks a complete definition with no general agreement [3, 4, 5] due to the varying requisite level from case to case [5]. Court analyses are inconsistent within the same circuit, making it more a matter of quality than quantity [6, 7].

¹ *Millett v. Snowden*, available at: <https://blogs.law.gwu.edu/mcir/case/millett-v-snowden/>



Melodic similarity is usually the determining element in assessing whether or not two musical works are substantially similar [8, 6]. Melody is the most memorable and characteristic feature of music [9, 10], and many cases involve the plagiarism of the melody of an original work [11, 9, 12]. Although numerous studies have developed various quantitative measures of melodic similarity [13, 12, 14, 15], it still remains unclear what constitutes substantial similarity. While a high degree of melodic similarity may suggest plagiarism, it does not necessarily indicate plagiarism. Instead, substantial parts of an existing work that are considered essential and worthy of protection can be crucial in determining plagiarism. For example, in the case of *Hawkes & Sons v. Paramount Film Services* (1934, as cited by [16] and [17]), twenty seconds (of 4 minutes) of a musical work without permission was deemed infringement. Therefore, the use of any “recognizable” parts may establish infringement, even if the overall similarity of the pieces is questionable [17].

This study aims to develop a novel approach for quantitatively evaluating the substantial similarities of melodies by employing natural language processing (NLP) techniques. Due to the shared characteristics between music and language [18, 19, 20], various NLP approaches have been applied to music analysis in different ways [21, 22, 23, 24]. The primary focus of the proposed approach is to define the individual elements of melody using NLP-based methods. To achieve this, we employ Mel2word [25], a novel method for melody segmentation using NLP tokenization techniques to represent melodies as word-like units and capture semantic information through word embeddings. In addition, two word weighting methods are proposed to understand the characteristics of individual melodic elements: a modified Tversky measure for *word salience* and the TF-IDF method for *word importance* and *word uniqueness*. The method is evaluated on 108 plagiarism cases with court decisions and perceptual similarity as ground truth, compiling data from diverse sources to represent one of the most extensive symbolic melodic datasets available. This study provides detailed case analyses, showcasing the numerical and graphical representation of the proposed method and its practical applications. By doing so, we aim to provide empirical and quantitative

evidence for the qualitative aspects of substantial similarities in music.

2. LITERATURE REVIEW

There have been numerous studies on plagiarism detection based on melodic similarities, which can be broadly categorized into two types of approaches: (1) audio-based and (2) text-based.

Audio-based approaches employ music signal processing to develop plagiarism detection tools that can identify similar parts of music [26, 27, 28, 29]. While they use advanced audio-based analysis techniques to determine the level of similarity between songs, they mainly focus on identifying similarities rather than explaining how the degree of plagiarism is related to the level of similarity. The audio-based approaches are particularly useful in plagiarism cases involving unauthorized sampling or use of musical works. However, for research purposes related to artistic analysis, notated music provides more useful information than audio-based analysis [15].

Text-based approaches analyze symbolic musical representations, such as notated music. The study by [12] is a remarkable attempt to quantitatively model court decisions in plagiarism cases. This study compared several similarity calculation algorithms and investigated how melodic similarity calculated by text-based algorithms relates to court decisions based on a sample of US copyright cases from 1970. The study unveiled that an algorithm rooted in statistical methods, notably Tversky’s similarity measures [30], outperformed in predicting court decisions. This finding was further corroborated by research conducted by [31]. Percent Melodic Identity (PMI) also stands out as another major measure in this context. Drawing from automatic sequence alignment algorithms in the field of molecular genetics, [32] introduced the PMI method to quantify melodic similarity, which was further utilized by [33, 34] to successfully predict plagiarism. Recent advancements in music research have demonstrated significant progress, particularly in utilizing vectorized representations. These include fuzzy vector-based approaches [8], CNN-based methods [35], and hybrid approaches [36].

While previous studies have explored the quantitative similarities between melodies, the specific elements contributing to plagiarism and the underlying reasons remain unclear. This gap highlights the need for further investigation into what exactly constitutes melodic plagiarism and “why” these particular elements are implicated. To address this, we propose an NLP-based approach to define individual melodic elements by defining words as the basic unit of text to reconstruct a melody as a sentence of meaningful word units. We also introduce a function that combines psychological and NLP models, particularly Tversky and TF-IDF approaches, aiming to provide a comprehensive framework for understanding melodic similarity.

3. METHODS

The proposed method involves three steps: 1) segmenting melodies using Mel2Word [25], 2) vectorizing melodies

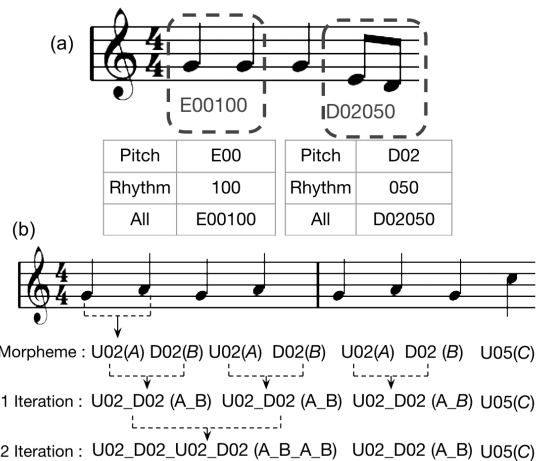


Figure 1: Example of (a) the Mel2Word representation with (b) Byte Pair Encoding (BPE) process.

using the Word2Vec [37] algorithm, and 3) applying word-by-word weighted measures to determine word salience, importance, and uniqueness.

3.1 Textual Representation

Mel2word is a novel text-based representation method to segment melodies into word-like units [25]. In this textual representation, each note is translated into a pitch feature indicating the interval’s direction and size, alongside a rhythm feature denoted by the inter-onset interval (IOI) between consecutive notes. Specifically, pitch features are represented by the first character indicating the melody’s direction (“U” for upward, “D” for downward, and “E” for no change), followed by a two-digit number specifying the interval size. Rhythm features are depicted with three-digit numbers, obtained by multiplying the IOI by 100, assuming a quarter note equals one beat with a 16th note quantization. This unit, as depicted in Figure 1-(a), composed of two notes, form “morphemes” utilized for constructing a melody word dictionary using Byte Pair Encoding, a commonly used tokenization technique in the field of NLP.

3.2 Byte-Pair Encoding

Mel2Word represents melodies as word-like units using Byte Pair Encoding (BPE), a data-driven NLP method. BPE is a bottom-up method that builds a vocabulary for computational text analysis by replacing frequently occurring byte pairs with a single and less frequently used byte [38]. Originally developed for data compression, BPE has found widespread adoption due to its successful application in word segmentation for NLP tasks [39]. The utilization of BPE in music, as implemented in Mel2word has been effectively adopted for melody analysis, classifying folk song families and jazz artists [25, 40]. Similar to its application in language, this method involves creating subwords or *tokens* based on the frequency of consecutive pairs. In other words, it identifies the most frequent consecutive pairs in the melody and merges them into a single unit. As a result, the most frequent pairs are combined using an underscore (‘_’) symbol. Figure 1-(b) illustrates the

basic BPE process and the resulting token outcomes.²

3.3 Word Embedding

Word embedding is a vector representation that captures the meaning and relationships of words by representing them as dense, distributed, and fixed-length vectors based on their context in text. Built on the distributional hypothesis [41], it maps words onto a high-dimensional space, placing similar words close together. In music information retrieval, word embeddings have been used to analyze and model relationships between melodic elements. Specifically, the Word2Vec model [37] has been successfully employed in previous studies to represent notes [42, 43], chords [44, 45] or motifs [46, 47] in a distributed vector space. To capture the semantic analysis of melodic elements, we utilize the Word2Vec model in this study.

3.4 Toward the Substantiality of Melody

In determining the substantiality of music, the court has considered the “distinctive characteristics” of the subject matter as a crucial factor [48]. To evaluate the distinctive features of a melody, we propose two methods drawn from the fields of psychology and NLP: 1) assessing the *salience* of a word or how noticeable it is, and 2) evaluating the *importance* and *uniqueness* of a word or how important and rare it is.

3.4.1 Word Salience

The Tversky ratio is a formula for similarity proposed by Amos Tversky, a cognitive psychologist who suggested that human perceptions and judgments of similarity are based on the number of features two objects have in common and the salience of these features [30]. Tversky’s formula is given by:

$$s(A, B) = \frac{|A \cap B|}{(|A \cap B| + \alpha|A \setminus B| + \beta|B \setminus A|)} \quad (1)$$

where A and B are sets, $|A \cap B|$ is the number of common elements in A and B , $|A \setminus B|$ is the number of elements in A that are not in B , $|B \setminus A|$ is the number of elements in B that are not in A . The parameters α and β adjust the impact of the unique elements of A and B respectively, with higher $s(A, B)$ indicating stronger similarity. In the context of melody, features and elements could refer to components such as note pitch or inter-onset interval.

Since the original Tversky model does not account for the individual salience of specific components, we introduce a modified measure specifically designed to evaluate the significance of individual melodic elements. This adaptation evaluates the significance of each melodic element by considering its prevalence in two melodies and its distribution within each, providing a refined perspective on

their commonality and relative frequency. To evaluate the significance of elements shared between two melodic sequences A , B and an element x of A , we propose a salience measure $TV_{A,B}(x)$. When a_x and b_x represent the counts of element x in sequences A and B respectively, along with the lengths l_A and l_B of the sequences, the formula for $TV_{A,B}(x)$ is given by:

$$TV_{A,B}(x) = \frac{\frac{a_x}{l_A}}{\frac{a_x}{l_A} + \alpha \left(1 - \frac{a_x}{l_A}\right) + \beta \left(1 - \frac{b_x}{l_B}\right)} \quad (2)$$

Here, α and β are coefficients designed to adjust for the lengths of A and B , calculated as $\alpha = \frac{l_A}{l_A + l_B}$ and $\beta = \frac{l_B}{l_A + l_B}$.³ The $TV_{A,B}(x)$ measure evaluates the salience of element x from the perspective of sequence A , taking into account both shared and unique elements. This method allows for a balanced evaluation across sequences, aligning with Tversky’s concept of asymmetrical similarity. By incorporating α and β , the measure provides a nuanced assessment of each element’s salience, considering its frequency within the sequences and the overall sizes of the melodies. This approach ensures a standardized measure, assigning a salience score ranging from 0 (indicating no shared elements) to 1 (indicating fully shared), facilitating equitable comparisons regardless of sequence length.

3.4.2 Word Importance and Uniqueness

TF-IDF is a widely used algorithm in NLP that measures the importance and uniqueness of a term in a document compared to a collection of documents. It takes into account the frequency and rarity of each term in the document and the corpus, respectively. The TF component considers the relevance of a term proportional to its frequency in the document, while the IDF component measures its rarity in the corpus. If a term is frequently used in the corpus, it is considered less representative of a specific document, and if it is rare, it is considered more relevant to a specific document. The TF-IDF value is obtained by multiplying the TF and IDF scores of a term in a document. The formula is as follows:

$$\text{TF-IDF}(t, d) = \text{TF}(t, d) \times \text{IDF}(t) \quad (3)$$

where t represents a term or word in a document, d , $\text{TF}(t, d)$ represents the term frequency of t in d , and $\text{IDF}(t)$ represents the inverse document frequency of t in a collection of documents. In this study, each token in a melody is treated as a single unit of text and the entire melody as a single document. To determine the importance and uniqueness of each melodic element, the TF and IDF scores are utilized as weightings for the word embedding vectors, respectively.

4. EXPERIMENTS

This section describes our empirical investigation of analyzing copyright infringement cases.

³ The constant 1 is derived from $\frac{l_A}{l_A}$ for sequence A and $\frac{l_B}{l_B}$ for sequence B .

² Figures 1-(a) and (b) are sourced from [25]. More details on the Mel2word representation and the BPE process, including the subsequent steps of Dictionary Generation (Section 4.2) and Tokenization (Section 4.3) are found at [25].

4.1 The Dataset

We collected copyright infringement cases from various sources, including previous research and law school databases. We prioritized the data provided by [34], who used a similar sampling approach to [12] and included updated metadata with perceptual data.⁴ Additionally, we extensively reviewed cases from the Music Copyright Infringement Resource (MCIR)⁵ and Lost in Music by Westminster Law School⁶ to compile a comprehensive analysis, aiming to consider as many legal cases as possible. After the landmark case of *Arnstein v. Porter*, which established the concept of substantial similarity, our analysis delved into an extensive repository of legal documents and accompanying materials, up to 314 cases from 1946 to 2023. Following an in-depth review, we excluded cases lacking audio or sheet transcription, those not involving similar musical elements (e.g., licensing, sampling, arrangements, rap lyrics, etc.), and those without relevant expert commentary and opinions for the rulings. As a result, we collected and transcribed into MIDI data on 116 cases (Infringed $N=32$, Denied $N=66$, Settled $N=18$), encompassing 232 songs. We included settlement cases collected in our database; however, for evaluation analysis, we included only settlements with official payments or public records of royalties or credit. In total, we analyzed 108 cases in this study.

4.2 Dictionary Generation

We utilized the Meertens Tune Collection - Folk Song dataset (MTC-FS) to train our BPE model, consisting of over 18,000 monophonic melodies from Dutch sources spanning five centuries [49]. The MTC-FS is one of the largest monophonic datasets, offering a rich repository of melodies that have influenced both classical and modern music. We selected this dataset for its diverse range rooted in oral transmission across generations, providing a strong foundation for analyzing copyright infringement cases across various eras and styles. With BPE applied to the MTC-FS dataset, we initially constructed a base dictionary, which serves as the primary resource for tokenization in subsequent analyses. We constructed the base dictionary using the Mel2word code by [25]⁷, applying BPE to extract words with a minimum frequency of 10 occurrences and limiting the maximum unit size to 11 to prevent redundancy.

4.3 Melody Tokenization

For tokenization, we utilized subsets of the base dictionary to enable tokenization with dictionaries of varying sizes for different levels of segmentation. The subsets were selected based on the most frequent tokens in the base dictionary. For instance, choosing 100 tokens would produce



Figure 2: An example of melody tokenization (Dictionary $N=1000$, pitch feature)

a dictionary with the 100 most frequent entries for tokenization. We relied on statistics from the base dictionary for the maximum length (Mode) and minimum count parameters (Q1, 1st quartile). Consequently, we tokenized melodies from copyright-infringed cases for subsequent analyses using dictionaries of sizes $N=100$, 500, 1000, and *Full-token*⁸, which indicates the maximum number of words available with the parameter settings. Figure 2 illustrates an example of the resulting melody tokenization in our dataset.

4.4 Melody Embedding

To build semantic word embeddings for melodic tokens, we utilized Word2Vec embedding in our experiment. Using the MTC-FS dataset, we tokenized all songs for different dictionary sizes ($N=100$, 500, 1000, and Full) and trained the corresponding Word2Vec models for each size. We used the Gensim module [50], a Python implementation of the Word2Vec⁹, with a dimension size of 512, a window size of 10, a minimum count of 2, and the skip-gram model option, which is known to better represent sparse words [51].

4.5 Similarity Calculation

Cosine similarity is a widely used measure of similarity between two vectors that quantifies the cosine of the angle between the two vectors in a high-dimensional space. In this study, we used the cosine similarity to quantify the similarity between two songs in infringement cases. In order to determine the essential effectiveness of different methods, we opted to calculate melody vectors by averaging as a baseline approach. Although vector summarization through averaging involves a loss of information, it also brings several advantages, such as simplicity in computation, low storage memory requirement, and faster processing speed [52]. Consequently, to generate the melody vectors for each song, we calculated the average of all word vectors for each word unit using the trained Word2Vec model.

4.6 Weight Functions

To assess individual melodic elements, we employed multiple weight functions. These weights are utilized for each token when calculating the average vector of words to derive the final melody vector. For each weight function,

⁴ Except for case 14 (*Vargas v. Pfizer*), as the supplied MIDI data did not contain a melody.

⁵ <https://blogs.law.gwu.edu/mcir/>

⁶ <https://www.lostinmusic.org/>

⁷ <https://github.com/saebuyulpark/Mel2word>

⁸ With dictionary sizes of $N=2399$ for pitch, $N=1184$ for rhythm, and $N=3112$ for both pitch and rhythm

⁹ <https://radimrehurek.com/gensim/>

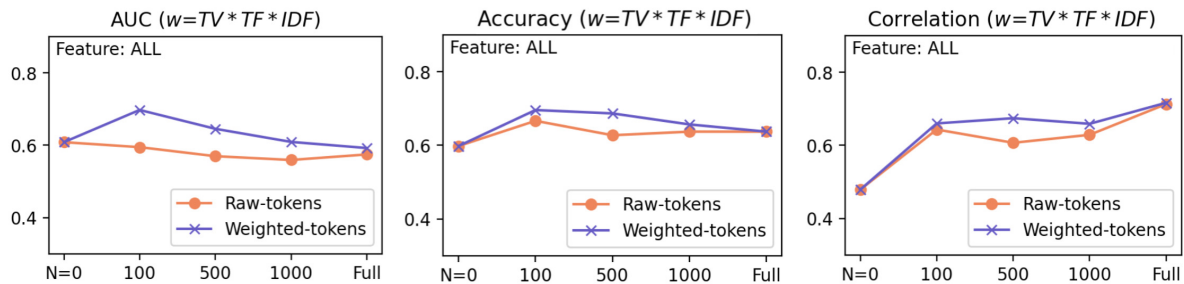


Figure 3: Summary of results with various dictionaries and weight presence.

Weight Function(w)	Foundational Method
TV	Tversky Model
TF	TF-IDF
IDF	TF-IDF
$TF * IDF$	TF-IDF
$TV * TF$	Tversky + TF-IDF
$TV * IDF$	Tversky + TF-IDF
$TV * TF * IDF$	Tversky + TF-IDF

Table 1: Summary of Weight Functions

we defined specific parameters, as summarized in Table 1. We calculated term frequency (TF) and inverse document frequency (IDF) values using the *TfidfVectorizer* module from the sklearn library¹⁰ with the default settings. We also computed the Tversky value (TV) using the formula described in Section 3.4.1 (Equation 2). To avoid zero-multiplication for hybrid variables (e.g., $TV * TF * IDF$), we added one to each variable and multiplied it by the subsequent value. Finally, all weights were experimented with various normalization methods.

4.7 Evaluation

We conducted three types of evaluations, following the previous work [32, 33, 34]. First, we assessed how well the similarities evaluated by the algorithm corresponded to the court’s decision. To measure this, we computed the Area Under the ROC Curve (AUC), a commonly used method to evaluate binary classification performance. Second, we utilized AUC to determine TPR and FPR at different thresholds, identifying the threshold with the highest accuracy (ACC). Finally, we measured how well the similarities correlated with human perceptual data provided by [34]¹¹, for which we computed the Pearson coefficient only for the subset of songs with perceptual data available.

5. RESULTS

5.1 Overall Result

Figure 3 presents an overview of the results considering different dictionaries and the presence of weights, with combined feature (Pitch + Rhythm) and $TV * TF * IDF$

weights achieving the highest scores. While the tokenization method has a minor impact on AUC and ACC metrics, it notably influences the correlation with perceptual data, showing better performance across all dictionary sizes. Additionally, the adoption of weights generally enhances performance across most cases (except for morpheme-level and Full-level).

Regarding measures related to legal decisions (AUC and ACC), as previously discussed about performance limits [34], once again, we found that the proposed method was more effective in correlating with perception than with court decisions. This is likely due to courts considering various factors such as lyrics, arrangement, and other musical elements, as well as the worthiness of a melody to be protected (e.g., *Intersong-USA v. CBS*). Additionally, they consider the possibility of subconscious copying (e.g., *Francis Day & Hunter v. Bron*), and proof of access to the original work (e.g., *Ellis v. Diffie*), even when similarities between melodies exist. Since our study targeted all possible cases involving melody, there may be various confounding variables.

Interestingly, we noticed that the weight function underperformed when analyzing melodies at the morpheme-level ($N = 0$), possibly due to the high number of randomly shared features at this level. Performance also decreased at the Full-level, likely because longer words led to a decrease in shared features. Additionally, we found that applying all weights multiplied by the Tversky model improved performance, while the default TF and IDF weights tended to reduce performance. This result supports the basic assumption of Tversky’s model that we perceive similarity based on how many features are shared, which is consistent with previous research [53, 12, 31] showing a strong association of the Tversky model with infringement decisions and perceptual similarity.

5.2 Comparison Results with Previous Studies

Table 2 compares evaluations conducted on subsamples of 17 ($N=17$)¹² and 39 ($N=39$)¹³ cases each, facilitating comparison with existing literature. As observed, our method performed remarkably well, achieving the highest scores for both sets.¹⁴ These subsets consist of cases with

¹⁰ <https://scikit-learn.org/>

¹¹ This data consists of a similarity scale ranging from 0 to 5 points, where 0 represents dissimilarity and 5 represents similarity, available at: <https://github.com/comp-music-lab/music-copyright-expanded>

¹² Based on [33], which includes 14 songs from [32].

¹³ While [34] included 40 cases; we analyzed 39, excluding *Vargas v. Pfizer* due to the absence of melody.

¹⁴ Pitch feature, $N=100$, with quantile Gaussian normalization for 17 cases; pitch + rhythm, $N=100$ (AUC) and Full (ACC) with quantile Gaus-

Cases		Savage [32]	Yuan1 [33]	Yuan2 [34]	Proposed
N=17	AUC	0.69	0.61	0.61	0.94
	ACC	0.80	0.71	0.71	0.94
N=39	AUC	N/A	N/A	0.73	0.79
	ACC	N/A	N/A	0.75	0.79

Table 2: Comparison Results with Previous Studies

a significant indication of melodic similarity from the outset, making them subjects of a number of previous studies [12, 31, 32, 33, 34]. Therefore, they exhibited effective discrimination based solely on the melody itself, compared to our overall findings. Given that our study examined the entire melodies obtained from the archive, we anticipate that further investigation focusing on specific parts or cases emphasizing the melody will yield even more intriguing results.

5.3 Exploratory Result Analysis

Beyond the performance, the strength of our method lies in its ability to numerically represent the characteristics of each melodic element within a song. For example, Figure 4 illustrates a copyright infringement case, *Three Boys Music v. Michael Bolton*, where distinctive and shared melodic features are quantified using TV , TF , IDF , and $TV*TF*IDF$. In this manner, by examining these melodies, we can observe the numerical values of their importance, uniqueness, and the degree to which they are shared for each melodic element. Moreover, this can play a crucial role when melodies are tokenized into more meaningful units, potentially enhancing their interpretability. For example, Figure 5 presents a cross-scape plot visualization, which provides a hierarchical analysis of the similarities between two songs, indicating where and how they are similar [54]. The left side represents the infringed case (*Three Boys Music v. Michael Bolton*), while the right side represents the denied case (*Baxter v. MCA, Inc.*). On the left, (a) depicts the morpheme-level, while on the right, (b) showcases the token-level melody with weighting applied.¹⁵ As observed, at the morpheme level, segmentation of each note leads to overall similarity across all parts due to the frequently shared elements at the note level. However, in the weighted tokenized songs, certain crucial phrases in the infringed case stand out notably darker (i.e., more similar). This visually demonstrates how our approach highlights specific parts that contribute to a stronger similarity between two pieces of music. In this way, by providing a quantitative method to identify the individual characteristics of melody elements, our research can be of significant help in practical applications such as legal analysis, as well as various fields of music research.

sian normalization for 39 cases.

¹⁵ The original plot was modified to compare song similarities using word vectors. Details and base code for the cross-scape plot are at [54] and https://github.com/saebuyulpark/cross_scapeplot.

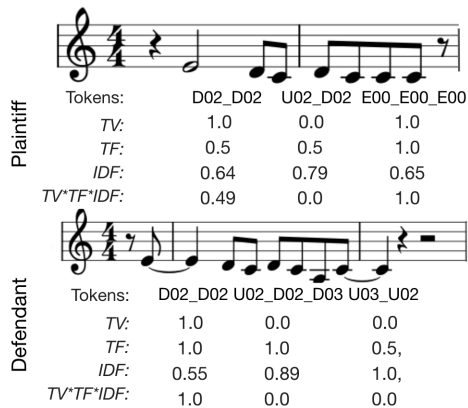


Figure 4: An example of melody weighting values (*Three Boys Music v. Michael Bolton*, N=100, pitch feature)

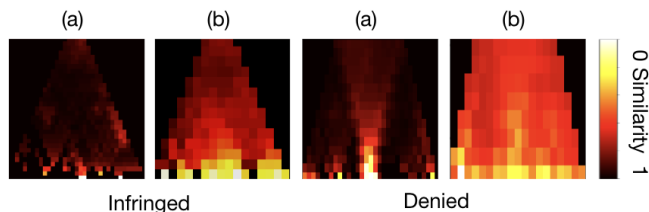


Figure 5: Cross-scape plots: (a) Word2Vec at morpheme-level, (b) Word2Vec at token-level with $TV * TF * IDF$.

6. CONCLUSION

In this study, we employed natural language processing (NLP) techniques to objectively grasp the substantial similarity of melodies, thereby making notable contributions in several key areas: First, after an extensive review of legal documents, we compiled one of the most extensive public datasets, the Music Copyright Infringement Collection (MCIC).¹⁶ Although it is not big data, this dataset is significant given the limited number of legal cases, as it includes MIDI transcriptions, sheet music, and metadata on legal issues and decisions, forming the crucial groundwork for future studies on music similarity and copyright issues. Second, we encoded melodies into word-like units using Mel2word to analyze melodic similarity for the music plagiarism study. This approach extends semantic analysis beyond the note- or n-gram level, surpassing conventional analysis methods. Third, we introduced the modified-Tversky measure to evaluate the salience of each melodic element. Derived from a prominent psychological theory, this refined measure offers potential applications beyond music, exhibiting its general versatility. Moreover, by incorporating traditional NLP-based weighting algorithms, we conducted an in-depth analysis of individual features to comprehensively grasp substantial similarity. Thus, by integrating computational methods, psychological models, data-driven techniques, and rule-based approaches, we performed a detailed exploration of melodic similarity.

¹⁶ <https://github.com/saebuyulpark/MCIC>. This site includes supplementary materials with comprehensive experimental details, including transcriptions, statistics, normalization methods, additional results, and full and sub-dataset lists.

7. ETHICS STATEMENT

This research adheres to ethical guidelines, ensuring data integrity and confidentiality. All sources are credited, and only publicly available data were used.

8. REFERENCES

- [1] A. Keyt, "An improved framework for music plagiarism litigation," *Cal. L. Rev.*, vol. 76, 1988.
- [2] S. J. Jones, "Music copyright in theory and practice: An improved approach for determining substantial similarity," *Duq. L. Rev.*, vol. 31, 1992.
- [3] A. B. Cohen, "Masking copyright decisionmaking: The meaninglessness of substantial similarity," *UC DAVIS l. rev.*, vol. 20, 1986.
- [4] M. F. Sitzer, "Copyright infringement actions: the proper role for audience reactions in determining substantial similarity," *S. Cal. L. Rev.*, vol. 54, 1980.
- [5] C. A. Tschider, "Automating music similarity analysis in "sound-alike" copyright infringement cases," *Entertainment, Arts and Sports Law Journal*, vol. 25, no. 2, 2014.
- [6] S. N. Hamilton, D. Majury, and D. Moore, *Sensing Law*. Taylor & Francis, 2016.
- [7] I. Stav, "Musical plagiarism: A true challenge for the copyright law," *DePaul J. Art Tech. & Intell. Prop. L.*, vol. 25, 2014.
- [8] R. De Prisco, D. Malandrino, G. Zaccagnino, and R. Zaccagnino, "Fuzzy vectorial-based similarity detection of music plagiarism," in *FUZZ-IEEE*. IEEE, 2017.
- [9] E. Selfridge-Field, "Conceptual and representational issues in melodic comparison," *Melodic similarity, concepts, procedures, and applications*, 1998.
- [10] R. Typke, "Music retrieval based on melodic similarity," *Ph.D thesis*, 2007.
- [11] J. P. Fishman, "Music as a matter of law," *Harv. L. Rev.*, vol. 131, p. 1861, 2017.
- [12] D. Müllensiefen and M. Pendzich, "Court decisions on music plagiarism and the predictive value of similarity algorithms," *Musicae Scientiae*, vol. 13, no. 1_suppl, 2009.
- [13] D. Müllensiefen and K. Frieler, "Cognitive adequacy in the measurement of melodic similarity: Algorithmic vs. human judgments," *Computing in Musicology*, vol. 13, no. 2003, 2004.
- [14] V. Velardo, M. Vallati, and S. Jan, "Symbolic melodic similarity: State of the art and future challenges," *Computer Music Journal*, vol. 40, no. 2, 2016.
- [15] J. S. Downie, "Music information retrieval," *Annual Review of Information Science and Technology*, vol. 37, no. 1, 2003.
- [16] L.-J. Stark and T. J. Perfect, "The effects of repeated idea elaboration on unconscious plagiarism," *Memory & cognition*, vol. 36, no. 1, pp. 65–73, 2008.
- [17] B. Challis, "The song remains the same: A review of the legalities of music sampling," *Wipo Magazine*, vol. 6, 2009.
- [18] A. D. Patel, "Language, music, syntax and the brain," *Nature neuroscience*, vol. 6, no. 7, 2003.
- [19] B. Maess, S. Koelsch, T. C. Gunter, and A. D. Friederici, "Musical syntax is processed in broca's area: an meg study," *Nature neuroscience*, vol. 4, no. 5, 2001.
- [20] S. Koelsch, E. Kasper, D. Sammler, K. Schulze, T. Gunter, and A. D. Friederici, "Music, language and meaning: brain signatures of semantic processing," *Nature neuroscience*, vol. 7, no. 3, 2004.
- [21] D. Conklin, "Representation and discovery of vertical patterns in music," in *International Conference on Music and Artificial Intelligence*. Springer, 2002.
- [22] J. Wołkowicz, Z. Kulka, and V. Kešelj, "N-gram-based approach to composer recognition," *Archives of Acoustics*, vol. 33, no. 1, 2008.
- [23] M. Mongeau and D. Sankoff, "Comparison of musical sequences," *Computers and the Humanities*, vol. 24, no. 3, 1990.
- [24] T. Crawford, "String matching techniques for musical similarity and melodic recognition," *Computing in musicology*, vol. 11, 1998.
- [25] S. Park, E. Choi, J. Kim, and J. Nam, "Mel2word: A text-based melody representation for symbolic music analysis," *Music & Science*, vol. 7, 2024.
- [26] C. Dittmar, K. F. Hildebrand, D. Gärtner, M. Wings, F. Müller, and P. Aichroth, "Audio forensics meets music information retrieval—a toolbox for inspection of music plagiarism," in *2012 Proceedings of the 20th European signal processing conference*. IEEE, 2012, pp. 1249–1253.
- [27] K. Suneja and M. Bansal, "Comparison of time series similarity measures for plagiarism detection in music," in *Annual IEEE India Conference*, 2015.
- [28] S. De, I. Roy, T. Prabhakar, K. Suneja, S. Chaudhuri, R. Singh, and B. Raj, "Plagiarism detection in polyphonic music using monaural signal separation," *arXiv preprint arXiv:1503.00022*, 2015.
- [29] N. Borkar, S. Patre, R. S. Khalsa, R. Kawale, and P. Chakurkar, "Music plagiarism detection using audio fingerprinting and segment matching," in *Smart Technologies, Communication and Robotics*, 2021.

- [30] A. Tversky, "Features of similarity." *Psychological review*, vol. 84, no. 4, 1977.
- [31] A. Wolf, D. Müllensiefen *et al.*, "The perception of similarity in court cases of melodic plagiarism and a review of measures of melodic similarity," in *International Conference of Students of Systematic Musicology*, 2011.
- [32] P. E. Savage, C. Cronin, D. Müllensiefen, and Q. D. Atkinson, "Quantitative evaluation of music copyright infringement," in *Proceedings of the 8th International Workshop on Folk Music Analysis*. Thessaloniki Greece, 2018.
- [33] Y. Yuan, S. Oishi, C. Cronin, D. Müllensiefen, Q. Atkinson, S. Fujii, and P. E. Savage, "Perceptual vs. automated judgments of music copyright infringement," in *Proceedings of the 21st International Society for Music Information Retrieval Conference*, 2020.
- [34] Y. Yuan, C. Cronin, D. Müllensiefen, S. Fujii, and P. E. Savage, "Perceptual and automated estimates of infringement in 40 music copyright cases," *Transactions of the International Society for Music Information Retrieval*, vol. 6, no. 1, 2023.
- [35] K. Park, S. Baek, J. Jeon, and Y.-S. Jeong, "Music plagiarism detection based on siamese cnn," *Human-centric Computing and Information Sciences*, 2022.
- [36] D. Malandrino, R. De Prisco, M. Ianulardo, and R. Zaccagnino, "An adaptive meta-heuristic for music plagiarism detection based on text similarity and clustering," *Data Mining and Knowledge Discovery*, 2022.
- [37] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013.
- [38] P. Gage, "A new algorithm for data compression," *C Users Journal*, vol. 12, no. 2, 1994.
- [39] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, 2016.
- [40] S. Park and J. Nam, "The language of jazz: A natural language processing-based analysis of the patterns and vocabulary of jazz solo improvisation," in *17th International Conference on Music Perception and Cognition*, 2023.
- [41] Z. S. Harris, "Distributional structure," *Word*, vol. 10, no. 2-3, 1954.
- [42] C.-H. Chuan, K. Agres, and D. Herremans, "From context to concept: exploring semantic relationships in music with word2vec," *Neural Computing and Applications*, vol. 32, no. 4, 2020.
- [43] D. Herremans and C.-H. Chuan, "Modeling musical context with word2vec," in *Proceedings of the First International Conference on Deep Learning and Music*, 2017.
- [44] C.-Z. A. Huang, D. Duvenaud, and K. Z. Gajos, "Chordriple: Recommending chords to help novice composers go beyond the ordinary," in *Proceedings of the 21st International Conference on Intelligent User Interfaces*, 2016.
- [45] S. Madjiheurem, L. Qu, and C. Walder, "Chord2vec: Learning musical chord embeddings," in *Proceedings of the constructive machine learning workshop at 30th conference on neural information processing systems*, 2016.
- [46] A. A. Alvarez and F. Gómez-Martin, "Distributed vector representations of folksong motifs," in *International Conference on Mathematics and Computation in Music*, 2019.
- [47] T. Hirai and S. Sawada, "Melody2vec: Distributed representations of melodic phrases based on melody segmentation," *Journal of Information Processing*, vol. 27, 2019.
- [48] R. P. Smith, "Arrangements and editions of public domain music: Originally in a finite system," *Case W. Res. L. Rev.*, vol. 34, 1983.
- [49] P. Van Kranenburg and M. De Bruin, "The meertens tune collections: Mtc-fs-inst 2.0," *Meertens Online Reports*, vol. 2019, no. 1, 2019.
- [50] R. Řehůřek and P. Sojka, "Software Framework for Topic Modelling with Large Corpora," in *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, 2010.
- [51] A. J. Landgraf and J. Bellay, "Word2vec skip-gram with negative sampling is a weighted logistic pca," *arXiv preprint arXiv:1705.09755*, 2017.
- [52] M. A. Kharazmi and M. Z. Kharazmi, "Text coherence new method using word2vec sentence vectors and most likely n-grams," in *2017 3rd Iranian Conference on Intelligent Systems and Signal Processing*. IEEE, 2017, pp. 105–109.
- [53] D. Müllensiefen and K. Frieler, "Measuring melodic similarity: Human vs. algorithmic judgments," in *Proceedings of the Conference on Interdisciplinary Musicology*, 2004.
- [54] S. Park, T. Kwon, J. Lee, J. Kim, and J. Nam, "A cross-scape plot representation for visualizing symbolic melodic similarity," in *Proceedings of the 20th International Society for Music Information Retrieval Conference*, 2019.

ROBUST LOSSY AUDIO COMPRESSION IDENTIFICATION

Hendrik Vincent Koops Gianluca Micchi Elio Quinton

Music and Audio Machine Learning Lab,

Universal Music Group, London, UK

{vincent.koops, gianluca.micchi, elio.quinton}@umusic.com

ABSTRACT

Previous research contributions on blind lossy compression identification report near perfect performance metrics on their test set, across a variety of codecs and bit rates. However, we show that such results can be deceptive and may not accurately represent true ability of the system to tackle the task at hand. In this article, we present an investigation into the robustness and generalisation capability of a lossy audio identification model. Our contributions are as follows. (1) We show the lack of robustness to codec parameter variations of a model equivalent to prior art. In particular, when naively training a lossy compression detection model on a dataset of music recordings processed with a range of codecs and their lossless counterparts, we obtain near perfect performance metrics on the held-out test set, but severely degraded performance on lossy tracks produced with codec parameters not seen in training. (2) We propose and show the effectiveness of an improved training strategy to significantly increase the robustness and generalisation capability of the model beyond codec configurations seen during training. Namely we apply a random mask to the input spectrogram to encourage the model not to rely solely on the training set's codec cut-off frequency.

1. INTRODUCTION

Audio codecs can be roughly categorized into two categories: *lossless* and *lossy*. *Lossless* means that an exact preservation of the signal is guaranteed by the codec. In other words, the signal resulting from encoding and decoding is exactly identical to the original. In contrast, *lossy* encoding means that some of the signal is lost in the encoding and decoding process. In other words, the signal resulting from encoding and decoding is not exactly identical to the original signal.

Popular lossy audio codecs like MP3 [1], Ogg Vorbis [2] or AAC [3] are known as "perceptual" codecs because they rely on models of human auditory cognition to prioritise the deletion of parts of the audio signal that have the least

perceptual impact on human listeners. Despite the signal degradation that they result in, perceptual lossy codecs can achieve much greater compression ratios than lossless codecs, and are therefore well suited for applications where data bandwidth is limited. For example, they have been instrumental in enabling music streaming over networks with limited bandwidth.

Digital audio codecs are readily available and are integrated into many widespread professional and consumer tools such as Digital Audio Workstations, software libraries, digital music players etc., which make converting an audio file from one format to another nowadays extremely easy and accessible to anyone. As a result it is easy to mistakenly encode a source audio signal with a lossy codec, which degrades the signal, and then decode it back into a lossless file container. This process may create the illusion that a lossless file container (e.g. WAV) contains unimpaired audio when it does in fact contain lossy-compressed audio.

Guaranteeing audio integrity is essential in many applied scenarios such as large scale music distribution or archiving. Because the aforementioned case of lossy audio disguised as a lossless file would violate this guarantee, there is a need to automatically detect such occurrences. Identification of audio that has been compressed with a lossy codec is a valuable component of quality assurance processes, which form an important part of many modern musical audio content pipelines.

Contributions. In this paper, we present an investigation into the robustness and generalisation capability of a lossy audio identification model. We show that when we naively train a lossy compression detection model on a dataset of music recordings processed with a range of codecs and their lossless counterparts, we obtain near perfect performance metrics on the held-out test set. However, we obtain severely degraded performance on lossy tracks produced with codec parameters not seen in training. We also propose a new training schema in which we randomly mask the input spectrogram to improve the model's robustness. We show that our approach significantly increases the robustness and generalisation capability of the model beyond codec configurations seen during training.

2. BACKGROUND

In the following sections, we will first provide a high level overview of lossy audio codecs (Section 2.1). Next, in Sec-



© F. Author, S. Author, and T. Author. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

Attribution: F. Author, S. Author, and T. Author, "Robust lossy audio compression identification", in *Proc. of the 25th Int. Society for Music Information Retrieval Conf.*, San Francisco, United States, 2024.

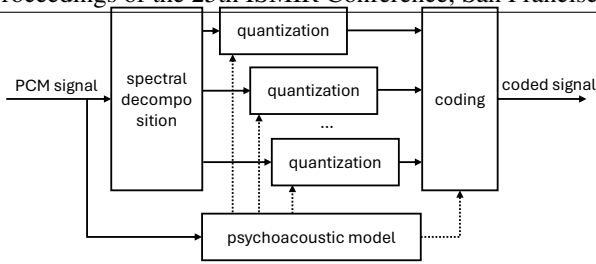


Figure 1. Basic block diagram of a perceptual audio coder. After spectral decomposition, a psychoacoustic model informs the quantization of individual spectral components.

tion 2.2, related work on lossy audio identification is discussed. Finally, in Section 2.3, we briefly present related work in MIR on robustness evaluation.

2.1 Lossy Codecs

Figure 1 shows a basic block diagram with the common modules of a perceptual audio coder. The process of encoding an audio signal with a lossy codec is commonly as follows. First, the original uncompressed (often pulse code modulated - PCM) signal is transformed into a time-frequency representation. This is typically done using a modified discrete cosine transform (MDCT), but many other transforms have been proposed [4]. Commonly used signal block for the spectral decomposition are between 2ms and 50ms. The components of the spectral decomposition are then individually quantized. The quantization of the spectral components is controlled by a psychoacoustic model that describes the time and frequency masking properties of the human auditory system. Auditory masking is a process where one sound (maskee) becomes inaudible in the presence of another sound (mask) [5].

Auditory masking can occur in the time domain (temporal masking) or in the frequency domain (frequency masking). The quantization controlled by the psychoacoustic model effectively controls which spectral coefficients will be removed, resulting in spectral band rupture and holes in spectrograms, as observed in Figure 2. After quantization, Huffman coding (or some other form of entropy coding) is applied to remove or reduce the redundancy in the signal [6]. The bit rate of a codec effectively controls both the size and the perceptual quality of the audio. A low bit rate (like 128 kbps) will produce a small storage footprint, but generally worse perceptual quality compared to a higher bit rate (like 320 kbps). For more detail on audio codecs and standards, we refer to [4].

2.2 Lossy Compression Identification

In previous research, multiple blind lossy compression identification models have been proposed. These can broadly be categorized into two approaches. One approach is to estimate codec parameters from the audio signal, to determine factors such as the decoder framing grid, filter bank parameters and/or quantization information. This

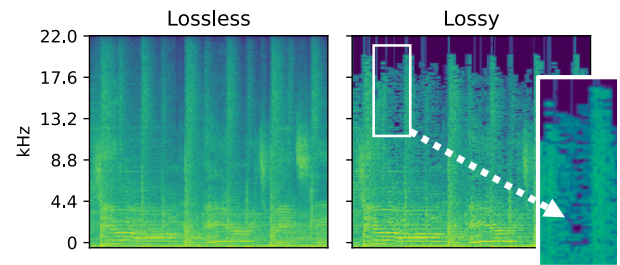


Figure 2. Spectrograms of examples of a lossless (left) and lossy version of the same audio excerpt (right). The latter is compressed with the LIBFDK_AAC codec at 128 kbps bit rate. The version on the right shows the hallmarks of lossy compression: removal of FFT coefficients, holes in the spectrum, and general loss of higher frequency content.

type of approach has been successfully applied for individual codecs like AAC [7], MP3 [8–10]. Although this type of approach can be very effective, it is computationally very expensive, especially when multiple codecs are considered.

The second method utilizes audio quality measures to determine whether the audio is lossy. One effect of lossy audio compression is the introduction of “holes” in the spectrogram, especially right after louder transients. This is the result of the fact that spectral coefficients can be removed when they are perceptually masked by other coefficients. Therefore, most approaches present some form of “hole-detection“, such as estimating the number of inactive spectral coefficients (e.g. [9, 11]) or computing spectral fluctuations [12–15].

In [16], Hennequin et al. presented a method for detecting lossy compression based on a convolutional neural network CNN applied to audio spectrograms. Similarly, Seichter et al. in [17] also proposed a CNN approach for AAC encoding detection and bit rate estimation. All research contributions on lossy compression identification almost uniformly report near-perfect performance metrics on their test set, across a variety of codecs and bit rates.

However, most codecs can be configured with parameters other than the bitrate too, such as a cutoff frequency that controls the amount of higher frequencies that will be preserved. AAC for example has a default cutoff frequency of around 17kHz [18] for constant bit rates of 96 kbps per channel and above, which means that the bandwidth of the encoder is set to 0 - 17kHz. None of the previous research explores what happens when this parameter is changed.

In this paper we show that a model naively trained on default parameters may not efficiently learn to discriminate lossy audio encoded with different parameters and we analyse what happens when varying the cutoff frequency as an example. Therefore, the good results previously reported must be taken with a pinch of salt.

2.3 Robustness Evaluation in Music Information Retrieval

Several studies in music information retrieval have shown that models can seemingly achieve very high evaluation performance, while further research reveals that what those models have learned is some confound with the ground truth dataset [19]. For example, in a research into the robustness of genre classification models, Sturm showed that although these systems might have high mean classification accuracies, they don't actually reflect the underlying properties of the genre [20]. Furthermore, it is shown that by filtering the audio signal in a minimal way, the models produce radically different genre predictions. For a larger overview of music adversaries in music information retrieval research, we refer to [19]. Bob Sturm in [21] introduced the term "horse"¹ to refer to system appearing capable of achieving high evaluation performance, but actually working by using irrelevant characteristics (confounds), and therefore not actually addressing the problem it appears to be solving.

3. METHOD

In the following sections, we will first describe our model setup (in Section 3.1), then our dataset (in Section 3.2) and finally our proposed evaluation methods (in Section 3.3).

3.1 Network Architecture

For the detection of lossy audio we propose a model (visualized in Figure 3) that can be divided into four parts: a spectrogram + random mask module, 4 convolutional blocks, an lstm block and a classification head made of a single dense layer. The architecture is partly inspired by prior work by Hennequin et al. in [16] and Seichter et al. in [17]. In the following sections, we will describe each part in detail.

The model takes as input 2 seconds of raw monophonic audio signal sampled at 44.1 kHz, which is passed to a *torchaudio* spectrogram layer that produces a magnitude spectrogram with 1024 FFT coefficients [22].

Random mask. A random mask is optionally applied to the input spectrogram. This is achieved by uniformly randomly sampling a cutoff frequency between 14 kHz and the Nyquist frequency of the sample, and nulling all *fft* coefficients above that frequency by setting them to the minimum of the input spectrogram. A similar approach called Specaugment was proposed by Park et al. in [23].

In our first experiment (as described in Section 4.1) this layer is not used, and the spectrogram is directly fed to the convolutional blocks. However, in the second experiment (as described in Section 4.2), we use this random mask layer with a different random cutoff frequency for every training example.

CNN. Each of the CNN blocks consist of four layers: a 2D convolutional layer with a kernel size of (3,3), a ReLU layer, a batch normalization layer and a 2D max-pooling

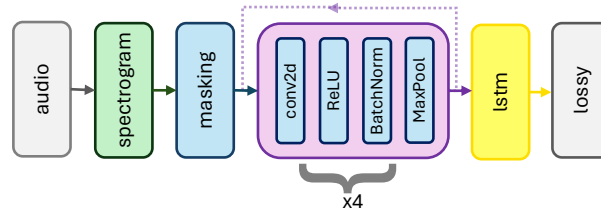


Figure 3. Proposed model for the detection of lossy audio Our model takes as input 2 seconds of audio, which is passed to a torchaudio spectrogram layer (in green). Depending on the experiment, the spectrogram is then passed to a masking layer (in blue), which simulates low-pass filtering. The spectrogram is then passed to four convolutional modules (in pink). We use a bi-directional LSTM (in yellow) for dimensionality reduction. We classify the audio into lossy or lossless in the final model head.

layer. The max pooling size for each block is (2,2), with the exception of the last block, which is (2,4).

LSTM. We connect the CNN to a *long short-term memory* (LSTM) block for two reasons. Firstly, we want to exploit possible sequential properties of the CNN output, and secondly, for dimensionality reduction for the last (dense) part of the network. We use a bidirectional LSTM with two layers of size 128.

Classification head. Our model's lossy/lossless classification head is connected to the LSTM output with a dense layer of size 256 (2x 128 because our LSTM is bi-directional). The classification head has a softmax activation and 2 outputs that model the probability of the example being lossless or lossy.

Training. We back-propagate our model on the binary cross entropy of the classification head and the ground truth. For each audio track, we take a 2-second random crop at training time.

3.2 Datasets

For our experiments, we sample 10k tracks of lossless 16 bit, 44.1kHz WAV files from a large private library of commercial music. From these tracks, we create two datasets.

3.2.1 DS1.

For the first dataset we encode each track with a codec randomly chosen among LIBMP3LAME (MP3), LIBFDK_AAC (AAC) and LIBVORBIS (OGG), with bit rate also randomly chosen among 128, 256 and 320 kbps. Each encoded file is then decoded back into a 16bit, 44.1 kHz WAV file that is used as input to the model. All the encoding/decoding is done using *ffmpeg* [24]. Between lossless and lossy tracks, the dataset comprises of 20k tracks.

3.2.2 DS2.

For the second dataset, we use the same original tracks as were used to create DS1. We also use the same codec parameters, but vary the cutoff frequency of the codecs,

¹ A nod to the Clever Hans horse, see https://en.wikipedia.org/wiki/Clever_Hans

Codec Bit rate	LIBFDK_AAC			LIBVORBIS			LIBMP3LAME			Lossless	Mean
	128k	256k	320k	128k	256k	320k	128k	256k	320k		
DS1	100.0	98.91	100.0	100.0	100.0	100.0	100.0	100.0	98.37	99.88	99.79
DS2	31.38	28.96	24.74	98.91	93.16	86.7	80.63	68.45	60.87	99.88	81.85

Table 1. Accuracy of evaluating the model without random mask on a dataset without (DS1) and with cutoff frequency variations (DS2). Varying the cutoff parameter in the codec greatly degrades model results.

choosing among 14, 16, 18 and 20 kHz. DS1 and DS2, therefore, differ only on the lossy versions obtained for each track. We use the same random 70/10/20 split for training/validation/testing for both datasets. All our experiments are run using DS1 for training and validation. Evaluation is done on DS1 (cf. Sec. 4.1) or DS2 (Sec. 4.2).

3.3 Evaluation

We evaluate the performance of our lossy/lossless detection model in three ways. Firstly, we provide quantitative evaluation and report the model accuracy. Secondly, we inspect saliency maps of the CNN blocks of our model to gain qualitative insight into what signal properties the model is sensitive to. Finally, we also inspect the errors of our model in detail to help us assess the effectiveness our proposed method to make our model more robust, and identify avenues for future work.

4. EXPERIMENTS & RESULTS

In this section we first describe our experiments and report our results on a naively trained lossy/lossless audio detection model (Section 4.1). After an analysis of our results, we report on a more robust variation of our model in Section 4.2, and an analysis of errors in Section 4.3.

4.1 Experiment 1: Naive Model Training

In our first experiment, we train our model on DS1. For each track in our test set, we extract 2-second windows of raw audio with 50% overlap. For each window, we perform a forward pass through our trained network, and collect the output of the classification head. We take the mean of all windowed local model outputs as the global output per track.

4.1.1 Results

In line with previous research (e.g. [16, 17]), we find near-perfect performance on lossy/lossless audio detection of audio with default codec settings. The top row of Table 1 shows the results broken down by codec and bit rate for DS1. We obtain near-perfect results per bit rate/code combination. On average, we obtain 99.79% accuracy across all codecs and lossless files.

However, if we slightly tweak the codec parameters at test time (i.e. we test our model on DS2) the performance drops significantly. The bottom row of Table 1 shows the results of evaluating the model on the dataset with cutoff frequency variations. The results show much poorer results for the lossy tracks across all codec/bit rate combinations. Specifically, we find a big drop in accuracy of around 70

percentage points for the LIBFDK_AAC codec and around 30 percentage points for the MP3 codec. The LIBVORBIS is less impacted, but is still significantly impacted by around 10 percentage points.

4.1.2 Analysis

To get a better sense of what our model has learned, we turn towards a feature analysis of the CNN part of the network. When inspecting the spectrogram of a potentially lossy file with the naked eye, one of the most striking aspects is the nulling of coefficients, resulting in “holes” in the spectrogram. We expected the convolutional part of the network to pick up on those, and to design features that capture this phenomenon.

However, when we visualize saliency maps from our network, we find a different pattern (see Figure 4, top row). It seems that the model is more concerned with the cutoff frequency of the lossy audio than with the holes in the spectrogram. Although the cutoff frequency is a useful feature, by itself it is neither necessary nor sufficient to determine whether an audio signal has been encoded with a lossy codec.

Table 2 shows the results of the model per cutoff frequency, in the columns marked with ‘No’. Here again we see that most cutoff frequency variations are severely underperforming when compared to the previous test dataset.

The model performs best at a cutoff frequency of 16 kHz. This can be explained by the fact that this is the default cutoff frequency of LIBVORBIS, which is therefore not affected by this transformation. In the next section, we adapt the model to be robust against this cutoff effect.

4.2 Experiment 2: Creating a Robust Model

In order to increase the model’s robustness against the lossy codec’s cutoff frequency, we present a second experiment where we randomly mask the upper end of the spectrum. The mask, defined in 3.1, is applied to all input files.

The application of this random mask is intended to force the model not to solely rely on the codec cutoff frequency to make a prediction, and instead also rely on other signal degradations included by codecs, such as “holes” in the spectrogram. A fixed mask at a specific cutoff frequency would have meant throwing away the information given by the spectral rolloff entirely and this would have been suboptimal in the opposite direction.

We train this model on DS1 and evaluate on DS2.

Cutoff	LIBFDK_AAC		LIBVORBIS		LIBMP3LAME		128k		256k		320k		MEAN	
	No	Mask	No	Mask	No	Mask	No	Mask	No	Mask	No	Mask	No	Mask
14 kHz	24.1	81.0	100.0	100.0	76.9	100.0	90.4	82.6	53.7	100.0	49.3	97.8	65.9	93.3
16 kHz	83.9	98.4	100.0	100.0	98.6	100.0	88.2	100.0	97.0	97.7	98.6	100.0	94.6	99.5
18 kHz	0.7	86.7	66.9	100.0	25.3	100.0	50.0	96.2	28.2	94.1	12.4	94.8	29.5	95.6
20 kHz	11.1	96.5	100.0	100.0	82.9	100.0	46.2	100.0	76.1	97.2	70.2	100.0	65.1	98.9
MEAN	28.3	90.1	92.9	100.0	70.1	100.0	70.1	93.6	64.1	97.9	57.3	98.8	63.7	96.8

Table 2. Accuracy (in percentage points) of evaluating our models without (No) and with (Mask) random mask on DS2, per codec and bit rate, for varying cutoff frequency. Lossless accuracy is 99.9% for No and 99.8% for Mask.

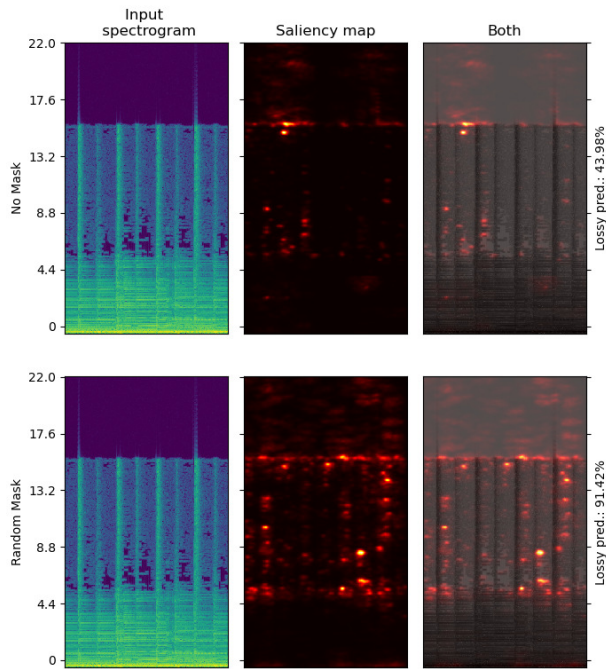


Figure 4. Saliency maps from exposing a model trained without (top) and with (bottom) random mask to lossy audio. The model with random mask shows more activation in the holes of the spectrogram without losing any of the activations at the cutoff frequency.

4.2.1 Results

Table 2 shows the results obtained for the model trained with the random mask on DS1 and evaluated on DS2. We observe good classification results on average, 96.8% on lossy files and 99.8% on lossless files. Overall, we obtain 98.4% lossy/lossless classification accuracy across the entire test dataset. Comparing with the naive model, the accuracy on DS2 improves significantly across the board.

With the mean classification accuracy at 90% or above in all conditions (last column of the table), this model is broadly robust against cutoff frequency variations. It is interesting to note that performance on the AAC codec is comparatively lower than on other codecs. This result suggests that the AAC codec is more challenging to detect, and warrants further investigation, which we leave for future work. We hypothesise it may be due to the AAC codec producing less artefacts in the magnitude spectrogram.

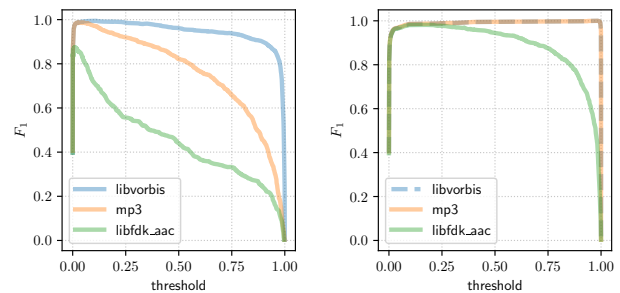


Figure 5. F_1 -score for varying thresholds, evaluated on DS2. Each line analyses the subset made of lossless files as negatives and the specified codec as positives; files encoded with different codecs are discarded. Left: model without random mask; Right: model with random mask.

In Fig. 5, for both the model without random mask (cf. Section 4.1) and the model with (cf. Section 4.2) we plot the F_1 score (i.e., the harmonic mean between precision and recall) as a function of the threshold of the binary classification prediction. The F_1 -score for the model without the random mask peaks at very low values of the threshold and then decays for increasing threshold at a rate that highly depends on the codec analysed.

This suggests three conclusions: (1) There are a number of test set files that yield a prediction $p(x)$ in the central region $0.1 < p(x) < 0.9$, which shows a high degree of uncertainty for the model; (2) since the F_1 -score is monotonously decreasing, the model tends to output false negatives rather than false positives; (3) different codecs are identified with different level of proficiency.

Compare this with the output for the model with the random mask: in the case of LIBMP3LAME and LIBVORBIS codecs, the F_1 -score is almost flat and close to 1 for the entire range of thresholds. The LIBFDK_AAC codec still shows some decrease in performance for increasing thresholds, but the peak value increased from 0.875 to 0.982 and the area under the F_1 curve jumped from 0.450 to 0.891. From the results above we can conclude that the introduction of the random mask brings higher peak performance and also reduces the impact of the choice of the threshold.

4.2.2 Analysis

Similarly to the analysis presented in Section 4.1.2, we visualise saliency maps of the model trained with the random mask in the bottom row of Figure 4. Compared to the saliency of the model with no mask (top row), we

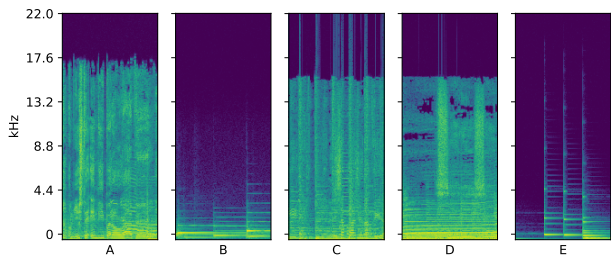


Figure 6. The five assumed lossless tracks misidentified as lossy. However, A, C, and D are in fact lossy. B and E are quiet tracks with a single instrument.

see a much brighter activation in the holes of the spectrogram without losing any of the activations at the cutoff frequency. The model has learned to rely on more markers to make its choice.

4.3 Qualitative Analysis of Errors

In this section, we present a qualitative analysis of the erroneous predictions produced by our model trained with the random mask.

4.3.1 Lossless Errors.

From our entire test subset of DS2, we observe only 5 cases (0.2%) where the model made a "lossy" prediction while the recording is in the lossless part of the dataset. The spectrogram of three out of the five tracks (A, C and D in Figure 6) show the hallmarks of lossy compression. It appears that our model was indeed correct in predicting a lossy encoding, and therefore revealed "in-the-wild" cases of accidental lossy compression that were present in our dataset.

The other two tracks (B and E) are quiet and sparsely orchestrated tracks. It is notable that the spectrogram also appears sparse, with very little energy in the upper frequency range. Given that lossy codecs often feature energy depletion in the top part of the frequency range, we hypothesise that the misclassification may be due to the model relying on the absence of energy in the upper register in this case.

4.3.2 Lossy Errors.

Table 2 shows that the entirety of cases where the model erroneously classified recordings as lossless when it should be lossy comes from tracks encoded with the LIBFDK_AAC codec. In Figure 7, spectrograms of 2 second excerpts from a random selection of error tracks are visualized.

From inspecting the spectrograms of LIBFDK_AAC encoded tracks, we find that common characteristics are (1) the spectral roll-off is relatively stable over time, (2) the preservation of transients above the cutoff frequency, which can often span upwards to the Nyquist frequency, and (3) less nulling of spectral coefficients, resulting in fewer holes in the spectrogram. The LIBFDK_AAC codec is a superior codec in terms of compression efficiency, meaning it can provide better audio quality at lower bitrates than other codecs [25].

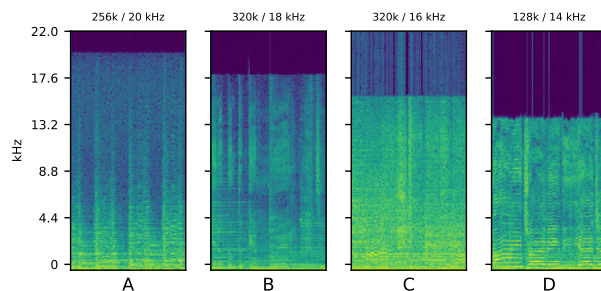


Figure 7. A random selection of lossy tracks misidentified as lossless. All tracks are encoded with LIBFDK_AAC. The spectrograms show less holes and band rupture compared to other codecs, especially under 14 kHz.

Table 2 shows that AAC with cutoff 14kHz is only 81% accuracy. We hypothesize that the LIBFDK_AAC codec does not produce as much "holes" in the spectrogram below this threshold. Our model applies the random mask to every example in our training dataset, which can be confusing on LIBFDK_AAC samples. That is, as the random mask is applied at a relatively low cutoff frequency, the resulting spectrogram is almost identical to a lossless example. One avenue for future work could be to apply the random mask with a lower probability, to allow the model to also learn other spectral characteristics of LIBFDK_AAC samples.

5. CONCLUSION

In this paper, we presented a lossy audio compression detection method that can robustly estimate whether a given audio file has been lossy encoded before. We show that naively training a model results in near-perfect lossy audio compression detection on the held-out test set generated using the same encoding parameters.

However, we find that, for several widely used lossy codecs, the performance of this model catastrophically degrades when exposed to variations of the cutoff frequency parameter that were not seen during training. This result suggests that a naively trained model is overly reliant on the cutoff value. In response to this shortcoming, we propose to amend the training strategy by applying a random mask to the upper range of the spectrogram, in order to reduce the model's reliance on the codec cutoff frequency value.

We show that this method results in a model that is significantly more robust against frequency cutoff variations. Our experiments reveal compelling performance on all codec and bit rate combinations we considered, but reveal that there remains room for improvement on the detection of the LIBFDK_AAC codec. We hypothesise that the AAC codec is comparatively more difficult to detect than MP3 and Ogg Vorbis because it generates less artefacts in the magnitude spectrogram. An avenue for future work may consist in exploring further development of the training strategy in order to improve performance on the AAC codec.

6. REFERENCES

- [1] H. G. Musmann, "Genesis of the mp3 audio coding standard," *IEEE Transactions on Consumer Electronics*, vol. 52, no. 3, pp. 1043–1049, 2006.
- [2] Xiph. (2024) Vorbis audio compression. [Online]. Available: <https://www.xiph.org/vorbis/>
- [3] M. Bosi, K. Brandenburg, S. Quackenbush, L. Fielder, K. Akagiri, H. Fuchs, and M. Dietz, "Iso/iec mpeg-2 advanced audio coding," *Journal of the Audio engineering society*, vol. 45, no. 10, pp. 789–814, 1997.
- [4] M. Bosi and R. E. Goldberg, *Introduction to digital audio coding and standards*. Springer Science & Business Media, 2002, vol. 721.
- [5] S. A. Gelfand, *Hearing: An introduction to psychological and physiological acoustics*. CRC Press, 2017.
- [6] D. A. Huffman, "A method for the construction of minimum-redundancy codes," *Proceedings of the IRE*, vol. 40, no. 9, pp. 1098–1101, 1952.
- [7] J. Herre and M. Schug, "Analysis of decompressed audio-the-inverse decoder," in *Audio Engineering Society Convention 109*. Audio Engineering Society, 2000.
- [8] P. Bießmann, D. Gärtner, C. Dittmar, P. Aichroth, M. Schnabel, G. Schuller, and R. Geiger, "Estimating MP3PRO encoder parameters from decoded audio," 2013. [Online]. Available: <https://dl.gi.de/handle/20.500.12116/20701>
- [9] S. Moehrs, J. Herre, and R. Geiger, "Analysing Decompressed Audio with the "Inverse Decoder" - Towards an Operative Algorithm," in *Audio Engineering Society Convention 112*. Audio Engineering Society, 2002.
- [10] S. Hiçsönmez, H. T. Sencar, and I. Avcibas, "Audio codec identification through payload sampling," in *2011 IEEE International Workshop on Information Forensics and Security*. IEEE, 2011, pp. 1–6.
- [11] R. Yang, Y.-Q. Shi, and J. Huang, "Defeating fake-quality MP3," in *Proceedings of the 11th ACM Workshop on Multimedia and Security*, 2009, pp. 117–124.
- [12] R. Yang, Z. Qu, and J. Huang, "Detecting digital audio forgeries by checking frame offsets," in *Proceedings of the 10th ACM Workshop on Multimedia and Security*, 2008, pp. 21–26.
- [13] B. Kim and Z. Rafii, "Lossy audio compression identification," in *2018 26th European Signal Processing Conference (EUSIPCO)*, 2018, pp. 2459–2463.
- [14] O. Derrien, "Detection of genuine lossless audio files: Application to the MPEG-AAC codec," *Journal of the Audio Engineering Society*, vol. 67, no. 3, pp. 116–123, 2019.
- [15] B. D'Alessandro and Y. Q. Shi, "MP3 bit rate quality detection through frequency spectrum analysis," in *Proceedings of the 11th ACM Workshop on Multimedia and Security*, 2009, pp. 57–62.
- [16] R. Hennequin, J. Royo-Letelier, and M. Moussallam, "Codec independent lossy audio compression detection," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 726–730.
- [17] D. Seichter, L. Cuccovillo, and P. Aichroth, "Aac encoding detection and bitrate estimation using a convolutional neural network," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 2069–2073.
- [18] Fraunhofer. (2024) Fraunhofer FDK AAC. [Online]. Available: https://wiki.hydrogenaud.io/index.php?title=Fraunhofer_FDK_AAC#Bandwidth
- [19] C. Kereliuk, B. L. Sturm, and J. Larsen, "Deep learning and music adversaries," *IEEE Transactions on Multimedia*, vol. 17, no. 11, pp. 2059–2071, 2015.
- [20] B. L. Sturm, "Two systems for automatic music genre recognition: what are they really recognizing?" in *Proceedings of the Second International ACM Workshop on Music Information Retrieval with User-Centered and Multimodal Strategies*, ser. MIRUM '12. New York, NY, USA: Association for Computing Machinery, 2012, p. 69–74. [Online]. Available: <https://doi.org/10.1145/2390848.2390866>
- [21] —, "A Simple Method to Determine if a Music Information Retrieval System is a "Horse"," *IEEE Transactions on Multimedia*, vol. 16, no. 6, pp. 1636–1644, 2014.
- [22] J. Hwang, M. Hira, C. Chen, X. Zhang, Z. Ni, G. Sun, P. Ma, R. Huang, V. Pratap, Y. Zhang, A. Kumar, C.-Y. Yu, C. Zhu, C. Liu, J. Kahn, M. Ravanelli, P. Sun, S. Watanabe, Y. Shi, Y. Tao, R. Scheibler, S. Cornell, S. Kim, and S. Petridis, "Torchaudio 2.1: Advancing speech recognition, self-supervised learning, and audio processing components for pytorch," 2023.
- [23] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A simple data augmentation method for automatic speech recognition," *Interspeech 2019*, Sep 2019. [Online]. Available: <http://dx.doi.org/10.21437/Interspeech.2019-2680>
- [24] FFMpeg. (2024) FFMpeg. [Online]. Available: <https://ffmpeg.org/>
- [25] K. Brandenburg, "MP3 and AAC explained," in *Audio Engineering Society Conference: 17th International Conference: High-Quality Audio Coding*. Audio Engineering Society, 1999.

RNBERT: FINE-TUNING A MASKED LANGUAGE MODEL FOR ROMAN NUMERAL ANALYSIS

Malcolm Sailor

Yale University

malcolm.sailor@gmail.com

ABSTRACT

Music is plentiful, but labeled data for music theory tasks like roman numeral analysis is scarce. Self-supervised pretraining is therefore a promising avenue for improving performance on these tasks, especially because, in learning a task like predicting masked notes, a model may acquire latent representations of music theory concepts like keys and chords. However, existing models for roman numeral analysis have not used pretraining, instead training from scratch on labeled data, while conversely, pretrained models for music understanding have generally been applied to sequence-level tasks requiring little explicit music theory, such as composer classification. In contrast, this paper applies pretraining methods to a music theory task by fine-tuning a masked language model, MusicBERT, for roman numeral analysis. We apply token classification to get a chord label for each note and then aggregate the predictions of simultaneous notes to achieve a single label at each time step. The resulting model substantially outperforms previous roman numeral analysis models. Our approach can readily be extended to other note- and/or chord-level music theory tasks (e.g., nonharmonic tone analysis, melody harmonization).

1. INTRODUCTION

Roman numeral analysis is the task of identifying the chords in a piece of music and then indicating their role with respect to the current key. Although it was developed in the European classical tradition, it is an essential element of the musical toolkit for musicians working in many different Western-derived styles (for instance, a jazz musician might speak of a “ii-V to vi (“two-five to six’”)” or a pop musician might say “the bridge starts on IV”).

A small but growing literature has employed deep learning models for automatic Roman numeral analysis of symbolic music (Section 2), training the models from scratch on labeled data. But labeled data for Roman numeral analysis is scarce, whereas symbolic music is comparatively plentiful. Self-supervised pretraining therefore seems likely to yield dividends, especially considering

that, in order to perform a self-supervised task, the model can be expected to learn latent representations of music theory concepts like chords and scales: if you need to predict a given musical note, you will do a lot better if you can estimate the expected key and chord. Moreover, as a general matter, it seems likely that it will prove more efficient and more practical for MIR researchers and music theorists to fine-tune large-scale foundation models for specific analytical tasks, rather than training bespoke models from scratch for every task. These considerations inspire the present work, where we fine-tune a masked language model on Roman numeral analysis, obtaining state-of-the-art classification accuracy.¹

2. RELATED WORK

Whereas automatic chord recognition using audio signals has an extensive literature [1], this paper contributes to a smaller body of work on Roman numeral analysis of symbolic music, which is both an easier and a harder problem. Easier, because working with symbolic data means the model does not need to devote capacity to identifying the sounding pitches, but harder, because Roman numeral analysis requires not only identifying chords but also describing their harmonic function within their tonal context (e.g., rather than simply labeling a chord as “E major”, labeling it as “V6/vi in C major”). Details of Roman numeral analysis are beyond the scope of this paper but are covered in any textbook of classical harmony such as [2, 3].

While earlier work employed various approaches for automated Roman numeral analysis, more recently, deep learning has come to predominate, including recurrent models [4–7], transformers [8, 9], and, most recently, graph-based architectures [10]. As far as we know, the best performance in the existing literature has been obtained by *AugmentedNet* [6,7] and *ChordGNN* [10], and we compare our results below with those reported in [6, 10].²

All of these models for Roman numeral analysis are trained from scratch, not making use of self-supervised pretraining. A hitherto separate area of research is pretraining large models for the understanding of symbolic music. Both MusicBERT [11], the model used in this paper, and MidiBERT-Piano [12] pretrain BERT-like [13]



© M. Sailor. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** M. Sailor, “RNBert: Fine-Tuning a Masked Language Model for Roman Numeral Analysis”, in *Proc. of the 25th Int. Society for Music Information Retrieval Conf.*, San Francisco, United States, 2024.

¹ We release the code to reproduce our results at <https://github.com/malcolmsailor/rnbert>.

² Unfortunately, the results of [7] are reported in a manner that makes them difficult to compare directly with these other papers and with our own results.

encoder-only transformers on a masked language modeling task. [14] adds a GPT-like causal language modeling task as well. So far, these pretrained models have mainly been fine-tuned on sequence-level tasks such as composer, genre, or emotion classification. As far as we know, none of these pretrained models have been applied to Roman numeral analysis, or any other problem involving the prediction of explicit music-theoretical labels.

3. EXPERIMENTAL SETUP

3.1 Corpus

To our knowledge, the corpus used in this study is the largest yet assembled for Roman numeral analysis. The major components of this corpus include the various corpora released by the Digital and Cognitive Musicology Lab [15–17], the TAVERN set of theme and variations by Beethoven and Mozart [18], the set of Beethoven Piano Sonatas first movements introduced in [4], and the various other items included in the When in Rome meta-corpus [19], including analyses of Bach preludes and chorales, and a large number of 19th century lieder, including works of women composers. The total contents of this corpus are enumerated in the first line of Table 1.

Data subset	Scores	Notes	Chords
All	1,404	1,289,888	161,473
AugmentedNet v1	347	701,703	77,570

Table 1. Overall contents of the datasets used for training and evaluation.

For a fair comparison with [6, 10], we also train and evaluate on the subset of our data used in those papers, employing the same training/validation/testing splits.³ We refer to this subset as “AugmentedNet v1” to distinguish it from the somewhat larger dataset used in [7]. Note that, for scores having two analyses (in particular, the scores of the TAVERN dataset, where each score was analyzed by two separate annotators), AugmentedNet v1 includes both versions (following [6]), whereas in the full corpus, we randomly choose only one of the two versions for inclusion. AugmentedNet v1’s note count is substantially increased by the inclusion of these duplicate TAVERN scores, which comprise 106,981 notes.

Unlike some prior work (e.g., [6, 9]), we do not experiment with training and/or evaluating on smaller, more homogenous subsets of our corpus (for example, the Beethoven piano sonatas only). Our goal is to train the best and most general Roman numeral analysis model we can and we expect that such a model will be best obtained and evaluated by using as much data as possible.

³ 7 scores from AugmentedNet v1 were excluded because of preprocessing errors (for example, because they include time signatures not supported by MusicBERT’s OctupleMIDI encoding scheme). These scores exclusively came from the training split and so, if their omission has any effect on RNBert’s performance relative to that of the other models trained on the AugmentedNet v1 dataset, it should bias it downwards.

3.2 Data representation

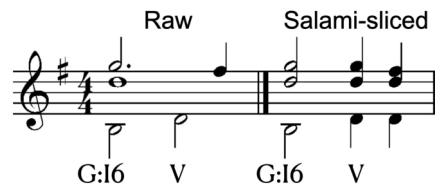


Figure 1. A hypothetical musical example and its analysis, before and after salami-slicing.

Onset	0	0	0	2	2	2	3	3	3
Release	2	2	2	3	3	3	4	4	4
Pitch	59	67	79	62	69	79	62	69	78
RN	I6	I6	I6	V	V	V	V	V	V
Key	G	G	G	G	G	G	G	G	G

Table 2. The example from Figure 1 in tabular format after salami-slicing. Time signatures, tempi, and bar lines are omitted.

The scores associated with the analyses in our dataset are encoded in a variety of formats, such as MusicXML (.xml or .mxl), MuseScore (.mcsx), or Humdrum (.krn). We convert these into a tabular format, illustrated in Table 2, labeling each note with the associated key and chord annotation. Note that, because we use a model that was pretrained on MIDI data, which does not specify pitch-spelling, our pitch inputs are not spelled (that is, they use midi numbers like “78” rather than pitch names like “F#5”). For further discussion, see Section 3.3.1.

When reading scores, we apply several preprocessing steps.

First, following MusicBERT, the score is quantized at the 64th note level.

Second, we “salami-slice” the score: at each timestep with one or more onsets or releases, we split any ongoing notes into two, in order to obtain a purely homophonic rhythmic texture in which all onsets and releases are synchronized across all parts. (The term “salami-slicing” is due to [20].) Salami-slicing is necessary to ensure that each note belongs to only one chord. Otherwise, a note may persist through multiple changes of chord, either because it is a common tone among each of the chords (like the alto D in Figure 1), or because it realizes a suspension or similar dissonant idiom (like the soprano G in Figure 1). Fortunately for our purposes, salami-slicing should not affect harmonic analysis, because it does not change the pitch content of the score. Moreover, musical idioms like suspensions and pedal tones that cross changes of chord are analyzed the same whether or not they are tied or sounded anew at the onset of the new chord.

Third, we dedouble the notes of the score, removing any notes that have the same pitch, onset, and release (regardless of whether they are performed by the same instrument). Dedoubling has two advantages. First, it reduces the sequence length. Second, it produces a more homogeneous texture between music for small ensembles,

where pitch doubling is less common, and music for large ensembles like orchestras, where pitch doubling is ubiquitous. Such homogeneity is particularly desirable since nearly all the available labeled scores are small-ensemble works like piano sonatas and string quartets, but we would like our model to generalize to large-ensemble works like symphonies and operas.

MusicBERT has a maximum sequence length of 1000 tokens. Therefore, in both training and evaluation, we crop scores into segments of 1000 tokens, stepping through the score with a hop size of 250.

3.3 Task

Roman numeral analysis involves labeling chords with integers (Roman numerals) indicating their root with respect to the scale of the current key (e.g., “V” in C major indicates that the root is G, the 5th scale degree). Figured-bass numerals are typically appended to indicate the chord’s inversion (e.g., “6” for first inversion). While the quality of the chord is often assumed to conform to the scale, alterations of quality can be indicated according to a variety of conventions (e.g., upper-/lower-case for major/minor, appending “+” or “o” to indicate augmented or diminished chords, or combining figured-bass numerals with accidental signs). The Roman numerals can be prefixed with accidentals to indicate altered scale degrees (for example, bVI in C major would be A-flat). “Tonicizations”—that is, one or more chords borrowed from another key—can be indicated by “secondary” Roman numerals, typically following a slash, as in “V/ii”, which in C major would indicate the V chord of d minor. Finally, since the Roman numeral only indicates a harmony with respect to some key, for a Roman numeral to be meaningful, we need to indicate the key as well.

Since a complete Roman numeral consists of multiple distinct elements, the combinatorial space of these elements is very large. Encoding each distinct combination as a token, would require a large and sparse vocabulary, posing challenges for training and generalization. Therefore, the approach adopted here and elsewhere is to treat Roman numeral analysis as a multitask learning problem, where we predict the key, quality, inversion, and degree separately. (The degree is sometimes further decomposed into “primary” and “secondary” components, but in the current work, we predict these jointly.) It should be noted, however, that this multitask may approach obtain a smaller vocabulary size at the expense of some coherence among the different elements of the Roman numeral. For example, suppose there is a passage that is ambiguous between I and vi6 (two chords which share two of their three pitch-classes as well as the same bass note). If the model distributes the probability roughly equally between the two possibilities it may easily occur that the inversion and degree predictions “decohere” and we end up with a plainly incorrect prediction like I6 (rather than I) or vi (rather than vi6). (One solution to this problem of decoherence was proposed by [21], which we discuss in Section 3.6 below.)

3.3.1 Pitch spelling

One difference between our approach and some prior work (e.g., [5, 7]) is that MusicBERT uses unspelled pitch inputs (midi numbers like “67”) rather than letter names (like “F#5”). Our output key predictions are therefore also unspelled (e.g., pitch-class 6, rather than “F-sharp”) because, with unspelled inputs, the output spelling is undefined.⁴

We consider our model’s inability to predict spelled keys unimportant. Given spelled inputs (e.g., pitches like “Db5” rather than MIDI numbers like “61”) and an unspelled key (e.g., “1 major”), predicting a spelled key (e.g., “Db major”) is trivial and could likely be performed with perfect accuracy by a rule-based algorithm (e.g., taking the enharmonically equivalent key closest to the centroid of the spelled pitches on the “line of fifths” [23]). Moreover, keys with plausible enharmonic equivalents (like F-sharp major or E-flat minor) are rarely used. Their classification is therefore unlikely to significantly affect validation/test performance.

3.4 Data augmentation

We employ two data augmentation techniques on the training data. First, we transpose each score to all 12 keys of the chromatic scale. Second, we create a version of each score with the durations scaled by a factor of 2. If the mean duration in the score is greater than the mean duration of the training set as a whole, we scale its durations down by 2; if it is less, we scale them up by 2.

We experimented with adding synthetic data similar to the procedure introduced in [6, 7] and also adopted in [10]. However, we did not find that it improved the model performance. It is possible that synthetic data was less helpful in our case than with AugmentedNet [7] because, whereas for that model, the inputs consist of pitch-vectors at each time step, for our model, the inputs are simply the notes of the score, and therefore the difference between synthetic and real data is more apparent to the model.

3.5 Model

3.5.1 MusicBERT

The model that we fine-tune, MusicBERT [11], is a bidirectional transformer encoder pretrained on a masked language modeling task. The dataset for pretraining is a corpus of over 1 million midi files, 3 orders of magnitude larger than our Roman numeral dataset. This difference in scale motivates the use of a pretrained model. We chose MusicBERT for our experiments because, of the pretrained symbolic music models of which we are aware, it used the largest pretraining dataset (by comparison, [12] pretrains on fewer than 5,000 scores of exclusively piano music) and also because of the elegance of the OctupleMIDI scheme MusicBERT uses to encode its inputs. A more detailed

⁴ This is because the only difference between enharmonically equivalent keys like F-sharp and G-flat is how they are notated. Certain pieces of music, such as Fugue no. 8 of Bach’s Well-tempered Clavier, Book 1, have even been variously printed in two enharmonically equivalent keys [22].

comparison fine-tuning symbolic music models for Roman numeral analysis will have to await further work.

In the OctupleMIDI encoding, eight features of a musical note are first embedded individually: time signature, tempo, bar number, metric position within the bar, pitch, duration, and velocity.⁵ To obtain a single input at each time step, these eight embeddings are concatenated and then projected to the model’s embedding dimension. OctupleMIDI reduces the sequence length when compared with other encoding schemes like Midi-like [24,25], REMI [26], or Compound Word [27]. This reduction occurs because in OctupleMIDI, tokens and notes are in one-to-one correspondence, whereas the other schemes use tokens for other items such as time-signatures or barlines. For our use case, the fact that all tokens correspond to notes has the added virtue that we do not waste computations classifying non-note tokens.

In our experiments, we use the MusicBERT “base” architecture, whose hyperparameters are modeled on those of the BERT base architecture ([13]), with a hidden dimension of 768, 12 layers, and 12 attention heads. We use the pretrained checkpoint provided by [11]. For further details we refer the reader to the original MusicBERT paper.

MusicBERT is not trained solely or even mainly on Classical music, whereas our annotated data consists entirely of Classical music. This does not seem likely to be a problem, because in the first place, a great deal of the tonal idiom (i.e., keys, chords) is shared between different styles of tonal music, and tonal music surely predominates in MusicBERT’s training set. Moreover, to pretrain on only classical music would mean greatly reducing the amount of training data, as it’s extremely unlikely that 1,000,000 distinct midi files of Classical music exist. ([20] is based on what is to our knowledge the largest corpus of Classical midi files in existence and features under 15,000 files.)

3.5.2 Token classification

To perform Roman numeral analysis, we adopt a token classification approach, predicting the key and Roman numeral for each token in the input. Since each token corresponds to a note, this amounts to predicting the chord during which each note occurs. While training, we calculate the loss on a per-token basis. In evaluation, in order to obtain a single prediction for each salami slice, we average the logits of simultaneous notes.

The token-classification heads are two-layer multilayer perceptrons (MLPs) whose inner dimension is the embedding dimension of the model (768, in our experiments).

To obtain the overall loss, we simply take the mean of the cross-entropy loss for each individual task. We tried learning a weighting of the contribution of each task to the global loss, following the approach introduced by [28] and implemented in an MIR context by [29], but observed a small degradation in model quality when doing so.

⁵ Midi velocity is encoded in the OctupleMIDI format but is absent from the symbolic scores of our dataset. Therefore, we use a default value of 96 for all note velocities in our dataset.

3.5.3 Fine-tuning procedure

In our fine-tuning experiments, we found it important to freeze parts of the model to reduce the number of trainable parameters and avoid overfitting. Freezing the first 9 layers of MusicBERT seemed to give the best results. The parameter counts are given in Table 3.

We used a learning rate of 2.5×10^{-4} with a linear warmup of 2500 steps followed by a linear decay of the learning rate to 0. When training on multi-task Roman numeral classification, we fine-tuned for 50,000 steps. When training on key classification only (see Section 3.6), we fine-tuned for 25,000 steps.

When experimenting with varying these hyperparameters, we did not typically find their precise values to have much effect on the performance of the model. This implies that the fine-tuning is fairly robust to different hyperparameter choices.

Model	Total	Trainable
Base	108,805,598	26,782,729
Key conditioned	111,023,816	28,859,891

Table 3. RNBert parameter counts.

3.6 Key conditioning

In preliminary versions of RNBert, we found that high entropy of the output probabilities for the degree task seemed to mainly occur in two distinct scenarios. The first scenario involved unusual or hard-to-analyze chords, where high entropy is to be expected. The second scenario involved chords that, given a key, were straightforward to analyze, but where the model appeared to be uncertain about the choice of key.⁶ To illustrate this latter scenario, suppose we are analyzing a passage, and we recognize an A minor chord, but we are uncertain whether the key is C major or G major. In that case, though we will not know whether to label it with “vi” or “ii,” this uncertainty isn’t about the chord itself, but only about the key. If the model distributes the probability mass roughly evenly between the two possibilities, it may emit an incoherent composite prediction like “ii of C major” (a D minor chord) or “vi of G major” (an E minor chord). In general, the degree task depends on the key task in this way. Therefore, in some of our experiments, we made the Roman numeral prediction by conditional on the key.

In these key-conditioned experiments, we embed the key tokens with a two-layer MLP with hidden and output dimensions of 256 and GELU activation.⁷ We then

⁶ We do not have space to discuss this further, but there are music theoretic reasons to think that a certain degree of uncertainty about key annotations is inevitable because the key of certain passages, especially transitional ones, can be analyzed in more than one way.

⁷ In principle, we should be able to replace this MLP with a simple embedding layer and obtain the same results. In practice, however, we found that using a simple embedding layer barely improved performance above the unconditioned baseline, even with teacher forcing. We suspect that this occurs because the loss landscape of the MLP has better training dynamics. After training, on the other hand, it should be possible to replace the MLP with an embedding table that simply encodes the output

concatenate this key embedding with the output from MusicBERT to obtain the input to the Roman numeral classification heads.

In training, we employ teacher forcing, that is, we condition on the ground-truth key annotations from the labeled data. In evaluation, we first predict the key with a separately fine-tuned model, then condition the chord predictions on these predicted keys.

Another attempt to encourage coherence between key and Roman numeral predictions is [21], who use a neural autoregressive distribution estimator (NADE). Their approach extends beyond ours insofar as it conditions each sub-task of the Roman numeral classification on the previous tasks. In preliminary experiments applying a similar approach to RNBert, we observed a small decline in performance across all metrics. We defer to future work a qualitative evaluation of these predictions and further similar experiments.

3.7 Post-processing steps

In postprocessing, we collate the predictions from each segment, combining the overlapping logits of adjacent segments by linearly interpolating between them. (By analogy to audio signal processing we could say that we cross-fade between the logits of neighboring segments.) We then average the logits of simultaneous notes to obtain a single set of logits for each salami-slice.

To avoid implausibly brief key changes of one or two salami-slices’ duration (which otherwise sometimes occur at transitions between keys, when the model estimates both keys to be approximately equiprobable), we use a dynamic programming approach to decode the key predictions. Specifically, we employ the Viterbi algorithm, using RNBert’s output probabilities as the emission probabilities and defining a transition probability matrix that is uniform, except for self-transitions, whose probabilities are upweighted. This decoding scheme has a negligible effect on the measured accuracy of the predictions, while effectively eliminating implausibly brief key changes.

4. RESULTS AND DISCUSSION

Table 4 provides our results, expressed following [6] as the proportion of time the predicted labels are accurate, with 32nd-note resolution. We give two sets of results: on lines 1 to 3, training on the dataset and training/validation/testing splits used by [6, 10] for a fair comparison with these prior papers, and on lines 4 to 6, training on our complete dataset.

Concerning the composite “RN” labels, RN_{root} refers to the conjunction of degree, quality, inversion, and key, while $RN_{\text{+root}}$ adds to these the chord root. Predicting the root is redundant: the root of a Roman numeral is a deterministic function of the degree and key (e.g., the root of #iv in C major is F-sharp). There is hence no need to include

of the MLP for each key. However, the MLP contributes such a small proportion of the model’s overall parameter count that we did not bother to do so.

it in the composite Roman numeral, or indeed, to predict it at all. Therefore, when training RNBert on our full dataset, we do not predict the root and report only RN_{root} . When training on the AugmentedNet v1 data subset, in contrast, in order to ensure a fair comparison with the prior models, both of which predicted the root, we train RNBert to predict the root and report the results for both $RN_{\text{+root}}$ and RN_{root} . It can be seen that the inclusion or exclusion of the root makes almost no difference, as one would expect. Finally RN_{alt} refers to an alternate task learned in [6, 10], replacing the quality, degree, and root predictions with a vocabulary of the 75 most common Roman numerals in the AugmentedNet v1 training set. We did not train RNBert on this task, but we report the prior results on it to facilitate comparison with our results.

When training on the AugmentedNet v1 subset (Table 4, line 3), RNBert substantially outperforms the prior models on degree and quality. However, it outperforms the earlier models by a much more substantial margin when predicting the composite Roman numeral $RN_{\text{+root}}$. This implies that there is more coherence among the various dimensions of its predictions. Such coherence may be due to the robustness of the representations MusicBERT learns in its pretraining. It implies that, even where RNBert’s predictions don’t agree with a human annotator’s, they are more likely to be useful, since they are more likely to be internally consistent.

When training on the complete dataset (Table 4, lines 4–6), RNBert exceeds the performance of the earlier models by an even larger margin, especially when predicting the composite Roman numeral. We defer a discussion of the effect of key conditioning to Section 4.1.

One thing to note about these results is that, while the models on lines 4 and 5 greatly exceed the performance of the models on lines 1–3 on degree, quality, inversion, and RN_{root} prediction, when it comes to key prediction, the AugmentedNet v1-trained models actually perform better (with the exception of the ChordGNN model). We believe this occurs because key prediction on the subset is simply an easier problem, since it contains less music from the late 19th century and beyond, a period when music tended to modulate more widely.

4.1 Effect of key conditioning

In Table 4, line 6, it can be seen that conditioning the Roman numeral prediction on the ground-truth key (i.e., teacher forcing) has a large effect on degree accuracy. This constitutes a sanity check that the conditioning works as expected: if the model knows the key of the annotation, its ability to predict the Roman numeral’s degree shoots up. By contrast, key conditioning, with or without teacher forcing, has little effect on the “quality” and “inversion” metrics. This is also expected, since these tasks do not depend on the key: a first-inversion minor chord is a first-inversion minor chord regardless of the key in which it occurs.

It is somewhat harder to interpret a comparison of RNBert, conditioned on the predicted key (line 5), with the

Model	Degree	Quality	Accuracy			
			Inversion	Key	RN _{+root}	RN _{-root}
<i>AugmentedNet v1 data subset</i>						
1 AugmentedNet [6]	.67	.797	.788	.829	.464	.515
2 ChordGNN+(Post) [10]	.714	.784	.803	.813	.518	.529
3 RNBert (key conditioned)	.731	.819	.796	.825	.574	.575
<i>All data</i>						
4 RNBert (unconditioned)	.762	.867	.872	.822		.620
5 RNBert (key conditioned)	.749	.864	.872	.823		.624
6 RNBert (key conditioned, teacher forcing)	.859	.865	.872	N/A		N/A

Table 4. Accuracy of RNBert and two prior models. The meanings of RN_{+root}, RN_{-root}, and RN_{alt} are described in Section 4. In the model comparison of lines 1–3, we indicate the best metric in **bold** type. Because the teacher-forcing model on line 6 does not predict key, and RN prediction involves key prediction, we do not report RN results for this model.

Ground truth: F:V V₅⁶ C:IV V₅⁶/V I₄⁶ V⁷ I

RNBert (unconditional): F:V⁶ V₅⁶ I V₅⁶/ii I₄⁶ V⁷/V V

RNBert (conditioned): F:V⁶ V₅⁶ I V₅⁶/ii I₄⁶/V V⁷/V V

Figure 2. Beethoven, String Quartet in F major, op. 18, no. 1, iv, mm. 7–8. Arguably correct predictions that do not agree with the human annotations are printed in *italic* type and serve to illustrate the discussion in Section 4.2. The prediction printed in ~~strikethrough~~ type is straightforwardly incorrect and illustrates the discussion in Section 4.1.

unconditional RNBert (line 4). The unconditional model does better predicting degree, but the conditioned model does better predicting the composite RN_{-root}. These results make sense if key conditioning makes the key and Roman numeral predictions more coherent with one another. Even when the unconditional model does not predict the labeled key, it should still predict the labeled degree some proportion of the time, causing its degree accuracy to be higher. The conditional model, on the other hand, should be less likely to do this, but its composite RN prediction should be more coherent and thus more accurate.

Figure 2 can serve as an illustration. The two RNBert analyses are almost identical, being in the key of F major throughout, with tonicized dominant chords at the half cadence that concludes the example. The lone difference occurs at the cadential 64 chord on the downbeat of the second measure. Here, while the conditioned model gives the correct annotation I₆₄/V, the unconditioned analysis gives I₆₄, which is incorrect, since this is a C major chord, and the annotated key is F. The unconditioned model’s key and Roman numeral predictions are each plausible on their

own—I₆₄ is the most common annotation for a cadential 64 chord—but they do not cohere with one another. And yet, in spite of being incorrect with respect to its predicted key, this I₆₄ prediction happens to agree with the ground truth, and thus the degree accuracy of this example is (spuriously) higher for the unconditioned model.

4.2 The problem of multiple acceptable analyses

One important problem in evaluating Roman numeral analysis models is that there is often more than one correct analysis of a musical passage, so that a model’s predictions can be labeled “inaccurate” even when they present valid alternate readings. For example, this may occur with brief passages in another key, which can be analyzed as either modulations (indicated by a change of key) or as tonicizations (indicated with secondary Roman numerals).

On a priori grounds, as well as based on qualitative sampling of the model’s predictions, we suggest that a high proportion of RNBert’s “inaccurate” predictions are likely to be acceptable alternate analyses. For example, in Figure 2, it is reasonable to analyze the half cadence that concludes the example as a brief modulation to C, as the human annotator did, or as a tonicization, as done by both versions of RNBert. Either analysis is acceptable, but the divergence means that nearly all labels in RNBert’s analysis do not agree with the ground truth, in spite of being arguably correct. These considerations may place a ceiling on the accuracy of all Roman numeral analysis models.

5. CONCLUSION

At the broadest level, our results imply that, by fine-tuning pretrained models, we can obtain state-of-the-art performance on music theory tasks. In the specific case of Roman numeral analysis, we suggest that Roman numeral analysis models have now matured to the point where they are ready to be used in large-scale musicological studies. Finally, we note that the approach described in this paper could be readily extended to many other music theory tasks that can be conceived of as a labeling of the notes of a score, including the analysis of dissonant idioms (suspensions, passing tones, and the like) or melody harmonization (that is, labeling each pitch of a melody with a chord).

6. REFERENCES

- [1] J. Pauwels, K. O’Hanlon, E. Gomez, and M. B. Sandler, “20 Years of Automatic Chord Recognition from Audio,” in *Proceedings of the International Society for Music Information Retrieval Conference*, 2019.
- [2] E. Aldwell, C. Schachter, and A. C. Cadwallader, *Harmony & Voice Leading*, 4th ed. Schirmer/Cengage Learning, 2011.
- [3] S. M. Kostka and D. Payne, *Tonal Harmony, with an Introduction to Twentieth-Century Music*, 5th ed. McGraw-Hill, 2004.
- [4] T.-P. Chen and L. Su, “Functional harmony recognition of symbolic music data with multi-task recurrent neural networks.” in *Proceedings of the International Society for Music Information Retrieval Conference*, 2018, pp. 90–97.
- [5] G. Micchi, M. Gotham, and M. Giraud, “Not All Roads Lead to Rome: Pitch Representation and Model Architecture for Automatic Harmonic Analysis,” *Transactions of the International Society for Music Information Retrieval*, vol. 3, no. 1, pp. 42–54, 2020.
- [6] N. N. López, M. Gotham, and I. Fujinaga, “AugmentedNet: A Roman Numeral Analysis Network with Synthetic Training Examples and Additional Tonal Tasks.” in *Proceedings of the International Society for Music Information Retrieval Conference*, 2021, pp. 404–411.
- [7] N. N. López, “Automatic Roman Numeral Analysis in Symbolic Music Representations,” Ph.D. dissertation, McGill University (Canada), 2022.
- [8] T.-P. Chen and L. Su, “Harmony Transformer: Incorporating chord segmentation into harmony recognition,” in *Proceedings of the International Society for Music Information Retrieval Conference*, 2019.
- [9] —, “Attend to Chords: Improving Harmonic Analysis of Symbolic Music Using Transformer-Based Models,” *Transactions of the International Society for Music Information Retrieval*, vol. 4, no. 1, pp. 1–13, 2021.
- [10] E. Karystinaios and G. Widmer, “Roman Numeral Analysis with Graph Neural Networks: Onset-wise Predictions from Note-wise Features,” in *Proceedings of the International Society for Music Information Retrieval Conference*. arXiv, 2023.
- [11] M. Zeng, X. Tan, R. Wang, Z. Ju, T. Qin, and T.-Y. Liu. (2021) MusicBERT: Symbolic Music Understanding with Large-Scale Pre-Training. [Online]. Available: <http://arxiv.org/abs/2106.05630>
- [12] Y.-H. Chou, I.-C. Chen, C.-J. Chang, J. Ching, and Y.-H. Yang. (2021) MidiBERT-Piano: Large-scale Pre-training for Symbolic Music Understanding. [Online]. Available: <http://arxiv.org/abs/2107.05223>
- [13] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” 2019. [Online]. Available: <http://arxiv.org/abs/1810.04805>
- [14] Z. Li, R. Gong, Y. Chen, and K. Su, “Fine-Grained Position Helps Memorizing More, a Novel Music Compound Transformer Model with Feature Interaction Fusion,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 4, pp. 5203–5212, 2023.
- [15] M. Neuwirth, D. Harasim, F. C. Moss, and M. Rohrmeier, “The Annotated Beethoven Corpus (ABC): A Dataset of Harmonic Analyses of All Beethoven String Quartets,” *Frontiers in Digital Humanities*, vol. 5, p. 16, 2018.
- [16] J. Hentschel, M. Neuwirth, and M. Rohrmeier, “The Annotated Mozart Sonatas: Score, Harmony, and Cadence,” *Transactions of the International Society for Music Information Retrieval*, vol. 4, no. 1, pp. 67–80, 2021.
- [17] J. Hentschel, Y. Rammos, M. Neuwirth, and M. Rohrmeier, “An Annotated Corpus of Tonal Piano Music from the Long 19th Century,” 2022. [Online]. Available: <https://zenodo.org/records/10171721>
- [18] J. Devaney, C. Arthur, N. Condit-Schultz, and K. Nisula, “Theme And Variation Encodings with Roman Numerals (TAVERN): A New Data Set for Symbolic Music Analysis.” in *Proceedings of the International Society for Music Information Retrieval Conference*, 2015, pp. 728–734.
- [19] M. Gotham, G. Micchi, N. N. López, and M. Sailor, “When in Rome: A Meta-corpus of Functional Harmony,” *Transactions of the International Society for Music Information Retrieval*, vol. 6, no. 1, pp. 150–166, 2023.
- [20] C. W. White and I. Quinn, “The Yale-Classical Archives Corpus,” *Empirical Musicology Review*, vol. 11, no. 1, pp. 50–58, 2016.
- [21] G. Micchi, K. Kosta, G. Medeot, and P. Chanquion, “A deep learning method for enforcing coherence in Automatic Chord Recognition.” in *Proceedings of the International Society for Music Information Retrieval Conference*, 2021, pp. 443–451.
- [22] P. Badura-Skoda, *Interpreting Bach at the Keyboard*. Oxford University Press, 1995.
- [23] D. Temperley, “The Line of Fifths,” *Music Analysis*, vol. 19, no. 3, pp. 289–319, 2000.
- [24] C.-Z. A. Huang, A. Vaswani, J. Uszkoreit, N. Shazeer, I. Simon, C. Hawthorne, A. M. Dai, M. D. Hoffman, M. Dinculescu, and D. Eck, “Music Transformer,” 2018. [Online]. Available: <http://arxiv.org/abs/1809.04281>

- [25] S. Oore, I. Simon, S. Dieleman, D. Eck, and K. Simonyan, “This time with feeling: Learning expressive musical performance,” *Neural Computing and Applications*, vol. 32, no. 4, pp. 955–967, 2020.
- [26] Y.-S. Huang and Y.-H. Yang, “Pop Music Transformer: Beat-based Modeling and Generation of Expressive Pop Piano Compositions,” in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020.
- [27] W.-Y. Hsiao, J.-Y. Liu, Y.-C. Yeh, and Y.-H. Yang, “Compound Word Transformer: Learning to Compose Full-Song Music over Dynamic Directed Hypergraphs,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.
- [28] L. Liebel and M. Körner. (2018) Auxiliary Tasks in Multi-task Learning. [Online]. Available: <http://arxiv.org/abs/1805.06334>
- [29] J. Qiu, C. L. P. Chen, and T. Zhang. (2022) A Novel Multi-Task Learning Method for Symbolic Music Emotion Recognition. [Online]. Available: <http://arxiv.org/abs/2201.05782>

Papers – Session VI

MUCHOMUSIC: EVALUATING MUSIC UNDERSTANDING IN MULTIMODAL AUDIO-LANGUAGE MODELS

Benno Weck*¹
Elio Quinton³

Ilaria Manco*²
George Fazekas²

Emmanouil Benetos²
Dmitry Bogdanov¹

¹Universitat Pompeu Fabra, ²Queen Mary University of London, ³Universal Music Group

* equal contribution benno.weck01@estudiant.upf.edu, i.manco@qmul.ac.uk

ABSTRACT

Multimodal models that jointly process audio and language hold great promise in audio understanding and are increasingly being adopted in the music domain. By allowing users to query via text and obtain information about a given audio input, these models have the potential to enable a variety of music understanding tasks via language-based interfaces. However, their evaluation poses considerable challenges, and it remains unclear how to effectively assess their ability to correctly interpret music-related inputs with current methods. Motivated by this, we introduce MuChoMusic, a benchmark for evaluating music understanding in multimodal language models focused on audio. MuChoMusic comprises 1,187 multiple-choice questions, all validated by human annotators, on 644 music tracks sourced from two publicly available music datasets, and covering a wide variety of genres. Questions in the benchmark are crafted to assess knowledge and reasoning abilities across several dimensions that cover fundamental musical concepts and their relation to cultural and functional contexts. Through the holistic analysis afforded by the benchmark, we evaluate five open-source models and identify several pitfalls, including an over-reliance on the language modality, pointing to a need for better multimodal integration. Data and code are open-sourced.¹

1. INTRODUCTION

Combining the success of large language models (LLMs) with new advances in machine perception that have led to image, audio and video foundation models [1], multimodal LLMs are becoming influential across many fields [2–6]. Recently, models of this kind have started supporting the audio modality, with a subset also being applied to the music domain [7–13]. We refer to such models exhibiting audio understanding capabilities as *Audio LLMs*. In a nut-

¹ Data: <https://doi.org/10.5281/zenodo.12709974>, website: <https://mulab-mir.github.io/muchomusic>

© B. Weck, I. Manco, E. Benetos, E. Quinton, G. Fazekas, and D. Bogdanov. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** B. Weck, I. Manco, E. Benetos, E. Quinton, G. Fazekas, and D. Bogdanov, “MuChoMusic: Evaluating Music Understanding in Multimodal Audio-Language Models”, in *Proc. of the 25th Int. Society for Music Information Retrieval Conf.*, San Francisco, United States, 2024.

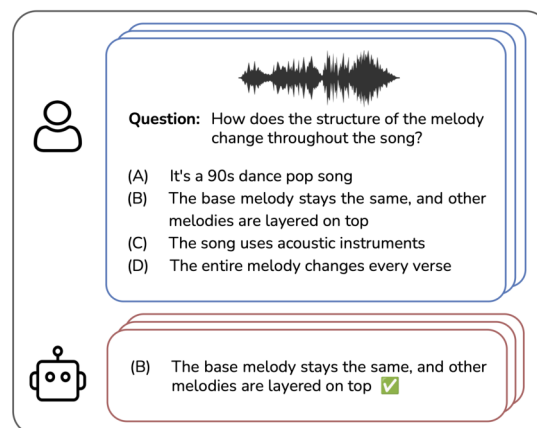


Figure 1. Multiple-choice questions in MuChoMusic have four answer options of different levels of difficulty.

shell, Audio LLMs consist of pre-trained LLMs whose input space has been expanded beyond text to include tokens from an audio encoder, granting them the ability to produce language outputs that require understanding of both modalities. While promising, these models also inherit many of the limitations of LLMs and little attention has so far been given to their evaluation. In most cases, current automatic evaluation relies on match-based metrics which measure the semantic or lexical overlap between model outputs and reference text. However, many works have pointed out deficiencies in this approach [14], which fails to capture the large space of acceptable language outputs admitted by open-ended tasks. For example, the question “*What are some possible uses for this music in a film or TV show?*” may be suitably answered in many different ways. Secondly, automatic music understanding evaluation via language is only supported by a handful of human-annotated datasets [15–17], of which only one [15] has widely been adopted in the context of Audio LLMs. Instead, many prior works have created a variety of ad-hoc datasets built upon synthetically generated captions from tags and other metadata [8, 9, 18] to train and evaluate their models, without explicit data validation mechanisms, which raises questions around their reliability. These three key issues, lack of standardisation, the inadequacy of text generation metrics, and the quality of annotations in current datasets, pose obstacles to the development of the field and has prompted some to resort to human evaluations [7], which can be costly and are hard to scale and reproduce.

In this paper, we present *MuChoMusic*, the first benchmark for evaluating music understanding in Audio LLMs. We design a test that is easy to evaluate by collecting a set of multiple-choice (MC) questions that are scrutinised by human annotators, on which simple classification accuracy can be obtained as a reliable indicator of music understanding over the categories covered by the test. The content of our benchmark is intended to be challenging, grounded in factual music knowledge, and tests core understanding and reasoning skills across several dimensions such as music theory, musical styles and traditions, historical and social contexts, structure and expressive analysis. Using *MuChoMusic*, we carry out a comprehensive evaluation of five existing Audio LLMs with music understanding capabilities. We envision that *MuChoMusic* will complement prior efforts to standardise music understanding evaluation [19–21] by including this new family of models and steering their early development towards robust progress.

2. RELATED WORK

In the music domain, Audio LLMs are commonly evaluated by assessing their text output in the context of a given task defined by an instruction template. Tasks are either designed to test whether the model is able to recognise predefined musical properties such as key (“*What is the key of this song?*”), genre, instrumentation, etc., or they probe for outputs that encompass a variety of musical concepts and that more closely resemble the dialogue format typical of chatbots. Tasks that fall under the former usually mirror canonical MIR tasks and their evaluation leverages standard metrics and benchmarks from the MIR literature. Evaluation of tasks that require broader understanding follows instead less established protocols. Prior works on Audio LLMs most commonly tackle this via two tasks, music captioning (“*Describe the contents of the provided audio in detail*”) [7–9, 11] and music question answering (“*What are some possible uses for this music in a film or TV show?*”) [8, 9]. To perform this kind of evaluation, the authors in [7, 9, 11] make use of the MusicCaps dataset [15], while others [8, 9] employ ad-hoc evaluation datasets created with the help of LLMs. In particular, Liu et al. [8] and Deng et al. [9] propose their own datasets for music question answering, MusicQA and MusicInstruct respectively. These are derived from captions in the MusicCaps dataset or tags from the MagnaTagATune dataset [22] (MusicQA only), by augmenting them into music-question pairs via pre-trained LLMs. Similarly to these works, we also leverage LLMs to generate our set of questions and answers, but we follow a multiple-choice format to ensure meaningful evaluation and validate all generated data through human annotators to guarantee high data quality.

Finally, we note that concurrent work also proposes evaluation benchmarks for music understanding in LLM-based models [23–25], but these all differ from our work in significant ways: MuChin [23] includes only text in Chinese and does not follow a multiple-choice format, while both MusicTheoryBench (MTB) and ZIQI-Eval focus on the symbolic domain and address the evaluation of text-

Benchmark	Size	Source(s)	Audio	HC	MC
MusicQA [8]	4.5k	MagnaTagATune	✓	✗	✗
MusicInstruct [9]	61k	MusicCaps	✓	✗	✗
ZIQI-Eval [25]	14k	Music books	✗	✗	✓
MTB [24]	372	(human-written)	✗	✓	✓
AIR-Bench [26]	400	MusicCaps	✓	✗	✓
MuChin [23]	1k	<i>unknown</i>	✓	✓	✗
MuChoMusic	1.2k	MusicCaps, SDD	✓	✓	✓

Table 1. Comparison of MuChoMusic to existing benchmarks. HC: human-curated, MC: multiple-choice.

based LLMs. AIR-Bench [26] includes a small subset of music-related tasks, but puts its focus on audio understanding more generally. We provide an overview of key differences with other benchmarks in Table 1.

3. MUCHOMUSIC

Through *MuChoMusic*, we aim to alleviate three prominent issues in the evaluation of music understanding in Audio LLMs: a lack of standardisation, the inadequacy of existing text generation metrics, and the quality of current evaluation sets. We address the first two by adopting a multiple-choice format, while our methodical generation and validation procedure attends to the third issue by grounding the data in human-written descriptions and ensuring that the final questions and answers are correct and contextually relevant, as judged by multiple annotators.

3.1 Overview

MuChoMusic consists of 1,187 multiple-choice questions aimed at testing the understanding of 644 unique music tracks sourced from the MusicCaps [15] and the Song Descriptor Dataset [16]. We adopt a multiple-choice format in order to standardise evaluation and follow widespread practice in LLM-centric evaluation scenarios [27–30]. As illustrated in Figure 1, each question has four possible answers. One option is the correct answer, the other three are distractors. Inspired by [31], we structure these as follows: one does not fit the track of interest but is related to the question (*incorrect but related*), one correctly fits the audio, but does not address the question (*correct but unrelated*), and one does not apply to the track and is also irrelevant to the question (*incorrect and unrelated*).

Evaluation dimensions *MuChoMusic* is built from a diverse set of musical works and their detailed descriptions, and serves as a foundation for evaluating Audio LLMs across various dimensions of music comprehension. To delineate the specific evaluation dimensions encompassed by our benchmark, we develop a taxonomy consisting of two primary categories: *knowledge* and *reasoning*. Each category is further divided into several dimensions, informed by insights from national music education programs and existing research on music folksonomies [32]. This structured approach allows us to assess the depth and breadth of music-related competencies systematically, offering a holistic view of models’ capabilities in the music domain.

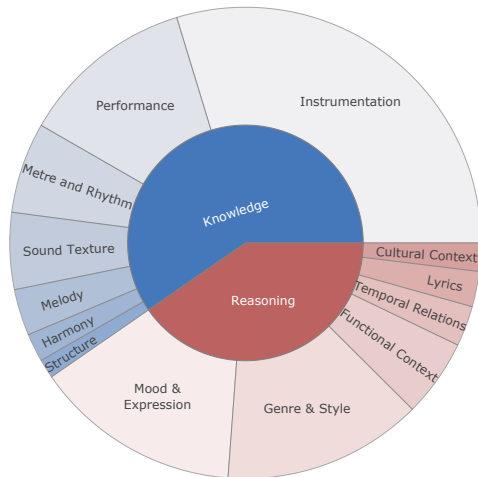


Figure 2. Distribution of evaluation dimensions covered by MuChoMusic across knowledge and reasoning.

In the *knowledge* category, questions probe a model’s ability to recognise pre-acquired knowledge across various musical aspects: (i) melody, (ii) harmony, (iii) metre and rhythm, (iv) instrumentation, (v) sound texture, (vi) performance, and (vii) structure. Questions that test *reasoning* are instead designed to require the synthesis and analytical processing of multiple musical concepts: (i) mood and expression, (ii) temporal relations between elements, (iii) interpretation of lyrics, (iv) genre and style, (v) historical and cultural context, and (vi) functional context. An example of reasoning might involve using an understanding of tempo, chord quality, and instrumentation in concert to ascertain the mood of a music piece. Each question can cover multiple dimensions and their categorisation is obtained automatically, as described in Section 3.2. Figure 2 shows the coverage of the two categories and their respective dimensions within the benchmark. Over half the questions test at least one aspect of musical knowledge, such as features relating to instrumentation or performance characteristics, while 44% are dedicated to probing reasoning skills. While the distribution of dimensions within each category is not balanced, we note that this reflects the distribution of different musical concepts within music captions [16], resulting in categories such as instrumentation, mood and genre appearing more frequently.

3.2 Dataset construction

To build our dataset, we automatically transform human-written music captions into multiple-choice questions. These are then carefully validated by multiple human annotators, alongside the associated audio, in order to filter out invalid, ambiguous or irrelevant questions resulting from inaccuracies or hallucinations in the model output.

Data sources We source our data from music caption datasets as we aim for elaborate and linguistically diverse information about the music. Currently, only two captioning datasets provide sufficiently detailed music descriptions, namely the Song Describer Dataset (SDD) and MusicCaps. SDD contains 2-minute-long music clips with

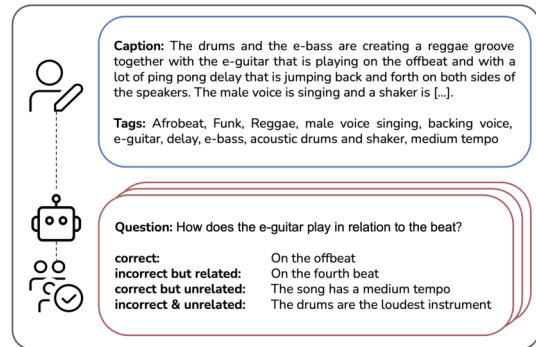


Figure 3. QA generation and validation pipeline. Example shown here is from MusicCaps [15].

single-sentence captions crowd-sourced from music enthusiasts, while the captions in MusicCaps, describing 10-second audio snippets, are written by professional musicians. From SDD, we select all tracks that have at least two captions, to ensure enough information is provided to the model to be able to formulate interesting and challenging questions. While this is not possible for the MusicCaps dataset, where only one caption is available for each track, we note that descriptions are, on average, longer than in SDD and designed to be more comprehensive. From the genre-balanced subset of the MusicCaps test split, we exclude all tracks for which the labels indicate a low recording quality, to prevent differences in audio quality from affecting the results. For both datasets, we employ a state-of-the-art genre tagging model [33] to identify non-musical tracks and to sub-sample songs from the most common genres (e.g. rock and electronic). Through this curation process, we select 227 unique tracks from SDD and 497 from MusicCaps. We supplement the descriptions with short text labels taken from the dataset itself in the case of MusicCaps and from a related dataset for SDD [34].

QA generation We generate the question-answer sets by instructing Gemini 1.0 Pro [3] to formulate question and answer options for a given human-written caption. To leverage the model’s in-context learning capability, we prompt it with a detailed task description and three examples of input (description and tags) and expected output. In addition to the question and answer pairs, we ask the model to start its output with a summary of the provided information about the music recording and to interleave the distractor answer options with explanations of their suitability. This way of prompting is inspired by the chain-of-thought methodology and helps to elicit the best model responses [35, 36]. This way, we obtain three multiple choice questions from each description on average and collect a total of 2,091 question-answer pairs. An example of the generated questions is shown in Figure 3.

Data validation In order to ensure that questions and answers in our benchmark are factually accurate, aptly written and that each question can be correctly answered based on the available audio, we validate all sets of questions via human annotators. For this step, we recruit 222 participants via the Prolific platform (www.prolific.com). During annotation, a question, the corresponding audio clip, and

all four answer options are presented to the participants in random order, for a total of 30 to 50 question items. Participants are then asked to select all options that correctly answer the question or skip the question by indicating that they are unable to provide an answer or that the question is not valid. Following this procedure, for each question, we collect three to five annotations, stopping early if different annotators are in agreement. This task setup is intended to vet questions and detect those that do not adhere to the intended multiple-choice format, either because the expected correct answer is not the only plausible option or because any one of the distractors is more likely. Consequently, we exclude questions from our final dataset for which i) less than 50% of the annotations indicate the intended correct answer or ii) more than 50% of the annotations mark any of the distractors as a plausible answer. The final dataset comprises 858 questions from MusicCaps descriptions and the remaining 329 from SDD captions.

Question categorisation Once questions are validated, we categorise them according to our taxonomy outlined in Section 3.1. To achieve this, we employ Gemini 1.0 Pro, this time prompting it to automatically label each question with one or more of the evaluation dimensions. The prompt includes the full taxonomy including detailed descriptions of all dimensions, a chain-of-thought instruction, and a single question with only the correct answer. The produced output contains an explanation of the categories and dimensions assigned to each question.

4. BENCHMARKING WITH MUCHOMUSIC

We now demonstrate the use of our benchmark, describing our proposed evaluation protocol and metrics, and then detailing our experiments on benchmarking Audio LLMs.

4.1 Evaluation Protocol

In multiple-choice-based evaluation, a model is provided with a question and a set of answer options, and is then tasked with selecting the most suitable answer. In practice, this can be accomplished in different ways [29]. In our experiments, we adopt *output-based* evaluation: given a music clip and an associated question-answer set, the language output produced by the model is mapped to one of the candidate options by string matching. Another common approach in MC evaluation is to determine the selected answer through the conditional log likelihood scores of the tokens forming each of the different options. While this can help estimate uncertainty and confidence in the model predictions, in our experiments, we explore only the output-based setting, for three reasons: (1) this corresponds to real-world use of the models, as interactions usually take the form of a conversation; (2) it has a lower computational cost; (3) prior work has demonstrated that sentence probabilities are not necessarily indicative of the probabilities assigned to the answers [37]. To extract the selected answer from the generated outputs, we match either the option identifier (*A*, *B*, *C* or *D*) or the full answer text, if one and only one is given in the output.

Model	Audio encoder	LLM
MusiLingo [9]	MERT [39]	Vicuna 7B [40]
MuLLaMa [8]	MERT [39]	LLaMA-2 7B [41]
M2UGen [12]	MERT [39]	LLaMA-2 7B [41]

SALMONN [11]	BEATS [42] & Whisper _{large-v2} [43]	Vicuna 7B [40]
Qwen-Audio [13]	Whisper _{large-v2} [43]	Qwen 7B [44]

Table 2. Overview of models we evaluate in our study.

Metrics We look at two main metrics to measure model performance on our benchmark: accuracy and instruction following rate (IFR). Accuracy is given by the percentage of correctly answered questions out of the total set of questions. IFR is given by the percentage of generated answers that correspond to one of the given options. In both cases, finegrained scores can be obtained by considering only the subset of questions covering at least one of the available evaluation dimensions shown in Figure 2.

Adaptation An important design factor in the evaluation of LLM-based models is adaptation [29], the process of adapting the input to a suitable format. While the format of the audio input is typically fixed by the model design, text inputs allow for more flexibility and different prompting techniques have been shown to significantly influence model’s behaviour [35, 36, 38]. Beyond simply passing the question and answer options as the input text, corresponding to *zero-shot prompting*, an effective alternative strategy is to leverage *few-shot in-context learning* (ICL), whereby the model is presented with a set of reference inputs that exemplify the task prior to being shown the question of interest. We experiment with in-context learning in our experiments, providing between 0 and 5 examples in the text input. In the interest of standardisation and to ensure a fair comparison between the models, unless otherwise specified, we keep the prompt selection fixed in our final experiments, following an initial exploration.

4.2 Models

In our evaluation, we consider three music-specific models, MuLLaMA [8], MusiLingo [9], and M2UGen [12], and two general-audio LLMs which can be applied to music, as reported in their respective papers, SALMONN [11] and Qwen-Audio [13]. To the best of our knowledge, these are all the existing Audio LLMs which can be applied to music and for which open-source weights are available. These all share a similar architectural design and are composed of a backbone LLM, an audio encoder and a lightweight learnable adapter module to align embeddings produced by the audio encoder to the input space of the LLM, based on either the LLaMA-adapter [45] (MuLLaMA, MusiLingo, M2UGen) or a Q-Former network [46] (SALMONN). An overview of the backbones used in each model is provided in Table 2. All systems are trained via instruction tuning [38, 47] and all employ a combination of different training datasets, often in multiple training stages including pre-training and fine-tuning. For all models, we follow the official implementation and use default

Model	Accuracy			IFR
	All	Knowledge	Reasoning	All
MusiLingo [9]	21.1	22.0	19.2	71.6
MuLLaMa [8]	32.4	32.3	31.3	79.4
M2UGen [12]	42.9	44.9	41.2	96.4
SALMONN [11]	41.8	41.0	43.3	99.8
Qwen-Audio [13]	51.4	51.1	51.0	89.7
<i>Random guessing</i>	25.0	25.0	25.0	100.0

Table 3. Overall benchmarking results.

inference settings. We repeat all experiments 3 times, randomly shuffling the order in which answer options are presented, and report average performance across all runs.

5. RESULTS AND DISCUSSION

In this section, we first presents findings from our benchmarking experiments, with the goal of elucidating the current state of music understanding in Audio LLMs. We then illustrate how MuChoMusic can be used to derive new insights via a diagnostic analysis, and discuss key takeaways.

5.1 Benchmarking Results

We report results for all models in Table 3, showing the overall accuracy score alongside detailed scores on knowledge and reasoning questions, and the instruction following rate (IFR). Figure 4 presents a breakdown of accuracy scores along all reasoning and knowledge dimensions. Unless otherwise specified, we show one-shot performance for all models, as we find this to be the overall optimal setting, as we discuss in more detail in Section 5.2. From this, we observe that current models generally perform poorly across all settings and along all evaluation dimensions. Among these, Qwen-Audio stands out with a score of 51.4%. Surprisingly, with the exception of M2UGen, music-specialised models generally perform worse than general-audio ones, in some cases performing only marginally above or even below random performance. As evidenced by the IFR, these models struggle to output answers in the correct format, which in turn negatively impacts their accuracy score. As shown later in Section 5.3, we find that, when none of the answer options is selected by the model, this is often due to *auditory hallucinations*, *language hallucinations* or *training biases*.

5.2 Analysis and Discussion

We now investigate factors influencing performance along different axes by using our benchmark as a diagnostic tool.

Are models sensitive to prompts? We first study the effect of varying the number of in-context examples. As shown in Figure 5, providing a single example is occasionally beneficial to accuracy and IFR, but with both the difference magnitude and overall impact differing between models. Additionally, this trend does not hold after the one-shot setting, and we see no consistent improvement when using a larger number of examples. Interestingly,

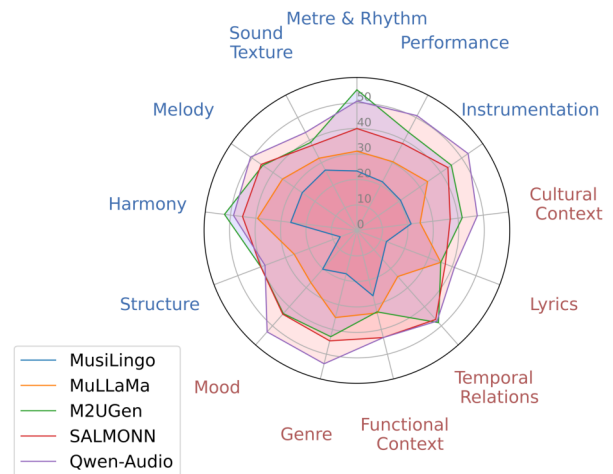


Figure 4. Finegrained accuracy across evaluation dimensions in knowledge (labelled in blue) and reasoning (red).

we observe that, for M2UGen, Qwen-Audio and MuLLaMa, changes in accuracy from zero- to one-shot prompts are accompanied by a reduction in variance, suggesting that ICL can help minimise variability in the model output. While we do not explore this in our experiments, we also hypothesise that the advantages of ICL may become more prominent through multimodal few-shot prompting [48, 49], which we leave for future work.

How do models respond to different distractors?

Next, we shift our attention to examining how distractors in our benchmark influence the difficulty of the task. To this end, we ablate answer options corresponding to the different kinds of distractors described in Section 3.2, and present the model with only two or three answer options. In Figure 6(a) we show how performance is affected when using only one distractor alongside the correct option, always randomising their order. From this, we observe that the two distractors containing information which is not related to the question (CU and IU) have a similar effect, while including the *incorrect but related* (IR) option consistently makes the task more challenging. This phenomenon persists when adding a second distractor (not shown here), with combinations which include IR invariably leading to worse performance. Intuitively, the two *unrelated* options can be ruled out based on the text input only, while selecting the correct answer between two options that appear relevant requires engaging multimodal understanding to relate information in the audio content to the text in the question. Crucially, this indicates that models particularly struggle to discern between options that are equally plausible based on the text input only, suggesting that less attention is given to the audio content. This forms the basis of our hypothesis that current Audio LLMs are characterised by a strong language bias, leading to poor performance in tasks that are more audio-dependent. We test this hypothesis in the next section.

Do models actually pay attention to the audio? In order to verify whether the audio input is effectively being ignored or is overshadowed by its text counterpart, we de-

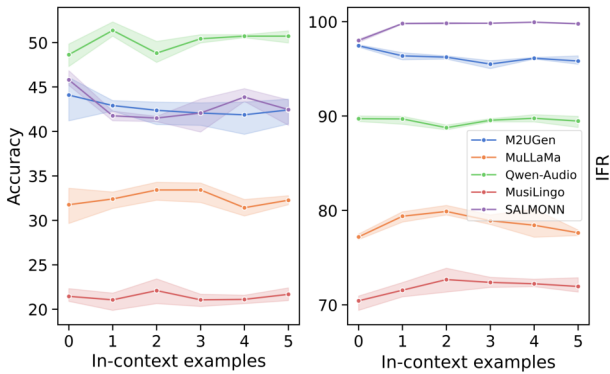


Figure 5. Effect of the number of in-context examples on accuracy (left) and instruction-following rate (right).

vis a simple test, which we call *audio attention test*, where we replace the audio clip corresponding to a given question with either white Gaussian noise or a randomly chosen track from the dataset. In order to pass this test, a model should display a statistically significant drop in performance when either form of audio perturbation is used, compared to its baseline performance. We showcase results on this test in Figure 6(b). From this, we clearly see that, with the exception of SALMONN and Qwen-Audio, all models fail the audio attention test, and the severity of this failure is often negatively correlated to their overall performance on the benchmark (see Table 3). This confirms that current Audio LLMs are biased towards textual information, often choosing answers that score well under their language prior. Additionally, it provides an explanation for their low performance on the benchmark, as this is effectively bounded by the maximum score they can attain mostly based on the language input. We argue that this constitutes a major pitfall in the design and training procedure of these models, which results in music understanding abilities that do not match the expected performance, as obtained through prior evaluations.

5.3 Failure Modes

While the core goal of our benchmark is to provide standardised automatic evaluation to objectively measure general music understanding capabilities, we argue that it can also constitute a useful tool for qualitative assessment. We showcase three examples here, focusing on the two lowest-performing models. While this is not an exhaustive analysis, these examples offer a bird’s-eye view of how language pre-training biases percolate through multimodal training, resulting in failures to attend to the inputs in our evaluation. To describe these, we borrow terminology from [50].

Auditory hallucination One of the ways models fail to provide a suitable answer falls under the category of *auditory hallucination*, whereby a response includes references to musical elements that are not present in the audio. For example, when asked about an accompaniment instrument, models with this type of hallucination may ignore any suitable option provided (“*acoustic guitar*” or “*strings*”), instead answering “*The song is accompanied by a piano.*”, when the audio clip clearly contains no piano.

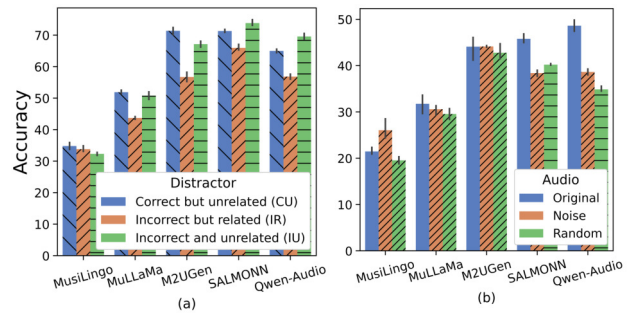


Figure 6. (a) Effect of using different types of distractors: models tend to perform worst when tasked with distinguishing between two related answers. (b) Audio attention test: only some models display a significant drop in performance when provided with incorrect audio inputs. For these experiments, we adopt zero-shot prompting.

Language hallucination Another instance of hallucination concerns mundane statements that deviate from the topic of the question altogether. Among others, an observed case of this failure mode is a statement of the form “*The song has a clear and coherent rhythm structure*” to a question specifically asking about the “*type of drum beat*”.

Training data bias The last failure mode we encounter is related to a bias towards frequent patterns occurring in the training data. While some of the benchmarked models undergo a stage of training that includes instruction-tuning examples with questions and answers, occasionally they still produce trivial outputs. For example, when asked “*What is the intended purpose of this song?*”, a model with this type of bias may answer “*The intended purpose of this song is not mentioned in the caption*”. Reviewing MusicQA, used in training MuLLaMa and MusiLingo, reveals that a high number of the LLM-generated training examples mention similar phrases, thus likely biasing the model towards this type of uninformative but highly likely output.

6. CONCLUSION

We have presented MuChoMusic, a multiple-choice music question answering benchmark designed to test music understanding in Audio LLMs. From an evaluation of five state-of-the-art systems, we find that our benchmark acts as a challenging and informative test, and that current models do not yet leverage both the audio and text modalities fully. All questions in our benchmark are synthesised from human-written music descriptions and manually reviewed to guarantee high data quality. A categorisation of the questions highlights that MuChoMusic offers a broad coverage of areas targeted by current models, and additionally pinpoints gaps that could guide future developments in the field. While we demonstrate that our approach leads to new insights, we note that the multiple-choice format presents some limitations [51]. Therefore evaluation on MuChoMusic should be complemented via further benchmarking efforts to address additional aspects of music understanding through different tasks and formats.

7. ETHICS STATEMENT

7.1 Annotator welfare

Prior to participation, the annotation experiment described in Section 3.2 was approved by the Queen Mary Ethics of Research Committee to ensure alignment with ethical guidelines and protections for human subjects in research. We did not collect any personal data from our annotators, safeguarding their privacy and confidentiality. Annotators were fully informed about the objectives of the research, the nature of their tasks, and the use of their annotations, underpinning their informed consent before contributing to the project. In an effort to provide a fair compensation for their contributions, annotators were paid £9 per hour.

7.2 Biases and fairness

In constructing the MuChoMusic benchmark, our data collection strategy included sourcing music tracks from a variety of backgrounds, acknowledging the inherent challenges in representing the rich diversity of global music cultures within our dataset. We recognise that our initiative does not fully balance the benchmark across all genres, languages, and cultural backgrounds, and annotations were conducted exclusively in English due to logistical constraints, highlighting areas for future expansion and improvement.

8. ACKNOWLEDGEMENTS

IM is a research student at the UKRI Centre for Doctoral Training in Artificial Intelligence and Music, supported jointly by UK Research and Innovation [grant number EP/S022694/1] and Universal Music Group. EB is supported by RAEng/Leverhulme Trust research fellowship LTRF2223-19-106.

9. REFERENCES

- [1] Rishi Bommasani et al. “On the opportunities and risks of foundation models”. In: *arXiv preprint arXiv:2108.07258* (2021).
- [2] Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, Qixiang Ye, and Furu Wei. “Grounding Multimodal Large Language Models to the World”. In: *The Twelfth International Conference on Learning Representations*. 2024.
- [3] Gemini Team et al. “Gemini: a family of highly capable multimodal models”. In: *arXiv preprint arXiv:2312.11805* (2023).
- [4] OpenAI et al. “Gpt-4 technical report”. In: *arXiv preprint arXiv:2303.08774* (2023).
- [5] Jean-Baptiste Alayrac et al. “Flamingo: a Visual Language Model for Few-Shot Learning”. In: *Advances in Neural Information Processing Systems*. 2022.
- [6] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. “Qwen-vl: A frontier large vision-language model with versatile abilities”. In: *arXiv preprint arXiv:2308.12966* (2023).
- [7] Josh Gardner, Simon Durand, Daniel Stoller, and Rachel Bittner. “LLark: A Multimodal Instruction-Following Language Model for Music”. In: *Proceedings of the 41st International Conference on Machine Learning (ICML)*. 2024.
- [8] Shansong Liu, Atin Sakkeer Hussain, Chen-shuo Sun, and Ying Shan. “Music Understanding LLaMA: Advancing Text-to-Music Generation with Question Answering and Captioning”. In: *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2024.
- [9] Zihao Deng, Yinghao Ma, Yudong Liu, Rongchen Guo, Ge Zhang, Wenhui Chen, Wenhao Huang, and Emmanouil Benetos. “MusiLingo: Bridging Music and Text with Pre-trained Language Models for Music Captioning and Query Response”. In: *Findings of the Association for Computational Linguistics: NAACL 2024*. Association for Computational Linguistics, 2024.
- [10] Soham Deshmukh, Benjamin Elizalde, Rita Singh, and Huaming Wang. “Pengi: An Audio Language Model for Audio Tasks”. In: *Thirty-seventh Conference on Neural Information Processing Systems*. 2023.
- [11] Changli Tang, Wenyi Yu, Guangzhi Sun, Xianzhao Chen, Tian Tan, Wei Li, Lu Lu, Zejun MA, and Chao Zhang. “SALMONN: Towards Generic Hearing Abilities for Large Language Models”. In: *The Twelfth International Conference on Learning Representations*. 2024.
- [12] Atin Sakkeer Hussain, Shansong Liu, Chenshuo Sun, and Ying Shan. “M²UGen: Multi-modal Music Understanding and Generation with the Power of Large Language Models”. In: *arXiv preprint arXiv:2311.11255* (2023).
- [13] Yunfei Chu, Jin Xu, Xiaohuan Zhou, Qian Yang, Shiliang Zhang, Zhijie Yan, Chang Zhou, and Jingren Zhou. “Qwen-audio: Advancing universal audio understanding via unified large-scale audio-language models”. In: *arXiv preprint arXiv:2311.07919* (2023).
- [14] Yonatan Bitton, Hritik Bansal, Jack Hessel, Rulin Shao, Wanrong Zhu, Anas Awadalla, Joshua P. Gardner, Rohan Taori, and Ludwig Schmidt. “VisIT-Bench: A Dynamic Benchmark for Evaluating Instruction-Following Vision-and-Language Models”. In: *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*. 2023.

- [15] Andrea Agostinelli et al. “Musiclm: Generating music from text”. In: *arXiv preprint arXiv:2301.11325* (2023).
- [16] Ilaria Manco et al. “The Song Describer Dataset: a Corpus of Audio Captions for Music-and-Language Evaluation”. In: *Machine Learning for Audio Workshop at NeurIPS 2023*. 2023.
- [17] Daniel McKee, Justin Salamon, Josef Sivic, and Bryan Russell. “Language-Guided Music Recommendation for Video via Prompt Analogies”. In: *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2023.
- [18] SeungHeon Doh, Keunwoo Choi, Jongpil Lee, and Juhan Nam. “LP-MusicCaps: LLM-Based Pseudo Music Captioning”. In: *Proceedings of the 24th International Society for Music Information Retrieval Conference, ISMIR 2023*. 2023.
- [19] Rachel M. Bittner, Magdalena Fuentes, David Rubinstein, Andreas Jansson, Keunwoo Choi, and Thor Kell. “mirdata: Software for Reproducible Usage of Datasets”. In: *Proceedings of the 20th International Society for Music Information Retrieval Conference, ISMIR 2019*. 2019.
- [20] Ruibin Yuan et al. “MARBLE: Music Audio Representation Benchmark for Universal Evaluation”. In: *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*. 2023.
- [21] Christos Plachouras, Pablo Alonso-Jiménez, and Dmitry Bogdanov. “mir_ref: A Representation Evaluation Framework for Music Information Retrieval Tasks”. In: *37th Conference on Neural Information Processing Systems (NeurIPS), Machine Learning for Audio Workshop*. 2023.
- [22] Edith Law, Kris West, Michael Mandel, Mert Bay, and J. Stephen Downie. “Evaluation of algorithms using games: The case of music tagging”. In: *Proceedings of the 10th ISMIR Conference*. 2009.
- [23] Zihao Wang, Shuyu Li, Tao Zhang, Qi Wang, Pengfei Yu, Jinyang Luo, Yan Liu, Ming Xi, and Kejun Zhang. “MuChin: A Chinese Colloquial Description Benchmark for Evaluating Language Models in the Field of Music”. In: *arXiv preprint arXiv:2402.09871* (2024).
- [24] Ruibin Yuan, Hanfeng Lin, Yi Wang, Zeyue Tian, Shangda Wu, Tianhao Shen, Ge Zhang, Yuhang Wu, Cong Liu, Ziya Zhou, et al. “Chatmusician: Understanding and generating music intrinsically with llm”. In: *arXiv preprint arXiv:2402.16153* (2024).
- [25] Jiajia Li, Lu Yang, Mingni Tang, Cong Chen, Zuchao Li, Ping Wang, and Hai Zhao. “The Music Maestro or The Musically Challenged, A Massive Music Evaluation Benchmark for Large Language Models”. In: *arXiv preprint arXiv:2406.15885* (2024).
- [26] Qian Yang, Jin Xu, Wenrui Liu, Yunfei Chu, Ziyue Jiang, Xiaohuan Zhou, Yichong Leng, Yuanjun Lv, Zhou Zhao, Chang Zhou, et al. “AIR-Bench: Benchmarking Large Audio-Language Models via Generative Comprehension”. In: *arXiv preprint arXiv:2402.07729* (2024).
- [27] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. “Measuring Massive Multitask Language Understanding”. In: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. 2021.
- [28] Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. “Agieval: A human-centric benchmark for evaluating foundation models”. In: *arXiv preprint arXiv:2304.06364* (2023).
- [29] Percy Liang et al. “Holistic Evaluation of Language Models”. In: *Transactions on Machine Learning Research* (2023).
- [30] Aarohi Srivastava et al. “Beyond the Imitation Game: Quantifying and extrapolating the capabilities of language models”. In: *Transactions on Machine Learning Research* (2023).
- [31] Yevgeni Berzak, Jonathan Malmaud, and Roger Levy. “STARC: Structured Annotations for Reading Comprehension”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2020.
- [32] Mohamed Sordo, Fabien Gouyon, Luís Sarmiento, Óscar Celma, and Xavier Serra. “Inferring Semantic Facets of a Music Folksonomy with Wikipedia”. In: *Journal of New Music Research* 42.4 (2013).
- [33] Pablo Alonso-Jiménez, Xavier Serra, and Dmitry Bogdanov. “Music Representation Learning Based on Editorial Metadata from Discogs”. In: *Proceedings of the 23rd International Society for Music Information Retrieval Conference, ISMIR 2022, Bengaluru, India, December 4-8, 2022*. 2022.
- [34] Dmitry Bogdanov, Minz Won, Philip Tovstogan, Alastair Porter, and Xavier Serra. “The MTG-Jamendo Dataset for Automatic Music Tagging”. In: *Machine Learning for Music Discovery Workshop, International Conference on Machine Learning (ICML)*. 2019.
- [35] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. “Large Language Models are Zero-Shot Reasoners”. In: *Advances in Neural Information Processing Systems*. 2022.

- [36] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. “Chain of Thought Prompting Elicits Reasoning in Large Language Models”. In: *Advances in Neural Information Processing Systems*. 2022.
- [37] Joshua Robinson and David Wingate. “Leveraging Large Language Models for Multiple Choice Question Answering”. In: *The Eleventh International Conference on Learning Representations*. 2023.
- [38] Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. “Finetuned Language Models are Zero-Shot Learners”. In: *International Conference on Learning Representations*. 2022.
- [39] Yizhi LI et al. “MERT: Acoustic Music Understanding Model with Large-Scale Self-supervised Training”. In: *The Twelfth International Conference on Learning Representations*. 2024.
- [40] Wei-Lin Chiang et al. *Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality*. 2023.
- [41] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajwal Bhargava, Shruti Bhosale, et al. “Llama 2: Open foundation and fine-tuned chat models”. In: *arXiv preprint arXiv:2307.09288* (2023).
- [42] Sanyuan Chen, Yu Wu, Chengyi Wang, Shujie Liu, Daniel Tompkins, Zhuo Chen, Wanxiang Che, Xiangzhan Yu, and Furu Wei. “BEATs: Audio Pre-Training with Acoustic Tokenizers”. In: *Proceedings of the 40th International Conference on Machine Learning*. Vol. 202. Proceedings of Machine Learning Research. PMLR, 2023.
- [43] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine Mcleavey, and Ilya Sutskever. “Robust Speech Recognition via Large-Scale Weak Supervision”. In: *Proceedings of the 40th International Conference on Machine Learning*. Vol. 202. Proceedings of Machine Learning Research. PMLR, 2023.
- [44] Jinze Bai et al. “Qwen Technical Report”. In: *arXiv preprint arXiv:2309.16609* (2023).
- [45] Renrui Zhang, Jiaming Han, Chris Liu, AoJun Zhou, Pan Lu, Hongsheng Li, Peng Gao, and Yu Qiao. “LLaMA-Adapter: Efficient Fine-tuning of Large Language Models with Zero-initialized Attention”. In: *The Twelfth International Conference on Learning Representations*. 2024.
- [46] Yiren Jian, Chongyang Gao, and Soroush Vosoughi. “Bootstrapping Vision-Language Learning with Decoupled Language Pre-training”. In: *Thirty-seventh Conference on Neural Information Processing Systems*. 2023.
- [47] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. “Self-Instruct: Aligning Language Models with Self-Generated Instructions”. In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 2023.
- [48] Sivan Doveh, Shaked Perek, M Jehanzeb Mirza, Amit Alfassy, Assaf Arbel, Shimon Ullman, and Leonid Karlinsky. “Towards multimodal in-context learning for vision & language models”. In: *arXiv preprint arXiv:2403.12736* (2024).
- [49] Haozhe Zhao, Zefan Cai, Shuzheng Si, Xiaojian Ma, Kaikai An, Liang Chen, Zixuan Liu, Sheng Wang, Wenjuan Han, and Baobao Chang. “MMICL: Empowering Vision-language Model with Multi-Modal In-Context Learning”. In: *The Twelfth International Conference on Learning Representations*. 2024.
- [50] Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, et al. “HallusionBench: an advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024.
- [51] Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. “Large Language Models Are Not Robust Multiple Choice Selectors”. In: *The Twelfth International Conference on Learning Representations*. 2024.

HUMAN POSE ESTIMATION FOR EXPRESSIVE MOVEMENT DESCRIPTORS IN VOCAL MUSICAL PERFORMANCES

Sujoy Roychowdhury Preeti Rao Sharat Chandran
{214077004, prao, sharat} @ iitb.ac.in
Indian Institute of Technology Bombay

ABSTRACT

Vocal concerts in Indian music are invariably associated with the performers’ hand gesticulations that are believed to convey emotion, music semantics as well as the individual style of the performers. Video recordings, with one or more cameras, along with markerless human pose estimation algorithms can be employed to capture such movements, and thus potentially solve music information retrieval (MIR) queries. Nevertheless, off-the-shelf algorithms are built for the most part for upright human configurations contrasting with seated positions in Indian vocal concerts and the upper body movements in the context of performing music. Current state-of-the-art algorithms are black box neural network based and this calls for an investigation of the components of such algorithms. Key decisions involve the choice of one or more cameras, the choice of 2D or 3D features, and relevant parameters such as confidence thresholds in common machine learning methods. In this paper, we quantify the increase in the performance with three cameras on two music information retrieval tasks. We offer insights for single and multi-view processing of videos.

1. INTRODUCTION

Performances of vocal music in the Indian classical traditions involve the use of hand gestures that accompany the singing. We therefore wish to perform the automated analysis of performances with audiovisual recordings. One or more video cameras can be used to record musical performances. Our goal in this work is to explore different Human Pose Estimation (HPE) methods for the computational analysis of expressive movements of upper body limbs of the vocalist.

Markerless Human Pose Estimation algorithms constitute a novel technology that is available to investigate hand gestures by looking at important *keypoints* such as wrists and elbows. These algorithms are trained with artificial deep neural networks on whole body movements, and occasionally on music recordings. One important concern in



Figure 1: We analyze seated vocalists with multiple cameras. We identify singers purely based on hand gestures, and predict stable notes.

the use of HPE in musical gesture studies is that the gestures are typically expressive movements and not routine motor movements such as walking, jumping, or performing yoga poses, the latter motor movements being the bulk of the training data used in the development of HPE algorithms. Indian classical music, in both Northern and Southern traditions, is particularly rich in the use of gestures invariably in a seated position.

Paschalidou [1] studied associations between sound and “effort” in gesture in *Dhrupad* performances using an optical motion capture system. Although she finds correspondences, generalizing to multiple singers was challenging. Pearson and Pouw [2] look at vocal-gesture coupling in Karnatak music performance; the Kinect camera and an older, machine learning technique is used for obtaining keypoints. With the current deep-learning HPE technology, Clayton et. al. [3,4] use OpenPose-based wrist keypoints to classify *raga* and identify singers on a dataset of multimodal Hindustani music recordings. Nadkarni et al. [5] also use OpenPose to explore the correspondence between vocal singing and gestures.

However, to the best of our knowledge, there has been no work which uses multiple camera views for studying gesture and vocal correspondence in music. While it is natural to expect that more cameras help HPE, on a careful examination of prior work, we see that the process of estimating keypoints requires multiple design decisions. Several options present themselves in terms of camera position, the number of views, the keypoint detection method and its parameters, and finally, methods for combining information from multiple camera views.

1.1 Scope of this paper

In this paper, we choose 3 recent models for keypoint detection and 3 different HPE methods to obtain information from multiple views for the purposes of analyzing gestures. We restrict our study to wrist and elbow of both hands since they appear to be the ones most relevant when singers are seated. We consider two MIR tasks.

Stable Note Prediction from Gestures This problem was studied by Nadkarni et al. [5]. The authors define a stable note as a region of at least 250 ms duration across which the singer’s pitch lies within a 25 cent interval of the mean intonation of the raga note.

Gesture-based Singer Identification In this task, the goal is to identify the singer purely from gestures, i.e., without accessing the audio stream, or the face. Rahaim [6] emphasises that gestures in music are not taught or rehearsed and therefore tend to be idiosyncratic. The proposed MIR task attempts to validate the hypothesis that it should be possible to identify the singer from the gesture using 12s randomly chosen snips from the video. Similar problems may be interesting in other MIR settings such as a western music conductor’s motions when the face is not visible, or in dance and musical performances where the face is hopelessly masked. A gesture-based singer identification system can also be used to validate a digital avatar system that is attempting to realistically mimic singers.

1.2 Contributions

Although our work is focused on the two tasks mentioned above, we offer insights more generally useful in the HPE analysis of multiview recordings of music performances.

- Every HPE algorithm provides confidence scores. We suggest an approach to the choice of confidence thresholds for fair comparisons across algorithms.
- Multiple cameras lend themselves to 3D reconstruction, and indeed a single camera can also be used in recent state-of-the-art methods to infer 3D. We suggest that decision fusing 2D information from multiple cameras can be almost as competitive as using 3D reconstruction.

In term of concrete results for the two problems we report the following:

1. By considering position coordinates and individual coordinates of velocity and acceleration as features, a systematic choice of confidence thresholds, and the best HPE method, we improve the performance of stable note prediction (from gestures) from $\sim 66\%$ [5] to $\sim 83\%$ (single camera).
2. We report the accuracy of the best performing HPE method for gesture-based singer identification (8-way) to be $\sim 93\%$.

It is to be noted that the two MIR tasks are solved using two different methods. The Stable Note problem uses the

classical machine learning method of SVM. There is no artificial neural network here. On the other hand, the gesture-based singer identification problem uses a deep neural network with an inception block.

The rest of the paper is organized as follows. In Sec. 2 we look at different keypoint detection methods, and the reported performance. Sec. 3 describes the dataset used. Task agnostic comparisons of HPE methods is discussed in Sec. 4. Our two suggested methods of consuming information from multiple cameras is described in Sec. 5. The details of our experiments and the results are reported in Sec. 6. A summary appears in Sec. 7.

2. BACKGROUND

We first briefly describe three popular HPE methods, one [7] of which is *proprietary*. Later we describe the applications of these to areas in sports, and medicine in order to understand current understanding of their usage. We are not aware of the direct use of HPE for gesture understanding except the ones mentioned in the introduction.

2.1 Human Pose Estimation Techniques

The pose of a human in HPE methods results in a stick diagram (similar to Fig. 1) of important joints such as the shoulder, wrist, elbow, hip, knee and so on from images and videos. At the turn of the century, the joints, referred to as *keypoints* were obtained with markers placed on different parts of the body — however this can only be set up in controlled experimental settings and may also affect the natural movement of the subject. Subsequently specialized cameras such as the Kinect was employed using classical machine learning techniques. With the advent of deep learning (DL), nowadays standard RGB cameras may be employed for markerless pose estimation.

One of the first DL-based techniques is OpenPose [8] which can identify 25 keypoints in terms of pixel coordinates reported as (x, y) . OpenPose is based on estimating confidence heatmaps for keypoints and part affinity fields (PAF) which are vector fields encoding the connection across limbs between different joints. Since their method estimates keypoints and parts directly from the image using a multi-stage Convolutional Neural Network (CNN) their method is called a bottom-up approach in the literature [9]. OpenPose is trained on the MP-II human dataset [10] and the COCO [11] datasets. The MP-II dataset, with 25K images in 20 activity categories like cycling, running, violin playing, etc., has both full body and seating position data. The COCO dataset has 200K annotated images of 17 body keypoints in both seated and full body positions.

Alternatively, approaches based on Mask R-CNN [12] perform semantic segmentation of the image to identify masks on people in the image. Detectron2 [13] uses this mask to identify 17 keypoints for the body parts. As this method uses an identified mask for the prediction of the keypoints, this is often referred to as a top-down keypoint estimation method. Detectron2 is trained on the COCO dataset [11].

Multiple calibrated cameras can use well understood geometric computer vision methods of the 90s for depth estimation from 2D keypoints, thus producing 3D coordinates (x,y,z). Aniposelib [14] is a library which implements the 3D reconstruction from multiple synchronized calibrated cameras. However, the 2D keypoints in Detectron2 can be extended to 3D by a different deep learning based model VideoPose3D [15]. Videopose3D uses two DL models – a temporal dilated convolution for estimating depth per person and a separate 3D trajectory model for the center of the body viz. center-hip coordinate. Depth is estimated as a distance with respect to the center-hip of the body. Videopose3D is trained on Human 3.6M [16] and Human Eva [17] datasets.

There is, however, a direct way of obtaining 3D provided the face of the image is visible. BlazePose [18] identifies thirty-three 3D keypoints from single-view images. This model is trained on a custom dataset consisting of 60K images and is used in the Mediapipe library [7].

2.2 HPE-based Applications

A number of studies (for example, [19]) in HPE have involved evaluation of the accuracy of the HPE models by comparing markerless pose estimation with marker based pose estimation and shown that the Mean Absolute Error to be less than 30mm on 80% of trials.

Markerless systems are evaluated in clinical settings by Zhang et al. [20] and Mroz et al. [21] where they compare Hyperpose [22] vs OpenPose and OpenPose vs BlazePose (Mediapipe) respectively. Zhang et al. [20] establishes that OpenPose is better than HyperPose using manual annotations and then compares OpenPose with BlazePose via Root Mean Squared Error (RMSE) and correlation metrics. Their findings are that while BlazePose is faster, OpenPose provides for more accurate results in their setting. A similar comparison study [23] between three models – OpenPose, BlazePose and AlphaPose looks at a multi-camera setting for estimating biomechanical parameters like Ground Reaction Force (GRF). They observe that the detection rate is dependent on the camera view and the model. Also, they observe the BlazePose has lower detection rate than the other models.

Mehdizadeh et al. [24] look at estimating gait variables comparing OpenPose, AlphaPose [25] and Detectron and do not find any differences between their correlation with gait variables. Since all of the HPE models estimate a confidence score, they choose a confidence threshold for each model independently and discard estimates with a lower confidence than threshold and interpolate the values linearly. They choose the confidence thresholds so that less than 10% of frames were interpolated as a result.

Evaluating athlete anterior cruciate ligament (ACL) injury risk in jumps is important for athletes and the studies by Blanchard et. al [26] and Roygaga et al. [27] look at this using a multi-view camera setting and OpenPose model for HPE. They train models to identify if the jump is erroneous on each view independently and also a fusion model combining the individual models. Their choice of a confidence

threshold of 0.3 for OpenPose confidence is validated by an ablation study across different thresholds. Their results indicate that the task performance depends on the view and the type of error. They drop frames below threshold of 0.3 but do not interpolate dropped frames.

2.3 Synopsis

All of these studies bring us to some conclusions which motivate our research in MIR space. First, we are aware of a variety of techniques and approaches for HPE estimation and 3D reconstruction. Second, markerless pose estimation models have acceptable accuracy This is necessary in a musical setting since performers may not be comfortable using markers. Third, we realize the possible benefits of doing multi-view reconstruction in downstream tasks. We understand that performance on any downstream task depends on the view and model of choice. Finally, we are aware of the importance of the choice of thresholds in rejecting or retaining HPE estimates and how these are dependent on the model and view in question.

3. DATASET

We used the dataset from our earlier study [5]. The details of the dataset, data processing and links to download the data are available on github.¹ The dataset consists of 11 professional singers singing 2 *alaps*² each of 9 *ragas*. Recordings are captured by 3 synchronized cameras. However, we discovered that the recordings for 3 of the 11 singers were done with uncalibrated cameras and thus, since we are interested in 3D information, Anipose [14] cannot be used. Therefore we base our MIR tasks described in Sec. 1.1 on only the remaining 8 singers. We are left with 143 recordings with about 7 hours 10 mins of recording. These recordings are at 24 fps with a resolution of 1920 × 1080. The angle between the front and the right camera is approximately 55 degrees and the front and left camera is approximately 47 degrees. We refer to left and right camera based on the singer’s point of view. Fig. 1 shows a sample of the singer in the three views.

4. TASK-AGNOSTIC COMPARISONS

We choose the three HPE models because they provide a mix of bottom-up (OpenPose), top-down (Detectron) single view 2D keypoint estimation as well as single view 3D keypoint estimation (Mediapipe). In addition, our reconstruction techniques involve both frame-wise geometric reconstruction via Anipose [14] as well as DL-based methods (Videopose3D [15]) which uses information from neighbouring frames. Thus our methods of estimation and reconstruction are relatively independent of each other.

4.1 Confidence Threshold

All HPE models provide a confidence score for each of the estimated keypoints and it is conventional to choose

¹ Dataset github

² Alap is the unmettered introduction in raga performances.

a threshold for the confidence score to ignore predictions with a lower confidence score. Various previous studies [5, 27–29] have used a 0.3 threshold for OpenPose. However, we find that this method is not based on the actual data distribution and also cannot be extended to other HPE models like Detectron and Mediapipe. Due to this, we approach the problem similar to [24]. In every frame, if any of the left and right wrist and elbow keypoints have a confidence score to be less than some value x then we remove the position coordinate in question and interpolate it from the available neighbouring frames. For each model-view combination we change the threshold from 0 to 1 in steps of 0.01 and, in line with [24], choose the threshold so that no more than 10% of frames are dropped in that model-view combination. The corresponding obtained confidence thresholds are given in Tab. 2 and used for all our experiments. Abbreviations used in this paper are in Tab. 1.

OpenPose - OP2	Detectron - DE2	Mediapipe - MP2
Aniposelib - AP3	Videopose3D - VP3	Mediapipe3D - MP3

Table 1: Abbreviations for the various HPE techniques.

View	OP2	DE2	MP3
Front	0.49	0.17	0.27
Left	0.20	0.10	0.01
Right	0.38	0.15	0.12

Table 2: Confidence values obtained when the maximum number of interpolated frames is 10%.

4.1.1 Observations

We observe that thresholds for the left view are lower in all the 3 HPE models and this indicates that the keypoints are predicted with lower confidence for this view. The obtained threshold is particularly low for Mediapipe. Also, we observe that thresholds for OpenPose are higher than the other models indicating that OpenPose predicts keypoints with a higher confidence score. If we use the previously reported threshold of 0.3 for OpenPose then we would have 3.67%, 15.49% and 7.23% of interpolated frames in front, left and right views respectively. However, as seen from Tab. 5, there is no particular advantage of using the previous reported confidence value of 0.3 with its performance lower than what we have in Tab. 4.

4.2 Correspondence between models

Given that the models are attempting to predict the same joints, we expect that that the predictions would be close to each other in the pixel coordinate system. To verify this we consider the Euclidean distance of the predicted keypoints between 2 models in a pair in every frame. We ignore frames where any of the four keypoints have a confidence less than the corresponding threshold as defined in Tab. 2. The results are presented in Fig. 2.

4.2.1 Observations

We observe that the three models correspond well to each other in the front view (noting that the dimensions of the

frame to be 1920×1080). However, in other views, while OpenPose and Detectron predictions maintain pair-wise consistency, the same cannot be said for Mediapipe. We repeat these experiments by choosing thresholds corresponding to 5% and 15% interpolated frames and the same trends hold true. These trends also hold when we study the pair-wise correlations (instead of RMSE distances). This analysis, however, can only say that the models concur with each other in the front view but not so in the other views with least concurrence in the left view. We cannot conclude that one model is better than the other based on this analysis. A partial intuition for these results is that the left hand obscures the right hand in the left view, and most singers are right-handed.

5. MULTIPLE CAMERAS

In this section, we provide the details of the use of multiple cameras for the downstream MIR tasks. Fig. 3 shows the algorithms we use to get 2D and 3D coordinates.

5.1 Reconstruction

Reconstruction involves the combination of the data from multiple views to estimate a depth-coordinate either via classical computer vision [14] or DL. The z-coordinate is measured in distance from the camera in geometric reconstruction. On the other hand, with DL the depth is estimated to be a distance with respect to the center hip with larger values indicating further distance from the center. In the recording setting, a higher value of the estimated z-coordinate (e.g., for an outstretched hand) would mean closer to the camera. Mediapipe which predicts the z-coordinate from a single view uses a similar definition of the z-coordinate. Reconstruction using both Aniposelib [14] and Videopose3D can be done using any of the cameras as reference view and information from other cameras used for reconstruction. The results for the downstream tasks can be different.

5.2 Model Fusion

The second method for consuming data from multiple views in a machine learning based MIR setting is to have the downstream task (e.g. classifiers) trained individually on each view and then use the probability predictions of these classifiers as an input to a further classifier. This can be done based on classifiers on three sets (each using 2D data) in which case this will be an alternative to reconstruction. On the other hand, one can use classifiers trained on three reference views using 3D reconstructed data (anipose, videopose3D) or predicted data (e.g. Mediapipe), and then use their probability outputs as an input to a further model. Both of these approaches are examples of multi-view fusion which exploit the complementary information present in different views.

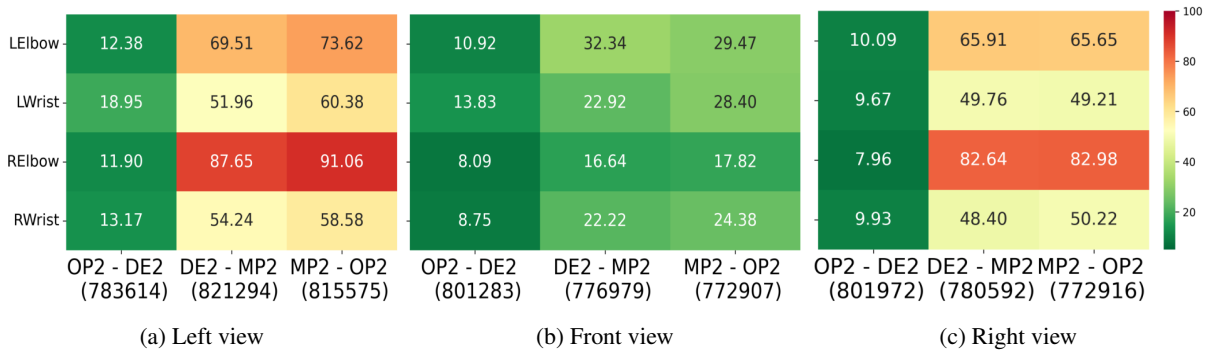


Figure 2: The average Euclidean distance in pixel coordinates different keypoints for pairwise HPE techniques. Non-interpolated frames considered are shown in parenthesis and four joints are considered. See Table 1 for the legend.

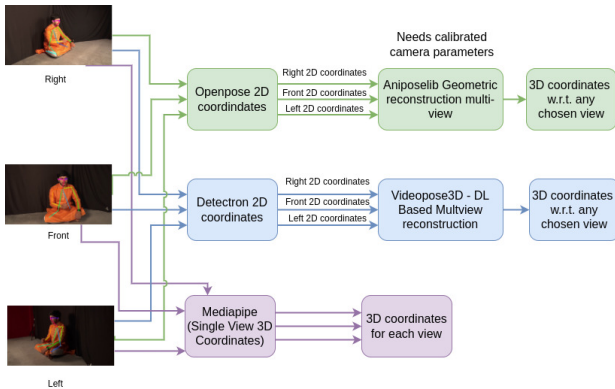


Figure 3: The three HPE methods (see Table 1) in this paper. *Left* and *Right* are views defined with respect to the singer.

6. EXPERIMENT DETAILS

6.1 Kinematic Features

The keypoint detection methods algorithms give us the x and y pixel coordinates for the keypoints and the 3D estimation gives us the z-coordinate to some scale. We linearly interpolate keypoints in frames that having confidence levels lower than the thresholds defined in Tab. 2. We use a low-pass Savitzky-Golay filter [30] to remove any jitter. We next interpolate the gesture data from the video frame rate to 10 ms sampling interval. We use z-score normalization for each keypoint by considering the mean and standard deviation for that keypoint and that coordinate across all frames of the recording. For repeatability, following [5], we estimate velocity and acceleration on each coordinate axis by a 101-point biphasic filter to get a smooth velocity and acceleration profile. We re-use the parameters of the biphasic filter defined in the supplementary link provided in [5].

6.2 Stable Note Detection

Although we replicate from [5] the stable note identification algorithm on audio, we use different features for the gesture classification. Instead of using the velocity and acceleration vector magnitudes, we consider the position, velocity and acceleration along each coordinate axis inde-

pendently. We thus have 9 kinematic features per wrist. As in [5], we only consider stable and non-stable segments which are at least of 500 ms duration. Using this for the 8 singers, we have 15312 segments with 40.65% of them as stable notes. We use the mean and standard deviation per segment of each of the kinematic features for both wrists as input to our classifier. Thus, for models trained on 3D data (aniposelib, Videopose3d, Mediapipe) we have $9 \times 2 \times 2 = 36$ features considering both wrists. For models trained on 2D data (aniposelib, Videopose3d, Mediapipe2D) we have $6 \times 2 \times 2 = 24$ features. With these features, we train a SVM classifier per singer using 10-fold cross-validation and report the mean cross-validation accuracy. (Note that Mediapipe outputs 3D coordinates but we drop the z-coordinate in the experiments for comparison of 2D results.)

6.3 Gesture-based Singer Identification

We use randomly chosen 12s splits from the video in an attempt to identify the singer. We use a time series for the position, velocity and acceleration (PVA) features along each coordinate axis at (each) 10ms time interval for both wrists and elbows. Thus we have 36 features considering wrist and elbow for models using 3D data and 24 features for models using 2D data. We keep aside data for 3 *ragas* as test data (4103 samples), and train on the rest of the data using a random 80–20 train–validation split. We use a deep neural network consisting of convolutional layers followed by a 2D inception block as shown in Fig. 3 of [3]. We find best hyperparameters separately for each HPE technique, retrain the model with best hyperparameters and then report the results on the test set.

6.4 Fusion models

In the fusion models (for both 2D and 3D classifiers) for the stable note classifier we use the predicted probability for the stable note classifier. Thus we have 3 features in the fusion classifier and we train per singer using 10-fold cross-validation a classification model by a hyperparameter choice over logistic regression, random forests and support vector machines. We report the average of the mean per-singer cross-validation F1-score.

For the fusion models (both 2D and 3D) for the gesture-based identification of 8 singers, we take the softmax output of the final layer of the classifier from all views. Thus we have 24 features in the fusion classifier and we train a classifier by 10 fold hyperparameter tuning across logistic regression, random forests and support vector machines. Our training data for the fusion model consists of the softmax predictions on the train and val data of the neural network. We report the accuracy using features generated from the excerpts corresponding to the 3 held-out ragas.

6.5 Results

The results of the stable note detection appears at the first row of Tab. 3. We report the best result across camera views (or reference view for 3D reconstruction) for the HPE method. We note that the performance for the 2D models, MP2 performs better than either OP2 or DE2. Moving to 3D coordinates, we see a significant improvement in AP3 and VP3.

The results of the (pure) gesture-based singer identification classifier is given in the second row of Tab. 3. We report the best result across camera views (or reference view in the case of 3D). The classification accuracies, compared to a chance accuracy of 12.5%, indicate (given gestures are idiosyncratically singer-specific) that the HPE methods are reliable. We observe to our surprise that the method of classical computer vision (AP3) is the best performing model.

	2D			3D		
	OP2	DE2	MP2	AP3	VP3	MP3
StableNote	77.7	78.6	83.0	78.6*	82.5*	83.5
SingerID	83.2	81.9	79.6	83.3	81.4	82.9*

Table 3: F1-score (%) for stable note detection and accuracy (%) for gesture-based singer identification. star(*) indicates significant (p<0.05) difference between 2D and 3D, and bold indicates best result for a task in corresponding methods of 2D/3D.

The first row of Tab. 4 shows the results of decision fusion models based on the corresponding models across views for the stable note task. The results show that 2D fusion gives us comparable performance to reconstruction. The results of the models using fusion of classifiers across views are present in the second row of Tab. 4 for the gesture-based singer identification task. We see that when we use fusion instead of reconstruction, the results are much better with every possible technique for both MIR tasks. Accordingly we recommend this method. All fusion results are statistically significantly better than the corresponding best single view results in Tab. 3.

6.5.1 Ablation Study of Thresholds

Tab. 5 has the OpenPose results using a constant threshold of 0.3 for all views and the Aniposelib result tasks. Tab. 6 has the results ablation study for various levels of interpolated frames.

The results show that our chosen threshold has comparable performance with default 0.3 threshold but our

	2D-Fusion			3D-Fusion		
	OP2	DE2	MP2	AP3	VP3	MP3
StableNote	82.0 [†]	82.1	83.9	82.0	85.0*	86.6*
Singer-ID	91.4 [†]	93.0[†]	92.3 [†]	93.3*	93.6	92.7

Table 4: Fusion based results. Values in %. Bold and star have same meaning as Tab. 3. Values with dagger (†) indicate the 2D-fusion model is better (p< 0.05) than the corresponding 3D model in Tab. 3

method is extensible to other HPE models. Results for 5%,10% and 20% interpolated frames are very similar. However if we set thresholds corresponding to 30% interpolated frames the performance is poorer.

	OP2-Front	OP2-Left	OP2-Right	AP3
Stable Note	77.1	77.8	77.5	78.2
Singer ID	80.8	81.5	82.6	82.3

Table 5: Performance (in %) of OP2 for all views and AP3 using the confidence threshold of 0.3 used in the literature.

Interpolated %	2D Models			3D Models		
	OP2	DE2	MP2	AP3	VP3	MP3
5	78.2	78.6	83.0	78.0	82.6	83.0
10	77.7	78.6	83.0	78.1	82.5	83.0
20	77.1	78.5	82.9	77.4	82.2	82.9
30	75.1	74.8	80.5	75.4	79.0	80.6

Table 6: F1-score (%) across HPE techniques.

7. SUMMARY AND CONCLUSION

Given the importance of reliable joint pose estimation in gesture analysis, we investigated a set of distinct available approaches to the keypoint detection of wrists and elbows for an application of expressive hand movements in two MIR tasks. We showed that the different ways of using multiple camera views, in terms of the single-view pose estimation method and the manner of combining multiple views, can influence task performance significantly. While 3D reconstruction affords a complete description of the gesture movements, the fusion of multiple 2D information is competitive. The fusion of multiple 3D representations is seen to bring in further benefits. The superiority of fusion results over single view is established via statistical significance. The two MIR tasks involve the use of distinctly different machine learning methods (classical SVM, and recent deep-learning) and involve scenes where the action is only in the upper body, providing evidence for the use modern HPE methods. We expect the outcomes of this study therefore to be useful in any application of expressive movement analysis involving upper-body limbs.

Future work would involve the fine-tuning of HPE algorithms with a set of manually labelled keypoints to see whether the optimization with respect to upper body keypoints helps improve the estimates.

8. REFERENCES

- [1] S. Paschalidou, "Effort inference and prediction by acoustic and movement descriptors in interactions with imaginary objects during dhrupad vocal improvisation," *Wearable Technologies*, vol. 3, p. e14, 2022.
- [2] L. Pearson and W. Pouw, "Gesture–vocal coupling in karnatak music performance: A neuro–bodily distributed aesthetic entanglement," *Annals of the New York Academy of Sciences*, vol. 1515, no. 1, pp. 219–236, 2022.
- [3] M. Clayton, P. Rao, N. Shikarpur, S. Roychowdhury, and J. Li, "Raga classification from vocal performances using multimodal analysis," in *Proceedings of the 23rd International Society for Music Information Retrieval Conference, ISMIR, Bengaluru, India, pp. 283-290.*, 2022.
- [4] M. Clayton, J. Li, A. Clarke, and M. Weinzierl, "Hindustani raga and singer classification using 2d and 3d pose estimation from video recordings," *Journal of New Music Research*, pp. 1–16, 2024.
- [5] S. Nadkarni, S. Roychowdhury, P. Rao, and M. Clayton, "Exploring the correspondence of melodic contour with gesture in raga alap singing," in *Proceedings of the 24th International Society for Music Information Retrieval Conference, ISMIR, Milan, Italy, 2023*.
- [6] M. J. Rahaim, *Gesture, melody, and the paramparic body in Hindustani vocal music*. University of California, Berkeley, 2009.
- [7] G. Developers, "Mediapipe pose landmarker," 2020, accessed: 2024-03-09. [Online]. Available: <https://developers.google.com/mediapipe/solutions/pose>
- [8] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Real-time multi-person 2d pose estimation using part affinity fields," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7291–7299.
- [9] S. Dubey and M. Dixit, "A comprehensive survey on human pose estimation approaches," *Multimedia Systems*, vol. 29, no. 1, pp. 167–195, 2023.
- [10] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, "2d human pose estimation: New benchmark and state of the art analysis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3686–3693.
- [11] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer, 2014, pp. 740–755.
- [12] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.
- [13] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2," <https://github.com/facebookresearch/detectron2>, 2019.
- [14] P. Karashchuk, K. L. Rupp, E. S. Dickinson, S. Walling-Bell, E. Sanders, E. Azim, B. W. Brunton, and J. C. Tuthill, "Anipose: A toolkit for robust markerless 3d pose estimation," *Cell reports*, vol. 36, no. 13, 2021.
- [15] D. Pavlo, C. Feichtenhofer, D. Grangier, and M. Auli, "3d human pose estimation in video with temporal convolutions and semi-supervised training," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 7753–7762.
- [16] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, "Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 7, pp. 1325–1339, 2013.
- [17] L. Sigal, A. O. Balan, and M. J. Black, "Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion," *International journal of computer vision*, vol. 87, no. 1, pp. 4–27, 2010.
- [18] V. Bazarevsky, I. Grishchenko, K. Raveendran, T. Zhu, F. Zhang, and M. Grundmann, "Blazepose: On-device real-time body pose tracking," in *CVPR Workshop on Computer Vision for Augmented and Virtual Reality, Seattle, WA, USA, 2020*.
- [19] N. Nakano, T. Sakura, K. Ueda, L. Omura, A. Kimura, Y. Iino, S. Fukashiro, and S. Yoshioka, "Evaluation of 3d markerless motion capture accuracy using openpose with multiple video cameras," *Frontiers in sports and active living*, vol. 2, p. 50, 2020.
- [20] F. Zhang, P. Juneau, C. McGuirk, A. Tu, K. Cheung, N. Baddour, and E. Lemaire, "Comparison of openpose and hyperpose artificial intelligence models for analysis of hand-held smartphone videos," in *2021 IEEE International Symposium on Medical Measurements and Applications (MeMeA)*. IEEE, 2021, pp. 1–6.
- [21] S. Mroz, N. Baddour, C. McGuirk, P. Juneau, A. Tu, K. Cheung, and E. Lemaire, "Comparing the quality of human pose estimation with blazepose or openpose," in *2021 4th International Conference on Bio-Engineering for Smart Technologies (BioSMART)*. IEEE, 2021, pp. 1–4.
- [22] R. Ferens and Y. Keller, "Hyperpose: Camera pose localization using attention hypernetworks," *arXiv preprint arXiv:2303.02610*, 2023.

- [23] M. Mundt, Z. Born, M. Goldacre, and J. Alderson, "Estimating ground reaction forces from two-dimensional pose data: a biomechanics-based comparison of alpha-pose, blazepose, and openpose," *Sensors*, vol. 23, no. 1, p. 78, 2022.
- [24] S. Mehdizadeh, H. Nabavi, A. Sabo, T. Arora, A. Iaboni, and B. Taati, "Concurrent validity of human pose tracking in video for measuring gait parameters in older adults: a preliminary analysis with multiple trackers, viewing angles, and walking directions," *Journal of neuroengineering and rehabilitation*, vol. 18, pp. 1–16, 2021.
- [25] H.-S. Fang, J. Li, H. Tang, C. Xu, H. Zhu, Y. Xiu, Y.-L. Li, and C. Lu, "Alphapose: Whole-body regional multi-person pose estimation and tracking in real-time," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.
- [26] N. Blanchard, K. Skinner, A. Kemp, W. Scheirer, and P. Flynn, "' keep me in, coach!": A computer vision perspective on assessing acl injury risk in female athletes," in *2019 IEEE Winter conference on applications of computer vision (WACV)*. IEEE, 2019, pp. 1366–1374.
- [27] C. Roygaga, D. Patil, M. Boyle, W. Pickard, R. Reiser, A. Bharati, and N. Blanchard, "Ape-v: Athlete performance evaluation using video," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 691–700.
- [28] J. Ripperda, L. Drijvers, and J. Holler, "Speeding up the detection of non-iconic and iconic gestures (spudnig): A toolkit for the automatic detection of hand movements and gestures in video data," *Behavior research methods*, vol. 52, no. 4, pp. 1783–1794, 2020.
- [29] D. Pagnon, M. Domalain, and L. Reveret, "Pose2sim: An end-to-end workflow for 3d markerless sports kinematics—part 1: Robustness," *Sensors*, 2021. [Online]. Available: <https://www.mdpi.com/1424-8220/21/19/6530>
- [30] W. H. Press and S. A. Teukolsky, "Savitzky-golay smoothing filters," *Computers in Physics*, vol. 4, no. 6, pp. 669–672, 1990.

ENHANCING PREDICTIVE MODELS OF MUSIC FAMILIARITY WITH EEG: INSIGHTS FROM FANS AND NON-FANS OF K-POP GROUP NCT127

Seokbeom Park¹ Hyunjae Kim¹ Kyung Myun Lee^{1,2}

¹ Graduate School of Culture Technology, KAIST, South Korea

² School of Digital Humanities and Computational Social Sciences, KAIST, South Korea

{fpfmxh, present, kmlee2}@kaist.ac.kr

ABSTRACT

Predicting a listener’s experience of music based solely on audio features has its limitations due to the individual variability in responses to the same music. This study examines the effectiveness of electroencephalogram (EEG) in predicting the subjective experiences while listening to music, including arousal, valence, familiarity, and preference. We collected EEG data alongside subjective ratings of arousal, valence, familiarity, and preference from both fans (N=20) and non-fans (N=34) of the K-pop idol group, NCT127 to investigate response variability to the same NCT127 music. Our analysis focused on determining whether the inclusion of EEG alongside audio features could enhance the predictive power of linear mixed-effect models for these subjective ratings. Specifically, we employed stimulus-response correlation (SRC), a recent approach in neuroscience correlating stimulus features with EEG responses to the ecologically valid stimuli. The results showed that familiarity and preference was significantly higher in the fan group. Furthermore, the inclusion of SRC significantly enhanced the prediction of familiarity compared to models based solely on audio features. However, the impact of SRC on predictions of arousal and valence exhibited variation depending on the correlated audio features, with certain SRCs improving predictions while others diminished them. For preference, only a few SRCs negatively affected model performance. These results suggest that correlations of EEG responses and audio features can provide information of individual listeners’ subjective responses, particularly in predicting familiarity.

1. INTRODUCTION

The neuroscience of music, employing neuroimaging methods, has revealed how the brain processes music through regions responsible for auditory, motor, and emotional functions, with recent approaches focusing on the brain’s predictive processes [1, 2, 3, 4]. The convergence

of music information retrieval (MIR) and neuroscience has gained significant traction in recent years [5, 6, 7]. For example, Rajagopalan and Kaneshiro have highlighted the potential of electroencephalogram (EEG) in the analysis of musical structure [8]. Furthermore, Ofner and Stober demonstrated the reconstruction of perceived and imagined music from EEG data [9]. These findings highlight the synergistic benefits of integrating MIR and neuroscience. In this paper, we aim to investigate how EEG can enhance the predictive model of subjective listening responses to music, given the individual variability in such experiences.

1.1 Predicting Subjective Music Listening Experience using Audio Features

Subjective music listening experience refers to the individual and unique responses that people have when they listen to music. It encompasses a wide range of aspects, including emotional reactions, preferences, familiarity, and overall enjoyment of the music. Subjective experience acknowledges that each listener’s response to music is personal and may be influenced by various factors such as their musical background and cultural upbringing [10, 11, 12, 13].

Predicting listeners’ subjective experiences of music through audio features has been a significant focus within MIR research. For example, Music Emotion Recognition (MER) aims to predict listeners’ emotional responses using various techniques [14, 15, 16, 17]. Audio features, including tempo, rhythm, melody, and harmony, have been shown to correlate with listeners’ emotional responses and preferences [18, 17]. However, the relationship between audio features and subjective experiences is complex, influenced by individual differences in musical background, culture, and personal taste [19, 20, 21]. Notably, emotional responses can significantly vary depending on individual differences [21, 22, 23, 24]. Thus, relying solely on audio features may not capture the full spectrum of music’s impact on the listener, emphasizing the need for incorporating physiological measures such as EEG in understanding subjective music experiences [18].

1.2 Stimulus-response Correlation

The use of EEG offers a breakthrough in predicting subjective music listening experiences [25]. EEG provides real-time measures of brain activity, allowing direct observa-



© S. Park, H. Kim, and K.M. Lee. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** S. Park, H. Kim, and K.M. Lee, “Enhancing predictive models of music familiarity with EEG: Insights from fans and non-fans of K-pop group NCT127”, in *Proc. of the 25th Int. Society for Music Information Retrieval Conf.*, San Francisco, United States, 2024.

tion of neural responses to musical stimuli. Specifically, stimulus-response correlation (SRC), a recent method correlating stimulus features with EEG responses, enhances the ecological validity of studies by using real-world music stimuli and offers interpretable insights into the direct effects of stimulus features on the listener’s experience [26, 27]. For example, SRC analysis revealed that neural responses are strongly correlated with specific task-relevant visual areas [28]. Additionally, SRC enabled the prediction of speech intelligibility [29]. Despite employing a different method to calculate the correlation between audio features and EEG responses, Weineck et al. found that neural response intensity increased with music familiarity [30]. Therefore, SRC is considered to be useful tool for predicting subjective music listening experiences.

1.3 Research Question

In this study, we aim to investigate the variability of subjective music listening experiences by comparing responses of fans and non-fans to K-pop idol music. Subsequently, we explore the effectiveness of SRC in predicting this individual response variability. To achieve this goal, we formulated the following research questions:

RQ 1: How do subjective music listening experiences, such as arousal, valence, familiarity, and preference, vary individually for the same music among fans and non-fans of K-pop idol music?

RQ 2: How does the inclusion of SRC alongside audio features affect the predictive power of models for arousal, valence, familiarity, and preference in music listening?

RQ 3: Does the effectiveness of SRC in predicting subjective experiences vary depending on the type of audio feature it is correlated with?

To address these questions, we conducted an experiment collecting EEG data and subjective ratings from both fans and non-fans of NCT127 as they listened to music by the group. Utilizing linear mixed-effects models, we analyzed the contribution of audio features and SRC in predicting subjective experiences, providing a comprehensive understanding of how these components interact to shape individual music listening experiences.

2. MATERIALS AND METHODS

2.1 Participants

We recruited 20 fans of NCT127 (mean age 24.8 years, 2 males) and 34 non-fans (mean age 26.1 years, 7 males). To participate in the experiment as part of the fan group, participants were required to meet at least one of the following conditions: they must have attended at least one event featuring NCT127, such as a concert or fan meeting, or they must own at least one piece of NCT127-related merchandise, such as an album, light stick, photocard, LP, or sheet music. This was verified through a photo submission process when applying for the experiment. All participants were Korean non-musicians. All participants had normal hearing and provided written informed consent before starting the experiment.

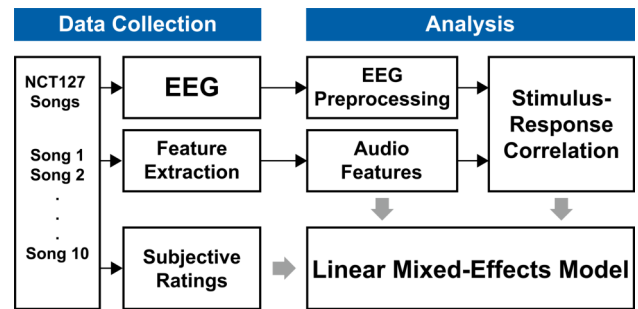


Figure 1: Schematic view of data collection and analysis.

2.2 Stimuli

The music used in the experiment consisted of the NCT127’s top 10 songs based on the YouTube Music rankings as of December 26, 2023. The music was edited from the beginning to the end of the first chorus. The length of the edited audio varied between 60 to 92 seconds. Each audio was edited to began with a 0.5 second fade-in and ended with a 0.5 second fade-out. Then, volume normalization was applied to each channel before being exported. As a result, ten stereo audio files with a 44100Hz sampling rate and 16-bit depth were created for the stimuli.

2.3 Experiment

The EEG experiment was conducted using the Compu-medics Neuroscan system. For EEG recordings, a Synamp RT 64-channel amplifier and a 64-channel Quik-Cap with sintered Ag/AgCl electrodes were used. The data collection was carried out through the Curry 8 acquisition software. EEG electrodes were placed in accordance with the international 10-20 system, and EEG data were collected at a sampling rate of 1000Hz across 64 channels.

The experiment was conducted using STIM2 software in a soundproof room to eliminate noise interference. Participants listened to each stimulus through insert earphone while focusing on a cross in the center of the monitor. Each stimulus was played once, and after listening to each, participants rated their arousal, valence, familiarity, and preference using a 7-point scale. Participants were able to proceed to the next stimulus after completing their ratings. There was a 5-second silence window before and after each stimulus, and the stimuli were played in a randomized order. An overall view of the data collection and analysis is presented in Figure 1.

2.4 Analysis

2.4.1 EEG Preprocessing

The preprocessing of EEG data was conducted using MATLAB with the EEGLAB toolbox [31]. From the 64 channels, the reference channels M1 and M2 were excluded. The EEG data underwent a 1-55 Hz bandpass FIR filter, followed by epoching for each stimulus. Subsequent steps included baseline removal and downsampling from 1000Hz to 125Hz. The data were re-referenced using

the common average reference method, and all EEG data were merged by each participant. Independent Component Analysis (ICA) decomposition (using *runamica15* function) was performed to remove artifacts [32]. Artifactual components (eye, muscle, heart) were chosen by automated artifact IC classifier 'ICLabel' and additional artifactual components were manually chosen [33]. Finally, the EEG data were epoched by each stimulus.

2.4.2 Stimulus-response Correlation

To calculate SRC, we applied a hybrid encoding-decoding technique, performing canonical correlation analysis to maximize the correlation between temporally filtered stimuli (audio) and spatially filtered neural responses (EEG). A detailed explanation of the method, including the computation of spatial and temporal response functions for each component, can be found in Dmochowski et al.'s paper [26].

For SRC calculations, stimulus features were extracted from each audio stimulus using MATLAB *mirtoolbox* [34]. From audio features that Lange and Frieler explored [18], only audio features permitting extraction in a time-by-feature value manner, thus enabling SRC calculation, were selected for investigation. This process resulted in extracting ten audio features: sound envelope, root mean square (RMS), spectral flux, zero-crossing rate, roughness, spectral entropy, spectral centroid, spectral spread, spectral rolloff, and spectral flatness. Each feature was extracted using *mirtoolbox* functions—*mirenvelope*, *mirrms*, *mirflux*, *mirzerocross*, *mirroughness*, *mirentropy*, *mircentroid*, *mirspread*, *mirrolloff*, *mirflatness*—and adjusted to a sample rate of 125Hz. If the sample number of audio features slightly differed from the EEG data, they were adjusted to match the length of the EEG data: longer samples were cut, and shorter ones were zero-padded. Finally, all audio features were z-scored for normalization.

The SRC calculation was performed using a modified version of a publicly available MATLAB implementation by Dmochowski¹. SRCs were computed on a per-stimulus basis for each participant. The regularization parameters were set to 7 for both stimuli and EEG data. The representative SRC value for each stimulus and participant was determined by summing the three components with the highest values. As a result, a total of 54 x 10 x 10 (participants x songs x audio features) SRC values were computed.

2.4.3 Modeling Subjective Experience

Our analysis used linear mixed-effects models to examine the effects of individual audio features, both in isolation and in conjunction with their corresponding SRC, on subjective music listening experiences: arousal, valence, familiarity, and preference. Separate models were constructed for each dependent variable, with each model incorporating a single audio feature as a fixed effect (AF model). In the case of AFSRC models, compared to AF model, SRC was added as a fixed effect. This approach

allowed for a detailed examination of the influence of specific audio features and their neural correlates on listeners' subjective experiences.

The general form of the linear mixed-effects model used in this study is given by:

$$y = X\beta + Z\gamma + \epsilon \quad (1)$$

where y is the vector of observed dependent variables (e.g., arousal, valence), X is the matrix of fixed effects, β represents the coefficients for the fixed effects, Z is the matrix for random effects, γ represents the coefficients for the random effects, and ϵ is the error term.

We fitted two types of models for each dependent variable:

For the audio feature only models, the general form of the model can be represented as:

$$Y_{ij} = \beta_0 + \beta_1 X_{ij} + u_j + \epsilon_{ij} \quad (2)$$

where Y_{ij} is the dependent variable (arousal, valence, familiarity, or preference) for the i -th song listened to by the j -th participant, β_0 is the intercept, β_1 is the fixed effect coefficient of the audio feature X_{ij} , u_j is the random effect for the j -th participant, and ϵ_{ij} is the error term.

For the models with audio feature and SRC as the fixed effects, the equation expands to include the SRC:

$$Y_{ij} = \beta_0 + \beta_1 X_{ij} + \beta_2 S_{ij} + u_j + \epsilon_{ij} \quad (3)$$

where β_2 is the fixed effect coefficient of the SRC S_{ij} related to the audio feature X_{ij} .

The fitting of models was carried out using the *lme4* and *lmerTest* packages in R software. All AF and AFSRC models were cross-validated using leave-one-subject-out cross-validation. To evaluate the significance of each model, we compared it against a null model predicting the same dependent variable using *anova* function. Specifically, we compared AF models with AFSRC models, again using the *anova* function. When comparing models, it is generally accepted that a difference of 2 or more in AIC values indicates a meaningful difference in model performance [35]. In our experiment results, we also categorized a difference in AIC value of 1.9 as a marginal but meaningful difference. This approach allowed us to quantitatively determine the added value of incorporating EEG-derived SRCs into the predictive models of subjective music listening experience.

3. RESULTS

3.1 Subjective Experience of Fans and non-Fans

Independent samples t-test were conducted to examine the group differences between NCT127 fan group and non-fan group while listening to 10 NCT songs in terms of arousal, valence, familiarity, and preference (Figure 2).

For arousal, there was no significant difference between the fan group ($M = 4.54$, $SD = 1.38$) and the non-fan group ($M = 4.98$, $SD = 0.75$); $t(25.812) = -1.320$, $p = .199$.

¹ <https://github.com/dmochow/SRC>

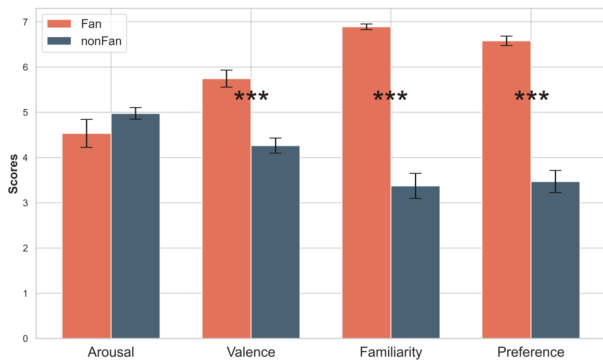


Figure 2: Average subjective ratings by fan and non-fan group. Error bar indicates standard error. *** = $p < .001$

Valence scores were significantly higher for the fan group ($M = 5.75$, $SD = 0.84$) compared to the non-fan group ($M = 4.26$, $SD = 0.97$); $t(52) = 5.665$, $p < .001$, indicating that fans experienced more positive emotions towards NCT127 songs.

A significant difference in familiarity with the songs was observed, with the fan group reporting much higher familiarity ($M = 6.89$, $SD = 0.28$) than the non-fan group ($M = 3.37$, $SD = 1.61$); $t(36.199) = 12.425$, $p < .001$.

Preference ratings were also significantly higher in the fan group ($M = 6.58$, $SD = 0.47$) compared to the non-fan group ($M = 3.47$, $SD = 1.42$); $t(43.705) = 11.734$, $p < .001$. This result suggests a strong preference for NCT127 music among fans.

Overall, the results indicate that while fans and non-fans do not differ significantly in arousal when listening to NCT127 songs, fans report significantly more positive valence, greater familiarity, and a stronger preference for the NCT127 songs compared to non-fans.

3.2 Predicting Subjective Music Listening Experience with Stimulus-response Correlation

Integrating SRC into the audio feature only models yielded variable results depending on the subjective ratings and audio features. Most importantly, for familiarity, SRC significantly enhanced predictive power of the models across various audio features (Figure 3C). Among ten audio features, SRC correlated with eight audio features showed significant improvement in predicting familiarity.

The prediction of arousal was enhanced from SRC calculated with specific audio feature—spectral flux, spectral centroid, spectral rolloff, and spectral flatness—while roughness was found to negatively impact model performance (Figure 3A). For valence, SRC correlated with spectral flux improved the model performance, whereas sound envelope, RMS, and zero-crossing rate increased the AIC values by 1.9 or more, suggesting reduction in model performance (Figure 3B). In models predicting preference, the addition of SRC related to sound envelope, roughness, spectral centroid, and spectral rolloff resulted in an increase of 1.9 or more in the AIC values, indicating a de-

cline in performance (Figure 3D).

In our analysis of the significance of AF models by comparison to null models, we observed distinct patterns across subjective music listening experiences (Table 1). Specifically, roughness and spectral flatness were key predictors for arousal, while sound envelope, RMS, spectral flux, and zero-crossing rate significantly predicted valence. Familiarity was well predicted by RMS, spectral rolloff, and spectral flatness, and preference was effectively predicted by sound envelope, RMS, spectral flux, zero-crossing rate, spectral entropy, and spectral flatness. Notably, the inclusion of SRC based on RMS, spectral flux, zero-crossing rate, spectral entropy, and spectral flatness did not significantly enhance the performance of models predicting preference (Figure 3D), yet these models demonstrated a good fit using only audio features. For detailed comparisons and summaries of all model fits and cross-validation results, refer to the supplementary materials².

4. DISCUSSION

We compared subjective music listening experiences, specifically focusing on arousal, valence, familiarity, and preference when fans and non-fans of NCT127 listened to the same NCT127 songs. The results showed that valence, familiarity, and preference were significantly higher in the fan group, while there was no significant difference in arousal. Then, we investigated the combined effects of audio features and SRC derived from EEG data on predicting subjective music listening experiences. Through comparing linear mixed-effects models based solely on audio features with those incorporating both audio features and SRC, we revealed that integrating SRC with audio features significantly enhances the predictive power for familiarity. However, the influence of SRC on predictions of arousal and valence showed variation depending on the correlated audio features. The inclusion of few SRC decreased the predictive power of preference.

The notably higher familiarity and preference ratings observed in the NCT127 fan group were anticipated outcomes, aligning with the criteria we set for participant recruitment: participants in the fan group were required to regularly listen to NCT127’s music, confirm their attendance at an NCT127 event, and own NCT127-related merchandise.

The absence of significant difference in arousal between groups suggests that arousal ratings were predominantly influenced by the acoustic characteristics of the music, such as tempo and timbre, rather than personal traits [36]. This finding aligns with previous research indicating minimal variability among individuals in arousal ratings for the same musical piece. [37, 38].

Incorporating SRC alongside audio features enhances the predictive accuracy for familiarity. SRC, derived from a hybrid encoding-decoding technique, captures distributed representations in neural response [26]. Since

² <https://blues95.github.io/ISMIR2024/>

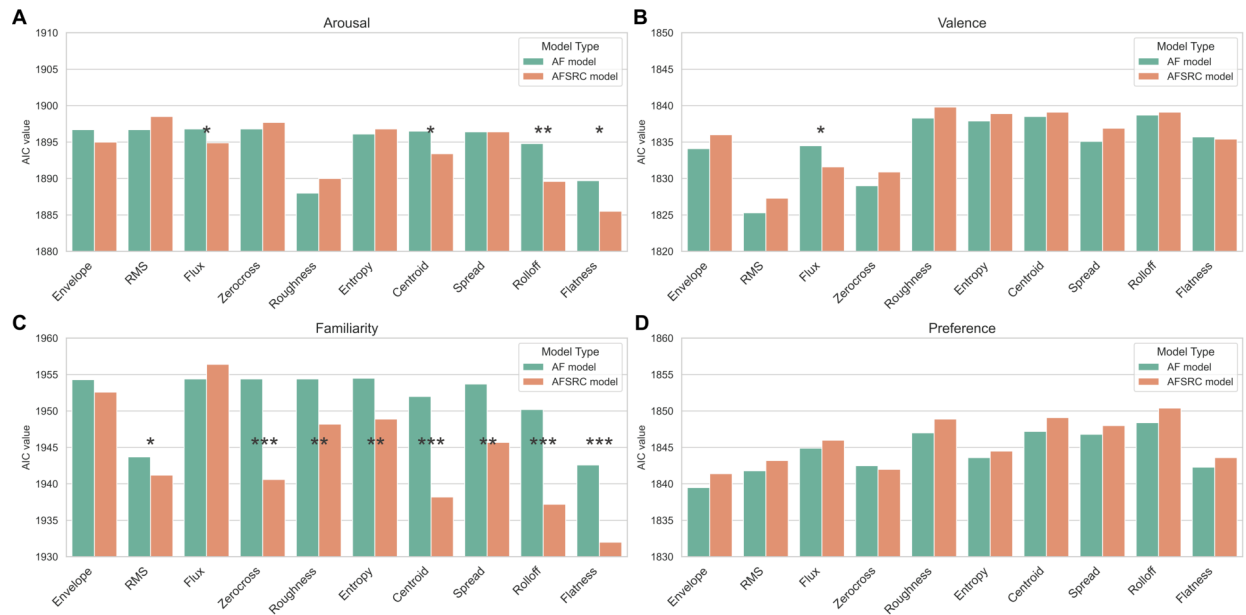


Figure 3: AIC values for each model. (A) Arousal (B) Valence (C) Familiarity (D) Preference. Asterisk symbols indicate the significant improvement of the AFSRC model compared to AF model. Note that the scale of y-axis are different. * = $p < .05$, ** = $p < .01$, *** = $p < .001$.

Table 1: Significance of AF models compared to null models. * = $p < .05$, ** = $p < .01$, *** = $p < .001$.

Audio Feature	Arousal		Valence		Familiarity		Preference	
	AIC	$Pr(> \chi^2)$	AIC	$Pr(> \chi^2)$	AIC	$Pr(> \chi^2)$	AIC	$Pr(> \chi^2)$
Envelope	1896.7	0.682	1834.1	0.029*	1954.3	0.628	1839.5	0.002**
RMS	1896.7	0.732	1825.3	<0.001***	1943.7	0.001**	1841.8	0.007**
Flux	1896.8	0.841	1834.5	0.037*	1954.4	0.698	1844.9	0.041*
Zerocross	1896.8	0.925	1829.0	0.002**	1954.4	0.686	1842.5	0.010*
Roughness	1888.0	0.003**	1838.3	0.448	1954.4	0.734	1847.0	0.150
Entropy	1896.1	0.394	1837.9	0.327	1954.5	0.795	1843.6	0.019*
Centroid	1896.5	0.597	1838.5	0.535	1952.0	0.108	1847.2	0.170
Spread	1896.4	0.531	1835.1	0.052	1953.7	0.357	1846.8	0.127
Rolloff	1894.8	0.150	1838.7	0.751	1950.2	0.036*	1848.4	0.417
Flatness	1889.7	0.008**	1835.7	0.076	1942.6	<0.001***	1842.3	0.009**

SRCs in this study were computed by correlating particular audio features with EEG responses, it is possible that audio features of highly familiar music were more effectively represented in neural responses. Familiar music is known to enhance brain activity related to recurring musical patterns and structures [39]. Familiarity may foster better recall of the song, leading to enhanced representation in the brain [40]. Thus, exposure to or familiarity with stimuli may facilitate the processing of specific stimulus features.

In a previous study examining the relationship between audio features and neural responses, Weineck et al. used temporal response function and reliable component analysis to calculate neural synchronization, employing methods distinct from our study [30]. They investigated how synchronization varied with music familiarity, enjoyment, and beat easiness. Their findings indicated that the in-

tensity of neural responses increased with familiar music. While a direct comparison with our study is challenging due to the methodological differences, both studies demonstrate that music familiarity is reflected in the relationship between stimulus (audio features) and response (EEG).

The impact of SRC on the predictions of arousal and valence varied depending on the correlated audio features. In the case of preference, the inclusion of few SRC decreased the model performance, suggesting that emotions or preferences evoked by music may be relatively less dependent on how the audio features are represented in the brain compared to familiarity. Contrary to our findings regarding preference, Pandey et al. demonstrated that stronger SRCs predict increased levels of enjoyment of music [41]. This difference may be due to the selection of features for SRC calculation. Our study used various audio features separately, whereas they used the principal component of 18

audio features for SRC calculation.

The fitting of AF models demonstrated that specific audio features alone can predict subjective music listening experiences. This aligns with the effectiveness of using audio features for training deep learning models in prior MER research.

There are few limitations of this work. First, the demographic composition of our participants, particularly regarding gender distribution, may limit the generalizability of our findings. The process of recruiting fans of a specific artist resulted in a gender imbalance among our participants. Future research should aim to recruit a more balanced participants to enhance the reliability of the results. Second, our analysis only used linear mixed-effects models, making it challenging to generalize the significance of specific audio features in relation to subjective music listening experiences. Since the impact of audio features and SRC on subjective experience may have an inherent nonlinear characteristics, future studies should validate the efficacy of SRC as a learning feature or predictor using a broader range of models, including deep learning-based models capable of capturing nonlinearity. Finally, we only considered ten low-level signal components as audio features in our study. However, the correlations of higher-level audio features, such as chromagrams and various rhythmic features, and EEG might contain unique information about the subjective music listening experience. Therefore, future research should investigate the use of a broader range of audio features, including higher-level audio features.

5. CONCLUSION

This paper explores individual differences in music listening experiences among both fan and non-fan groups of the K-pop idol group NCT127. We aim to demonstrate how responses to the same NCT127 music vary in arousal, valence, familiarity, and preference across different individuals. Furthermore, we investigate the predictive capability of EEG responses, particularly through SRC, regarding subjective music listening experiences. By comparing linear mixed-effects models that solely rely on audio features with those incorporating SRC, our findings underscore the significant role of EEG data in improving the prediction accuracy of music familiarity. This result suggests that using SRC could enable the prediction of individual music listening experiences, which would be challenging using audio features alone.

6. ACKNOWLEDGMENTS

This work was supported by the Ministry of Education of the Republic of Korea and the National Research Foundation of Korea (NRF-2023R1A2C100475512).

7. ETHICS STATEMENT

The ethics of the study were approved by the Institutional Review Board of the Korea Advanced Institute of Science

and Technology.

8. REFERENCES

- [1] R. J. Zatorre, J. L. Chen, and V. B. Penhune, “When the brain plays music: auditory–motor interactions in music perception and production,” *Nature Reviews Neuroscience*, vol. 8, no. 7, pp. 547–558, Jul. 2007.
- [2] S. Koelsch, “Towards a neural basis of music-evoked emotions,” *Trends in Cognitive Sciences*, vol. 14, no. 3, pp. 131–137, 2010. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1364661310000033>
- [3] —, “Brain correlates of music-evoked emotions,” *Nature Reviews Neuroscience*, vol. 15, no. 3, pp. 170–180, Mar. 2014.
- [4] P. Vuust, O. A. Heggli, K. J. Friston, and M. L. Kringelbach, “Music in the brain,” *Nature Reviews Neuroscience*, vol. 23, no. 5, pp. 287–305, May 2022.
- [5] B. Kaneshiro and J. P. Dmochowski, “Neuroimaging Methods for Music Information Retrieval: Current Findings and Future Prospects.” in *Proceedings of the 16th International Society for Music Information Retrieval Conference*. ISMIR, Sep. 2015, pp. 538–544. [Online]. Available: <https://doi.org/10.5281/zenodo.1416082>
- [6] E. B. Abrams, E. M. Vidal, C. Pelofi, and P. Ripollés, “Retrieving musical information from neural data: how cognitive features enrich acoustic ones,” in *Proceedings of the 23rd International Society for Music Information Retrieval Conference*. ISMIR, Nov. 2022, pp. 160–168. [Online]. Available: <https://doi.org/10.5281/zenodo.7343078>
- [7] A. Ofner and S. Stober, “Modeling perception with hierarchical prediction: Auditory segmentation with deep predictive coding locates candidate evoked potentials in EEG,” in *Proceedings of the 21st International Society for Music Information Retrieval Conference*. ISMIR, Nov. 2020, pp. 566–573. [Online]. Available: <https://doi.org/10.5281/zenodo.4245496>
- [8] N. Rajagopalan and B. Kaneshiro, “Correlation of EEG Responses Reflects Structural Similarity of Choruses in Popular Music,” in *Proceedings of the 24th International Society for Music Information Retrieval Conference*. ISMIR, Dec. 2023, pp. 264–271. [Online]. Available: <https://doi.org/10.5281/zenodo.10265273>
- [9] A. Ofner and S. Stober, “Shared Generative Representation of Auditory Concepts and EEG to Reconstruct Perceived and Imagined Music,” in *Proceedings of the 19th International Society for Music Information Retrieval Conference*. ISMIR, Nov. 2018, pp. 392–399. [Online]. Available: <https://doi.org/10.5281/zenodo.1492433>

- [10] E. Przysinda, T. Zeng, K. Maves, C. Arkin, and P. Loui, "Jazz musicians reveal role of expectancy in human creativity," *Brain and Cognition*, vol. 119, pp. 45–53, 2017. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0278262617300994>
- [11] E. H. Margulis, P. C. M. Wong, C. Turnbull, B. M. Kubit, and J. D. McAuley, "Narratives imagined in response to instrumental music reveal culture-bounded intersubjectivity," *Proceedings of the National Academy of Sciences*, vol. 119, no. 4, p. e2110406119, 2022. [Online]. Available: <https://www.pnas.org/doi/abs/10.1073/pnas.2110406119>
- [12] K. B. Doelling and D. Poeppel, "Cortical entrainment to music and its modulation by expertise," *Proceedings of the National Academy of Sciences*, vol. 112, no. 45, pp. E6233–E6242, 2015. [Online]. Available: <https://www.pnas.org/doi/abs/10.1073/pnas.1508431112>
- [13] Y.-H. Yang and X. Hu, "Cross-cultural Music Mood Classification: A Comparison on English and Chinese Songs." in *Proceedings of the 13th International Society for Music Information Retrieval Conference. ISMIR*, Sep. 2012, pp. 19–24. [Online]. Available: <https://doi.org/10.5281/zenodo.1416666>
- [14] L. Zhang, X. Yang, Y. Zhang, and J. Luo, "Dual Attention-Based Multi-Scale Feature Fusion Approach for Dynamic Music Emotion Recognition," in *Proceedings of the 24th International Society for Music Information Retrieval Conference. ISMIR*, Dec. 2023, pp. 207–214. [Online]. Available: <https://doi.org/10.5281/zenodo.10265259>
- [15] S. Chaki, P. Doshi, S. Bhattacharya, and P. P. Patnaik, "Explaining perceived emotion predictions in music: An attentive approach," in *Proceedings of the 21st International Society for Music Information Retrieval Conference. ISMIR*, Nov. 2020, pp. 150–156. [Online]. Available: <https://doi.org/10.5281/zenodo.4245388>
- [16] Y. E. Kim, E. M. Schmidt, R. Migneco, B. G. Morton, P. Richardson, J. J. Scott, J. A. Speck, and D. Turnbull, "State of the Art Report: Music Emotion Recognition: A State of the Art Review." in *Proceedings of the 11th International Society for Music Information Retrieval Conference. ISMIR*, Sep. 2010, pp. 255–266. [Online]. Available: <https://doi.org/10.5281/zenodo.1417945>
- [17] T. Eerola and J. K. Vuoskoski, "A Review of Music and Emotion Studies: Approaches, Emotion Models, and Stimuli," *Music Perception*, vol. 30, no. 3, pp. 307–340, 02 2013. [Online]. Available: <https://doi.org/10.1525/mp.2012.30.3.307>
- [18] E. B. Lange and K. Frieler, "Challenges and Opportunities of Predicting Musical Emotions with Perceptual and Automated Features," *Music Perception*, vol. 36, no. 2, pp. 217–242, 12 2018. [Online]. Available: <https://doi.org/10.1525/mp.2018.36.2.217>
- [19] H. Lee, F. Höger, M. Schönwiesner, M. Park, and N. Jacoby, "Cross-cultural Mood Perception in Pop Songs and its Alignment with Mood Detection Algorithms," in *Proceedings of the 22nd International Society for Music Information Retrieval Conference. ISMIR*, Oct. 2021, pp. 366–373. [Online]. Available: <https://doi.org/10.5281/zenodo.5625680>
- [20] J. H. McDermott, A. F. Schultz, E. A. Undurraga, and R. A. Godoy, "Indifference to dissonance in native amazonians reveals cultural variation in music perception," *Nature*, vol. 535, no. 7613, pp. 547–550, Jul. 2016.
- [21] E. Mas-Herrero, J. Marco-Pallares, U. Lorenzo-Seva, R. J. Zatorre, and A. Rodriguez-Fornells, "Individual Differences in Music Reward Experiences," *Music Perception*, vol. 31, no. 2, pp. 118–138, 12 2013. [Online]. Available: <https://doi.org/10.1525/mp.2013.31.2.118>
- [22] S. Liljeström, P. N. Juslin, and D. Västfjäll, "Experimental evidence of the roles of music choice, social context, and listener personality in emotional reactions to music," *Psychology of Music*, vol. 41, no. 5, pp. 579–599, 2013. [Online]. Available: <https://doi.org/10.1177/0305735612440615>
- [23] M. Park, K. Hennig-Fast, Y. Bao, P. Carl, E. Pöppel, L. Welker, M. Reiser, T. Meindl, and E. Gutyrchik, "Personality traits modulate neural responses to emotions expressed in music," *Brain Research*, vol. 1523, pp. 68–76, 2013. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0006899313007816>
- [24] K. Mori and M. Iwanaga, "General reward sensitivity predicts intensity of music-evoked chills," *Music Percept.*, vol. 32, no. 5, pp. 484–492, Jun. 2015.
- [25] I. Daly, D. Williams, J. Hallowell, F. Hwang, A. Kirke, A. Malik, J. Weaver, E. Miranda, and S. J. Nasuto, "Music-induced emotions can be predicted from a combination of brain activity and acoustic features," *Brain and Cognition*, vol. 101, pp. 1–11, 2015. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0278262615300142>
- [26] J. P. Dmochowski, J. J. Ki, P. DeGuzman, P. Sajda, and L. C. Parra, "Extracting multidimensional stimulus-response correlations using hybrid encoding-decoding of neural activity," *NeuroImage*, vol. 180, pp. 134–146, 2018, new advances in encoding and decoding of brain signals. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1053811917304299>
- [27] B. Kaneshiro, D. T. Nguyen, A. M. Norcia, J. P. Dmochowski, and J. Berger, "Natural music evokes

- correlated eeg responses reflecting temporal structure and beat,” *NeuroImage*, vol. 214, p. 116559, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S105381192030046X>
- [28] J. J. Ki, J. P. Dmochowski, J. Touryan, and L. C. Parra, “Neural responses to natural visual motion are spatially selective across the visual field, with selectivity differing across brain areas and task,” *European Journal of Neuroscience*, vol. 54, no. 10, pp. 7609–7625, 2021. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1111/ejn.15503>
- [29] I. Iotzov and L. C. Parra, “Eeg can predict speech intelligibility,” *Journal of Neural Engineering*, vol. 16, no. 3, p. 036008, mar 2019. [Online]. Available: <https://dx.doi.org/10.1088/1741-2552/ab07fe>
- [30] K. Weineck, O. X. Wen, and M. J. Henry, “Neural synchronization is strongest to the spectral flux of slow music and depends on familiarity and beat salience,” *eLife*, vol. 11, p. e75515, sep 2022. [Online]. Available: <https://doi.org/10.7554/eLife.75515>
- [31] A. Delorme and S. Makeig, “Eeglab: an open source toolbox for analysis of single-trial eeg dynamics including independent component analysis,” *Journal of Neuroscience Methods*, vol. 134, no. 1, pp. 9–21, 2004. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0165027003003479>
- [32] S.-H. Hsu, L. Pion-Tonachini, J. Palmer, M. Miyakoshi, S. Makeig, and T.-P. Jung, “Modeling brain dynamic state changes with adaptive mixture independent component analysis,” *NeuroImage*, vol. 183, pp. 47–61, 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1053811918306888>
- [33] L. Pion-Tonachini, K. Kreutz-Delgado, and S. Makeig, “Iclabel: An automated electroencephalographic independent component classifier, dataset, and website,” *NeuroImage*, vol. 198, pp. 181–197, 2019. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1053811919304185>
- [34] O. Lartillot and P. Toivianen, “MIR in Matlab (II): A Toolbox for Musical Feature Extraction from Audio.” in *Proceedings of the 8th International Conference on Music Information Retrieval. ISMIR*, Sep. 2018, pp. 127–130. [Online]. Available: <https://doi.org/10.5281/zenodo.1417145>
- [35] S. Müller, J. L. Scealy, and A. H. Welsh, “Model selection in linear mixed models,” *Statistical Science*, vol. 28, no. 2, pp. 135–167, 2013. [Online]. Available: <http://www.jstor.org/stable/43288485>
- [36] R. T. Dean, F. Bailes, and E. Schubert, “Acoustic intensity causes perceived changes in arousal levels in music: An experimental investigation,” *PLOS ONE*, vol. 6, no. 4, pp. 1–8, 04 2011. [Online]. Available: <https://doi.org/10.1371/journal.pone.0018591>
- [37] S. Yang, C. N. Reed, E. Chew, and M. Barthet, “Examining emotion perception agreement in live music performance,” *IEEE Transactions on Affective Computing*, vol. 14, no. 2, pp. 1442–1460, 2023.
- [38] B. Gingras, M. M. Marin, E. Puig-Waldmüller, and W. T. Fitch, “The eye is listening: Music-induced arousal and individual differences predict pupillary responses,” *Frontiers in Human Neuroscience*, vol. 9, 2015. [Online]. Available: <https://www.frontiersin.org/articles/10.3389/fnhum.2015.00619>
- [39] C. Freitas, E. Manzato, A. Burini, M. J. Taylor, J. P. Lerch, and E. Anagnostou, “Neural correlates of familiarity in music listening: A systematic review and a neuroimaging meta-analysis,” *Frontiers in Neuroscience*, vol. 12, 2018. [Online]. Available: <https://www.frontiersin.org/journals/neuroscience/articles/10.3389/fnins.2018.00686>
- [40] V. Vuong, P. Hewan, M. Perron, M. H. Thaut, and C. Alain, “The neural bases of familiar music listening in healthy individuals: An activation likelihood estimation meta-analysis,” *Neuroscience Biobehavioral Reviews*, vol. 154, p. 105423, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0149763423003925>
- [41] P. Pandey, P. S. Bedmutha, K. P. Miyapuram, and D. Lomas, “Stronger correlation of music features with brain signals predicts increased levels of enjoyment,” in *2023 IEEE Applied Sensing Conference (APSCON)*, 2023, pp. 1–3.

MOSAIKBOX: IMPROVING FULLY AUTOMATIC DJ MIXING THROUGH RULE-BASED STEM MODIFICATION AND PRECISE BEAT-GRID ESTIMATION

Robert Sowula

TU Wien

robert@sowula.at

Peter Knees

Faculty of Informatics, TU Wien

peter.knees@tuwien.ac.at

ABSTRACT

We present a novel system for automatic music mixing combining diverse music information retrieval (MIR) techniques and sources for song selection and transitioning. Specifically, we explore how music source separation and stem analysis can contribute to the task of music similarity calculation by modifying incompatible stems using a rule-based approach and investigate how audio-based similarity measures can be supplemented by lyrics as contextual information to capture more aspects of music. Additionally, we propose a novel approach for tempo detection, outperforming state-of-the-art techniques in low error-tolerance windows. We evaluate our approaches using a listening experiment and compare them to a state-of-the-art model as a baseline. The results show that our approach to automatic song selection and automated music mixing significantly outperforms the baseline and that our rule-based stem removal approach significantly enhances the perceived quality of a mix. No improvement can be observed for the inclusion of contextual information, i.e., mood information derived from lyrics, into the music similarity measure.

1. INTRODUCTION

DJs have become an essential aspect of many large social events today. The quality of their performance heavily depends on the DJ's experience, knowledge of music, and understanding of what resonates with the audience [1]. Although many attempts [2–8] have been made to automate this role, a DJ is still considered indispensable for providing enjoyable and seamless listening experiences and mixing, i.e., transitioning of tracks.

In this paper, we propose *Mosaikbox*, an automatic music mixing system primarily focused on EDM, incorporating state-of-the-art MIR methods to mimic aspects considered by DJs when selecting and mixing tracks. For selection, these aspects include timbre, which has been shown to improve methods for judging music similarity if combined

with other auditory descriptors [9]. For mixing, a typical transition technique is the fade in/fade out. During its transition period, the two songs are audible for some time. Even if the tempo and key match perfectly and the timbral compatibility is high, a dissimilar drum pattern, e.g., with off-beats at different times than the original track or clashing vocals, can result in a combination that does not sound right. Removing incompatible stems during transitions would solve many traditional mixing challenges.

A rather open question is the use of contextual information for track selection by DJs. Contextual information, such as song lyrics, contains information that audio-based approaches cannot capture and vice versa. Since approaches such as [10] have shown that lyrics can be used to predict the mood of a song, combining audio-based and contextual information might further improve the quality of track selection.

The objectives of this paper are therefore: (1) to introduce a novel automatic music-mixing pipeline, (2) to investigate how a rule-based stem modification procedure can support a music similarity measure in automatic music mixing, and (3) to explore whether we can improve the used musical similarity measure by complementing it with contextual information.

2. RELATED WORK

Besides commercial, closed-source tools, such as VirtualDJ, djay, and NI Traktor, various academic approaches have been proposed for automatic music mixing. Jehan [2] introduced an automated DJ system focused on beat matching on downbeats and transitioning on rhythmically similar segments without incorporating harmonic or timbral information or automatic track selection. Building on this, Lin et al. [3] incorporated pitch information and introduced a method for automatic track selection and ordering. Ishizaki et al. [4] further proposed a method for reducing discomfort when mixing songs with heavily differing tempi in his automatic DJ system. Davies et al. developed AutoMashUpper (AMU) [5], an automatic mashup system that mixes songs using a mashability estimate over phrase-level segments. AMU incorporates a weighted combination of rhythmic and harmonic similarity and spectral balance into its mashability measure. Hiari et al. [6, 7] introduced another automated DJ system based on latent topic modeling of the chroma features and beat similarity for



song selection and cue point estimation. Vande Veire and De Bie [11] built an automatic mixing system similar to AMU with multiple transition methods and a focus on musical style similarity, but less powerful similarity measures compared to AMU to optimize for runtime performance. Huang et al. [8] proposed a pure mashup system that uses isolated stems of different songs to create a mashup. Unlike the previously described automated mixing systems, this approach focuses on mixing a combination of stems, ensuring that each stem type is used only once.

Our work differs from existing methods by proposing a more comprehensive mixability measure to capture additional audio and contextual aspects to better match DJ music selection techniques. Furthermore, we focus on working with completely mastered tracks, integrate state-of-the-art MIR techniques, and perform stem separation to support our mixability measure.

3. METHOD

Our method for automatic mixing comprises the following components to build the *Mosaikbox* system: beat grid estimation and tempo detection, structural segmentation, multi-faceted estimation of music similarity, and mixing of tracks.

3.1 Beat Grid Estimation and Tempo Detection

We build our beat and tempo detection pipeline upon a fixed beat grid approach using a 4/4 time signature, similar to popular DJ software. To build a beat grid, we need two types of information: the song’s tempo and the location of the first downbeat. We derive the beat positions, including the beat types (1st, 2nd, 3rd, and 4th beat) using the state-of-the-art beat tracking system *BeatNet* [12].

Calculating the tempo by averaging inter-beat intervals or using their median can lead to octave errors ($\frac{1}{2}$, $\frac{1}{3}$, 2, 3 multiples of the tempo), where the problematic tempi are the $\frac{1}{3}$ and 3 multiples of the true tempo for non-duple meter music. To address this, we model the beat grid estimation as a 2-dimensional constrained minimization problem, given the detected beat timings t_i , where $i = 1, 2, \dots, n$. Note that some beats might be missing due to detection errors. We want to find the optimal first downbeat position g_1 and the tempo bpm such that the beat positions of the constructed beat grid g_j are evenly spaced and have minimal deviation from the detected beat positions t_i .

To restrict the search space, we estimate the tempo bpm_{est} by using the inter-beat median Δt_{Mdn} . We then perform a global search twice using the dual annealing algorithm, a variant of the simulated annealing algorithm, paired with a local search algorithm for accepted solutions [13]. The objective is to minimize the mean of the absolute differences between each estimated beat grid position g_i and detected beat positions t_j . The initial global search spans a wide range, from 60 bpm to +15% of bpm_{est} and the first downbeat from 0 to +40% of Δt_{Mdn} . To avoid local minima, we conduct a subsequent narrower search within $\pm 5\%$ of bpm_{est} and 0 to +5% of Δt_{Mdn} . Finally, we

fine-tune the beat grid by performing local minimization over only the offset of the first downbeat position from 0 to +40% of Δt_{Mdn} .

3.1.1 Benchmark

We evaluated the performance of our beat grid and tempo estimation algorithm on the GiantSteps dataset [14, 15], as it has not been used for training BeatNet [12] nor current state-of-the-art tempo estimation approaches such as the one by Böck and Davis [16].

Slight deviations in the estimated tempo lead to significant errors in the beat grid estimation. Thus, we deem the metrics *Accuracy 1* and *Accuracy 2* as defined by Gouyon et al. [17] using a 4% tolerance window as too loose, and additionally evaluate the performance of our tempo estimation algorithm for smaller tolerance windows of 1% and 0%. Table 1 compares our tempo estimation algorithm and its inter-beat interval (IBI) pre-estimation with the state-of-the-art tempo estimation algorithm by Böck and Davis [16] on the GiantSteps dataset. While our approach does not outperform the state-of-the-art algorithm for the 4% tolerance window, it demonstrates better performance for the 1% and 0% tolerance windows. The results also show that while IBI is important, it is not the primary contributor to our method’s performance.

	Böck & Davis [16]	IBI	Ours
Accuracy 1 (4%)	87.29	74.13	82.30
Accuracy 1 (1%)	67.02	58.40	69.59
Accuracy 1 (0%)	0.15	3.03	19.97
Accuracy 2 (4%)	96.97	78.08	90.77
Accuracy 2 (1%)	74.38	61.35	76.70
Accuracy 2 (0%)	0.45	3.11	24.51

Table 1. Comparison of our tempo estimation algorithm and its inter-beat interval estimation with a state-of-the-art approach on unseen data from the GiantSteps dataset.

3.2 Structural Segmentation

Music transitions sound most pleasing when performed at musically fitting positions of a song. We therefore combined the boundary detection algorithm by Serrà et al. [18] with the labeling approach by Nieto and Bello [19].

In electronic music, segments typically align with downbeats. Therefore, we quantize the detected segment boundaries to the nearest beat position and shift them by one beat to the nearest downbeat. Boundaries starting or ending on the third beat are not shifted, due to potential causes, such as errors in the downbeat detection, time signature estimation, or different song structures.

Although mixing intros with outros is a straightforward way of transitioning whole songs, we abstain from this practice as we aim for a more energetic mix. Thus, we penalize intro and outro segments by the factor 0.5, which is then multiplied by the similarity measure. The progression of the energy level is a task addressed in the similarity measure. We assume that low-energy and high-energy

segments will not be mixed and thus do not differentiate between other segment types.

3.3 Music Similarity

3.3.1 Rhythmic Similarity

We believe that drums are the primary rhythmic component in EDM music. Instead of relying on onset detection functions, which have poor performance in polyphonic audio, we employ the drum transcription system by Southall et al. [20, 21] to extract drum patterns from the audio. To be able to detect different kinds of rhythm patterns besides the classical "straight" pattern, such as "swing", "shuffle" or "offbeats" which are a primary component in EDM sub-genres such as drum and bass, we follow the AMU approach of Davies et al. [5] and sub-divide the beat grid into 12 equally spaced intervals. We then detect the *kick*, *snare*, and *hi-hat* drum positions, quantize them over the sub-beat grid, and stack them on top of each other to obtain a 3-dimensional binary vector R_n for all songs n of length $K * 12$, where K is the number of beat positions of a song. The rhythmic similarity is then calculated between phrase sections p of the seed song s and a candidate song c for all k beat shifts of c . While AMU uses cosine similarity as a rhythmic similarity measure, we decided to employ a stricter similarity measure to capture dissimilarities in the drum patterns. Thus, we defined the similarity measure as the average of the sub-beat positions where the drum patterns of the seed song section and the candidate song section match.

For each drum vector $R_{s,p,d}$ within phrase section p of the seed song s , where $d \in 1, 2, 3$ denotes the drum vector dimensions corresponding to the *kick*, *snare*, and *hi-hat*, we compute the average number of matching sub-beat positions l over all beat shifts k against all candidate songs c . The overall rhythmic similarity $M_{R,s}(k)$ is then derived by averaging the similarities obtained across the three drum dimensions d as

$$M_{R,c}(k) = \frac{1}{3} \sum_{d=1}^3 \left(\frac{1}{m} \sum_{l=1}^m [R_{s,p,d,l} = R_{c,k,d,l}] \right), \quad (1)$$

where m is the length of the drum vector $R_{s,p}$ of the phrase section in the seed song, and $[R_{s,p,d,l} = R_{c,k,d,l}]$ denotes the Iverson bracket.

3.3.2 Timbral Similarity

To model the timbral component, we will follow the approach of Rocha et al. [22] and Panteli et al. [23], using MFCCs and the auditory descriptors spectral flatness and dirtiness. By stacking the MFCCs, spectral flatness, and dirtiness descriptors on top of each other, we obtain a 28-dimensional vector $T_{n,p}$ for a song n and phrase section p . Due to high computational demands, we calculate the timbral component once per phrase section instead of every beat shift k of the candidate song c , assuming the timbral component remains relatively constant across phrase sections. The timbral similarity is then calculated by computing the cosine similarity between the timbral component

$T_{s,p}$ of phrase section p of seed song s and the timbral component $T_{c,q}$ of all phrase sections q of candidate song c as

$$M_{T,c}(q) = \frac{T_{s,p} \cdot T_{c,q}}{\|T_{s,p}\| \|T_{c,q}\|}. \quad (2)$$

3.3.3 Key Similarity

Harmonic compatibility is essential when mixing songs, as it avoids dissonance and supports continuity between songs by enabling smooth transitions. We decided to use the key detection algorithm *KeyFinder* [24] due to its open-source availability and still good performance compared to recent state-of-the-art key detection algorithms. Additionally, we incorporate pitch shifting in the song selection process to be more flexible and less constrained by the harmonic aspect of the songs. As pitch-shifting algorithms can hurt the audio quality [25], we nonetheless want to keep pitch-shifts as small as possible. To this end, we identify key distances.

We define a harmonic key distance measure $D_{K_1}(K_2)$ as the minimum semitone distance between the tonic notes of two keys K_1 and K_2 . The key similarity measure $M_{K,c}$ is then defined as

$$M_{K,c} = \begin{cases} 1, & \text{if } D_{K_s}(K_c) = 0 \\ D_{K_s}(K_c)^{-1}, & \text{otherwise} \end{cases}, \quad (3)$$

where $D_{K_s}(K_c)$ is the key distance between the key K_s of the seed song s and the key K_c of the candidate song c .

3.3.4 Harmonic Similarity and Spectral Balance

Harmonic content and the energy across the low-, mid-, and high-frequency bands change throughout a song and thus must be reflected in the similarity measure. We compute the harmonic similarity and spectral balance measure, $M_{H,c}(k)$ and $M_{L,c}(k)$, respectively, based on the approach by Davies et al. [5].

3.3.5 Contextual Similarity

Mixing songs at positions with similar lyrics is a transition technique that could make the transition more related and seamless, independently of audio-based similarity. This method is commonly executed by playing a repeated phrase of the first song and then mixing in the second song with a similar vocal phrase.

Although lyrics are content information, they are often analyzed using contextual methods and are thus treated accordingly [26]. Due to the significant variation in lyrics across song sections, we find classical textual similarity measures such as *TF-IDF* unsuitable for our task. Instead, we capture the lyrics' similarity by extracting the whole lyrics' semantic meaning. We use Reimers and Gurevych [27] approach to compute sentence embeddings C_n over the lyrics of all songs n . The similarity measure $M_{C,c}$ is then calculated by computing the cosine-similarity between the sentence embedding C_s of the seed song s and the sentence embedding C_c of the candidate song c as

$$M_{C,c} = \frac{C_s \cdot C_c}{\|C_s\| \|C_c\|}. \quad (4)$$

3.3.6 Mixability

We compute the beat-wise mixability for a candidate song c against the phrase section p of the seed song s by combining the weighted similarity measures of rhythm, timbre, key, harmony, and spectral balance, as follows:

$$M_c(k) = \omega_R M_{R,c}(k) + \omega_T M_{T,c}(q) + \omega_K M_{K,c} + \omega_H M_{H,c}(k) + \omega_L M_{L,c}(k), \quad (5)$$

where q is the phrase section of c corresponding to the beat shift k . The mixability measure considers the 64 beats after the phrase section p of the seed song s instead of the entire phrase section p . This forward-moving approach enables us to maintain a song's dynamics by focusing on the upcoming segments instead of past segments. Through extensive, informal testing, we found the following weights to give the most convincing results: $\omega_R = 0.3$, $\omega_T = 0.75$, $\omega_K = 0.2$, $\omega_H = 0.2$, and $\omega_L = 0.1$.

To incorporate the contextual similarity measure, we extend the audio-based mixability measure $M_c(k)$ by the contextual similarity measure $M_{C,c}$ with the weight $\omega_C = 0.25$, as follows:

$$M'_c(k) = M_c(k) + \omega_C M_{C,c}. \quad (6)$$

Our initial experiments showed that choosing the transition point by selecting the beat shift k with the highest mixability score did not yield satisfactory results. Songs were transitioned at non-downbeat positions or unnatural downbeat intervals (e.g., 7, 9, 15, 17 downbeats), leading to a misaligned mix. To counteract this, we consider only beat shifts k that correspond to the segment boundary q of the candidate songs c and calculate the transition (cue) point as follows:

$$k_{\text{cue}}(c) = \arg \max_{k \in q} M_c(k). \quad (7)$$

We also record the timbral and rhythmic similarity at the transition point, $t_{\text{cue}}(c)$, $r_{\text{cue}}(c)$, and will use this information to improve the equalization in the mixing process.

We compute the song schedule by selecting the candidate song c with the highest mixability and extract the phrase section p for c up to the next segment boundary q , but at least for a minimum of λ_{minPlay} . We found that a λ_{minPlay} value of 55 seconds leads to a good balance between how long a song is played and how often songs are changed. We then select the phrase section p of c as the seed phrase section and repeat the process until the desired length of the mix is reached.

3.4 Mixing

Before transitioning, we first bring the loudness of each song to a consistent level of -14 LUFS. We then pitch-shift the audio to a harmonically compatible key and beat-match the song by time-stretching the audio to the same tempo as the previous song, using a maximal tempo change of $\pm 8\%$ as a limit. We use a transition length of 16 downbeats, where the transition starts with eight downbeats before the song excerpt's end and ends with eight downbeats after the transition point of the current song.

To prevent clashing frequency bands in the mix, we mainly base our equalization process on the "bass-swap" technique [28, Chapter 16] and extend it to the high-frequency band as well. A visualization of our standard equalization process is depicted in Figure 1.

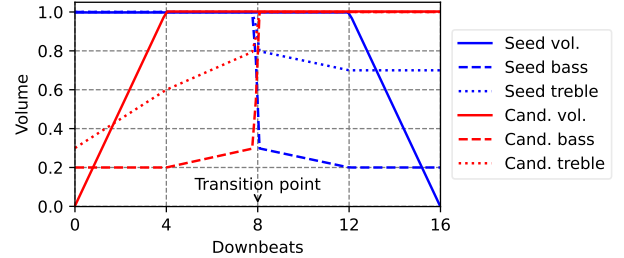


Figure 1. Standard equalization applied to both excerpts.

Overequalization can lead to a dull mix, which is why we will use information from our mixability calculation to identify and adjust problematic frequency ranges. We consider mid frequencies of songs with a dissimilar timbre ($t_{\text{cue}}(c) < 0.95$) as clashing and reduce the mid frequencies of the song that is currently playing, shifting the focus on the mid frequencies to the new song. In case of a high rhythmic similarity ($r_{\text{cue}}(c) \geq 0.95$), we apply less attenuation to the bass frequencies. Finally, we also assume that songs with an attenuated drum stem need even less equalization in the high frequencies, as drums, especially hi-hats, are a primary contributor to the high frequencies. We therefore introduce the high frequencies of the song that are to be mixed in earlier and with less attenuation.

3.4.1 Rule-based Stem Modification

We employ the pre-trained music source separation (MSS) model *HT Demucs* [29, 30] to separate the audio into the four stems: vocals, drums, bass, and other.

As previously noted, our tempo estimation algorithm predicts the tempo for only around 25% of songs with perfect accuracy. Even though the rhythmic similarity measure implicitly captures errors in tempo detection, rhythmic compatibility is only one of the components of the mixability measure, thus opening up the possibility of mixing in a rhythmic incompatible song. To counteract this, without entirely excluding rhythmic incompatible songs, we introduce a drum stem modification procedure for songs with rhythmic compatibility below $r_{\text{cue}}(c) < 0.95$.

Further, we generally want to prevent mixing song excerpts containing vocals, as vocal clashing can similarly lead to a reduced mix quality. We detect vocal segments by splitting the vocal stem, obtained by our MSS stage, into boundaries on "silent" sections that persist for one second or longer with a loudness below -40 dBFS and filter our vocal segments with a length below 400ms. We consider two song excerpts as clashing if the vocals during the transition intersect for more than two seconds and attenuate the vocals of the currently playing song. Figure 2 depicts the drum stem and vocal modification procedure.

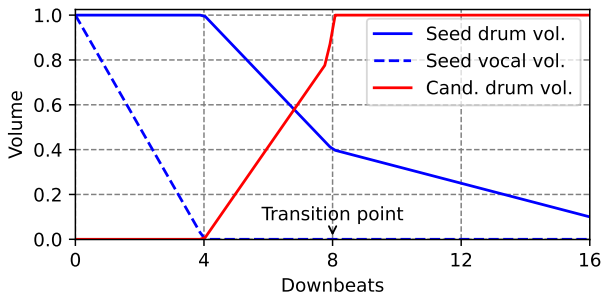


Figure 2. Equalization over the vocal and drum stems.

4. LISTENING EXPERIMENT

Due to the subjective nature of mixes [8] and the lack of ground truth corpora for similarity ratings between songs [22], we will evaluate the performance of our proposed solutions with qualitative methods, specifically using a listening experiment. For this, we developed a web-based survey that facilitates the evaluation of the models by human participants.

4.1 Models

We select AMU by Davies et al. [5] as our baseline model because it aligns with our methodology in prioritizing optimal mixing over runtime compromises and continues to be recognized as a relevant benchmark in recent research, such as [8]. To ensure a fair comparison of our models and counteract the negative influence of mismatched beats, we replace outdated components of AMU with state-of-the-art approaches. In particular, we replace their beat tracking and percussion detection method with our approaches and utilize our mixing procedure to create the mix.

To compare the performance of our models, we evaluate three models: MB_{base} , our approach without the stem modification and contextual information; MB_{stem} , our approach with the stem modification but without contextual information; and MB_{full} , our approach with the stem modification and contextual information (using M'_c). The code of our implementations is available open-source¹.

Note that, in order to maintain full control over the approaches and integration into a common interface, no commercial tools are included in the evaluation.

4.2 Setup

To understand the impact of musical knowledge on evaluation, we first ask the participants about their musical background and DJing experience. We split the following survey for each model into two parts. In the first part, we gather Song-Pair Compatibility (SPC) ratings by asking the participants to rate the song-scheduling aspect of the models. This allows us to compare the song selection of the models to the collected SPC ratings later on. For all pairs of songs, the participants assess the compatibility based on four categories by answering the following questions: *Timbre*: Are the songs similar regarding timbre?

¹ <https://github.com/robaerd/mosaikbox>

Rhythm: Do the songs have a similar rhythmic pattern?
Harmony: Do the songs have a similar harmonic structure?
Overall Mixable: Are the songs overall mixable?

In the second part, the participants are presented with the generated mix of a model and are asked to rate the overall quality of each transition of the mix on a scale of 1 (awful) to 5 (excellent), where 3 represents a neutral rating. The models are presented in random order to prevent presentation bias, with no details about the model type disclosed to the participants.

4.3 Dataset

Due to the tempo "lock-in", only songs with a tempo tolerance of maximum $\pm 8\%$ are considered. This commonly results in a genre "lock-in" as well, as songs of the same genre usually have a similar tempo. A preliminary poll among potential participants revealed that most are familiar with the drum and bass genre (DnB). We therefore decided to base our dataset on this genre to make the evaluation more relevant and valid.

We collected a dataset of 250 songs from the most popular DnB playlists of streaming services and randomly sampled 16 songs from this collection to use as input for the mix generation of the models. Out of these 16 songs, we sampled one song as the starting song for all models. To highlight the song selection aspect of the models, we used a top-k approach with $k=8$ for song selection instead of forcefully mixing all 16 songs.

5. RESULTS AND DISCUSSION

We recruited 30 participants (22 male/8 female), primarily academics aged 23-30 with backgrounds in STEM and economics, 8 of whom had prior experience in DJing. Among the participants, 10 classified their musical background as novice, 13 as intermediate, 7 as advanced, and none stated being a professional musician.

Model	Transition	SPC _{Timb}	SPC _{Rhy}	SPC _{Har}	SPC _{Mix}
AMU	2.490	0.457	0.505	0.429	0.624
MB_{base}	3.076	0.486	0.648	0.505	0.648
MB_{stem}	3.457	0.486	0.648	0.505	0.648
MB_{full}	3.033	0.500	0.619	0.529	0.705

Table 2. Average transition and SPC ratings for all models from all transitions. MB_{base} and MB_{stem} share identical SPC ratings due to the same song selection.

Table 2 shows that all our models significantly outperformed the AMU baseline in average transition and SPC ratings. The MB_{full} model received the highest SPC ratings for timbre, harmony, and mixability, while the $MB_{base, stem}$ models scored higher in rhythm and the MB_{stem} model achieved the best average transition rating.

In Figure 3, we can observe that AMU mostly received negative ratings, while those of MB_{full} had a more consistent distribution, declining towards the end of the mix. Except for the first two transitions, MB_{stem} consistently

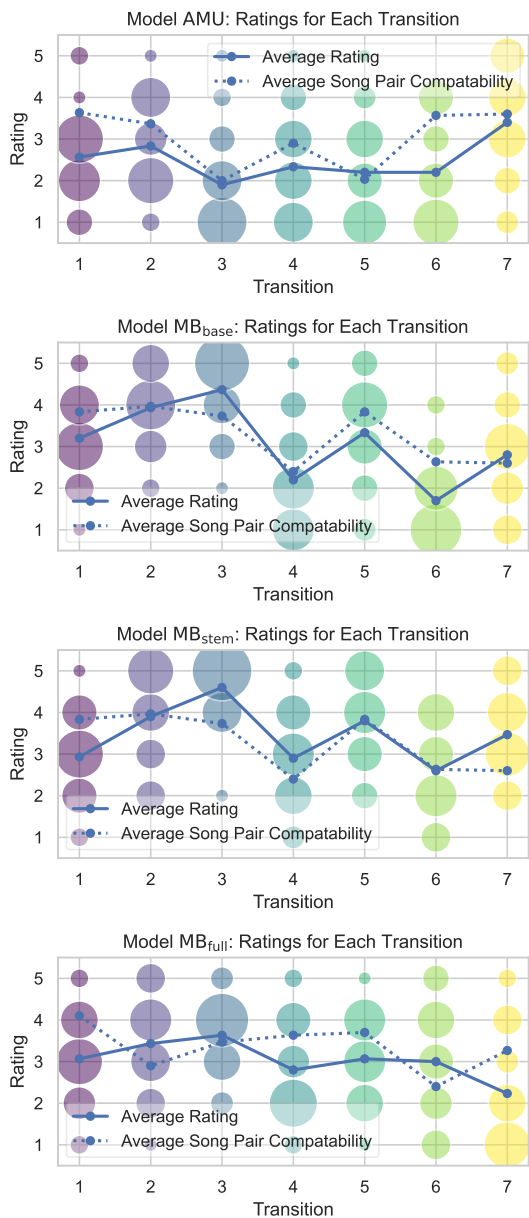


Figure 3. Transition ratings across all models, with bubble size indicating the number of ratings per transition and lines representing each transition’s average rating and SPC.

outperformed MB_{base}, suggesting that stem separation positively impacts mix quality.

After confirming non-normality with the Shapiro-Wilk test, we test for significant differences using the Friedman test, resulting in a p-value < 0.00001. To determine the best-performing model, we then conduct Wilcoxon signed-rank tests for pairwise comparisons. To account for the family-wise error rate, we correct the p-values using the Holm-Bonferroni method and reject the null hypothesis if the corrected p-value \hat{p} is less than the significance level $\alpha = 0.05$. The results in Table 3 show that all our models significantly outperform AMU, while MB_{stem} significantly outperforms its base counterpart MB_{base}. No significant difference is found between the MB_{full} and MB_{stem} models,

which suggests that contextual information does not have a significant impact on the mix quality.

Model 1 (F)	Model 2 (G)	\hat{p} -value _{F(u)<G(u)}
MB _{base}	AMU	< 0.0001
MB _{stem}	AMU	< 0.0001
MB _{full}	AMU	< 0.0001
MB _{stem}	MB _{base}	< 0.0001
MB _{full}	MB _{base}	×
MB _{full}	MB _{stem}	×

Table 3. Pairwise tests for significance between models showing corrected p-value levels of the Wilcoxon signed-rank test (‘×’ means no significance at 0.05 level).

Further significance tests using Mann-Whitney U tests revealed a significant difference in ratings between DJ experience and all musical knowledge levels only for the baseline model AMU, with p-values of 0.001 and 0.0001, respectively. Participants with DJing experience rated the AMU model significantly worse. Analogous, based on the mean ranks of the transition ratings, the higher the musical knowledge level, the worse the rating.

Finally, we tested for significance of the Pearson correlation between transition ratings and the averaged SPC values, indicating a significant strong correlation for MB_{base} with ($r = 0.83, p = 0.02$), suggesting mixes align closely with participant expectations. In contrast, there was a moderate non-significant correlation ($r = 0.6, p = 0.1$) for AMU and MB_{stem} and no significant correlation for MB_{full} ($r = -0.03, p = 0.95$).

The performance gains of our models over AMU in SPC ratings may stem from our rhythmic similarity calculation and the integration of timbral and key similarities into the mixability estimate. Improved transition ratings could be linked to our updated structural segmentation approach. Higher timbre, harmony, and mixability SPC ratings, alongside lower rhythm ratings, might be influenced by mood-related contextual similarities. Lower transition ratings could stem from the lesser relevance of lyrics’ semantic meaning in DnB. The listening experiment results are available online.²

6. CONCLUSION

In this paper, we proposed the automatic mixing system *Mosaikbox* and demonstrated that it outperforms comparable state-of-the-art systems. We showed that our rule-based stem modification significantly improves the overall mix quality. However, we could not show that including contextual information has any significant positive impact on the mix quality.

Future work will include the impact of new features, such as the energy level of songs and the use of similarity measures obtained by collaborative filtering approaches. In addition, a dynamic transition length will be explored to enhance creativity and adaptability across various genres.

² <https://github.com/robaerd/mosaikbox-survey>

7. ACKNOWLEDGMENTS

This research was funded in whole or in part by the Austrian Science Fund (FWF) (10.55776/P33526). For open access purposes, the author has applied a CC BY public copyright license to any author-accepted manuscript version arising from this submission.

8. ETHICS STATEMENT

One of the main factors in the similarity measure of *Mosaikbox* is timbre, which could lead to a bias towards songs of the same artist or label, neglecting songs of other artists, due to production effects captured by MFCCs (a phenomenon often referred to as album or artist effect [31, 32]). This constitutes a technical algorithmic bias, cf. [33,34]. In the interest of transparency and ethical rigor, we also acknowledge a potential bias in the participant demographics. The participants were mainly academics aged 23-30, with a majority having backgrounds in STEM and economics and a significant portion having familiarity with the drum and bass genre.

Additionally, the rule-based approach of our system offers the advantage of not requiring training on a large dataset of copyrighted music, including DJ interpretations. However, this does not remove the issues related to automating a craft traditionally performed by humans. This raises several concerns, including potential impacts on artistic expression and reception, the role of human creativity, and the future of DJing as a skilled profession.

Furthermore, as with every form of automation, a general adoption of automatic music mixing systems could significantly reduce the demand for DJs, especially in smaller venues. However, automatic music mixing systems, such as ours, can also be used as a tool by DJs to explore new ideas, get suggestions for transitions they might not have thought of, break out of their comfort zone, and increase diversity and creativity in mixes, if designed accordingly.

9. REFERENCES

- [1] F. Broughton and B. Brewster, *How to DJ Right: The Art and Science of Playing Records*. Grove/Atlantic, Inc., Dec. 2007.
- [2] T. Jehan, “Creating Music by Listening,” Ph.D. dissertation, Massachusetts Institute of Technology, Jan. 2005.
- [3] H.-Y. Lin, Y.-T. Lin, and M.-C. Tien, “Music Paste: Concatenating Music Clips based on Chroma and Rhythm Features,” in *Proceedings of the 10th International Society for Music Information Retrieval Conference*. ISMIR, Jan. 2009, pp. 213–218.
- [4] H. Ishizaki, K. Hoashi, and Y. Takishima, “Full-Automatic DJ Mixing System with Optimal Tempo Adjustment based on Measurement Function of User Discomfort,” in *Proceedings of the 10th International Society for Music Information Retrieval Conference*. ISMIR, Jan. 2009, pp. 135–140.
- [5] M. Davies, P. Hamel, K. Yoshii, and M. Goto, “AutoMashUpper: Automatic Creation of Multi-Song Music Mashups,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 12, pp. 1726–1737, Dec. 2014.
- [6] T. Hirai, H. Doi, and S. Morishima, “MusicMixer: computer-aided DJ system based on an automatic song mixing,” in *Proceedings of the 12th International Conference on Advances in Computer Entertainment Technology*. Association for Computing Machinery, Nov. 2015, pp. 1–5.
- [7] —, “Musicmixer: Automatic dj system considering beat and latent topic similarity,” in *MultiMedia Modeling - 22nd International Conference*, Q. Tian, R. Hong, X. Liu, N. Sebe, B. Huet, and G.-J. Qi, Eds. Springer International Publishing, 2016, pp. 698–709.
- [8] J. Huang, J.-C. Wang, J. B. L. Smith, X. Song, and Y. Wang, “Modeling the Compatibility of Stem Tracks to Generate Music Mashups,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 1, pp. 187–195, May 2021.
- [9] K. Seyerlehner, G. Widmer, and T. Pohle, “Fusing Block-level Features for Music Similarity Estimation,” in *Proceedings of the 13th International Conference on Digital Audio Effects (DAFx-10)*. DAFx, 2010.
- [10] X. Hu, J. Downie, and A. Ehmman, “Lyric Text Mining in Music Mood Classification,” in *Proceedings of the 10th International Society for Music Information Retrieval Conference*. ISMIR, Jan. 2009, pp. 411–416.
- [11] L. Vande Veire and T. De Bie, “From raw audio to a seamless mix: creating an automated DJ system for Drum and Bass,” *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2018, no. 134, pp. 1–21, Sep. 2018.
- [12] M. Heydari, F. Cwitkowitz, and Z. Duan, “BeatNet: CRNN and Particle Filtering for Online Joint Beat Downbeat and Meter Tracking,” in *Proceedings of the 22nd International Society for Music Information Retrieval Conference*. ISMIR, 2021.
- [13] Y. Xiang, S. Gubian, B. Suomela, and J. Hoeng, “Generalized Simulated Annealing for Global Optimization: The GenSA Package,” *The R Journal*, vol. 5, no. 1, pp. 13–28, 2013.
- [14] P. Knees, Á. Faraldo, P. Herrera, R. Vogl, S. Böck, F. Hörschläger, and M. L. Goff, “Two Data Sets for Tempo Estimation and Key Detection in Electronic Dance Music Annotated from User Corrections,” in *Proceedings of the 16th International Society for Music Information Retrieval Conference*. ISMIR, Sep. 2018, pp. 364–370.

- [15] H. Schreiber and M. Müller, “A Crowdsourced Experiment for Tempo Estimation of Electronic Dance Music,” in *Proceedings of the 19th International Society for Music Information Retrieval Conference*. ISMIR, 2018.
- [16] S. Böck and M. Davies, “Deconstruct, analyse, reconstruct: How to improve tempo, beat, and downbeat estimation,” in *Proceedings of the 21st International Society for Music Information Retrieval Conference*. ISMIR, Nov. 2020, pp. 574–582.
- [17] F. Gouyon, A. Klapuri, S. Dixon, M. Alonso, G. Tzanetakis, C. Uhle, and P. Cano, “An experimental comparison of audio tempo induction algorithms,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 5, pp. 1832–1844, Sep. 2006.
- [18] J. Serrà, M. Müller, P. Grosche, and J. L. Arcos, “Unsupervised Music Structure Annotation by Time Series Structure Features and Segment Similarity,” *IEEE Transactions on Multimedia*, vol. 16, no. 5, pp. 1229–1240, Aug. 2014.
- [19] O. Nieto and J. P. Bello, “Music segment similarity using 2D-Fourier Magnitude Coefficients,” in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Florence, Italy: IEEE, May 2014, pp. 664–668.
- [20] C. Southall, R. Stables, and J. Hockman, “Automatic Drum Transcription Using Bi-Directional Recurrent Neural Networks,” in *Proceedings of the 17th International Society for Music Information Retrieval Conference*. ISMIR, Sep. 2016, pp. 591–597.
- [21] —, “Automatic drum transcription for polyphonic recordings using soft attention mechanisms and convolutional neural networks,” in *Proceedings of the 18th International Society for Music Information Retrieval Conference*. ISMIR, 2017.
- [22] B. Rocha, N. Bogaards, and A. Honingh, “Segmentation and timbre- and rhythm-similarity in Electronic Dance Music,” University of Amsterdam, Elephantcandy, Tech. Rep., Apr. 2013.
- [23] M. Panteli, B. Rocha, N. Bogaards, and A. Honingh, “A model for rhythm and timbre similarity in electronic dance music,” *Musicae Scientiae*, vol. 21, no. 3, pp. 338–361, Sep. 2017.
- [24] I. Sha’ath, “Estimation of key in digital music recordings, MSc Computer Science Project Report,” Master’s thesis, Birkbeck College, University of London, 2011.
- [25] T. Royer, “Pitch-shifting algorithm design and applications in music,” Master’s thesis, KTH Royal Institute of Technology, School of Electrical Engineering and Computer Science, 2019.
- [26] P. Knees and M. Schedl, *Music Similarity and Retrieval*, ser. The Information Retrieval Series. Springer Berlin Heidelberg, 2016, vol. 36.
- [27] N. Reimers and I. Gurevych, “Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks,” in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Nov. 2019, pp. 3982–3992.
- [28] J. Steventon, *DJing For Dummies*, 2nd ed. For Dummies, Sep. 2010.
- [29] S. Rouard, F. Massa, and A. Défossez, “Hybrid transformers for music source separation,” in *2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023.
- [30] A. Défossez, N. Usunier, L. Bottou, and F. Bach, “Music Source Separation in the Waveform Domain,” Apr. 2021, arXiv preprint arXiv:1911.13254.
- [31] A. Flexer and D. Schnitzer, “Effects of album and artist filters in audio similarity computed for very large music databases,” *Computer Music Journal*, vol. 34, no. 3, p. 20–28, sep 2010. [Online]. Available: https://doi.org/10.1162/COMJ_a_00004
- [32] I. Vatolkin, G. Rudolph, and C. Weihs, “Evaluation of Album Effect for Feature Selection in Music Genre Recognition,” in *Proceedings of the 16th International Society for Music Information Retrieval Conference*. ISMIR, Sep. 2018, pp. 169–175. [Online]. Available: <https://doi.org/10.5281/zenodo.1416328>
- [33] A. Flexer, M. Dörfler, J. Schlüter, and T. Grill, “Hubness as a case of technical algorithmic bias in music recommendation,” in *Proceedings of the 2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, 2018, pp. 1062–1069.
- [34] A. Holzapfel, B. L. Sturm, and M. Coeckelbergh, “Ethical dimensions of music information retrieval technology,” *Transactions of the International Society for Music Information Retrieval*, Sep 2018.

MIDICAPS: A LARGE-SCALE MIDI DATASET WITH TEXT CAPTIONS

Jan Melechovsky *, Abhinaba Roy *, Dorien Herremans

Singapore University of Technology and Design

jan_melechovsky@mymail.sutd.edu.sg, abhinaba_roy@sutd.edu.sg, dorien_herremans@sutd.edu.sg

ABSTRACT

Generative models guided by text prompts are increasingly becoming more popular. However, no text-to-MIDI models currently exist due to the lack of a captioned MIDI dataset. This work aims to enable research that combines LLMs with symbolic music by presenting **MidiCaps**, the first openly available large-scale MIDI dataset with text captions. MIDI (Musical Instrument Digital Interface) files are widely used for encoding musical information and can capture the nuances of musical composition. They are widely used by music producers, composers, musicologists, and performers alike. Inspired by recent advancements in captioning techniques, we present a curated dataset of over 168k MIDI files with textual descriptions. Each MIDI caption describes the musical content, including tempo, chord progression, time signature, instruments, genre, and mood, thus facilitating multi-modal exploration and analysis. The dataset encompasses various genres, styles, and complexities, offering a rich data source for training and evaluating models for tasks such as music information retrieval, music understanding, and cross-modal translation. We provide detailed statistics about the dataset and have assessed the quality of the captions in an extensive listening study. We anticipate that this resource will stimulate further research at the intersection of music and natural language processing, fostering advancements in both fields.

1. INTRODUCTION

The recent development of large-language models (LLMs) has revolutionised how we interact with text, images, and even audio. By incorporating elements of multimodal learning, researchers have combined LLMs with other modalities. The resulting models can analyze and generate accurate descriptions and captions, which in turn facilitates downstream tasks such as question answering [1], image generation [2], and music generation [3]. However, we have yet to see such an evolution for MIDI files.

*These authors contributed equally to this work.

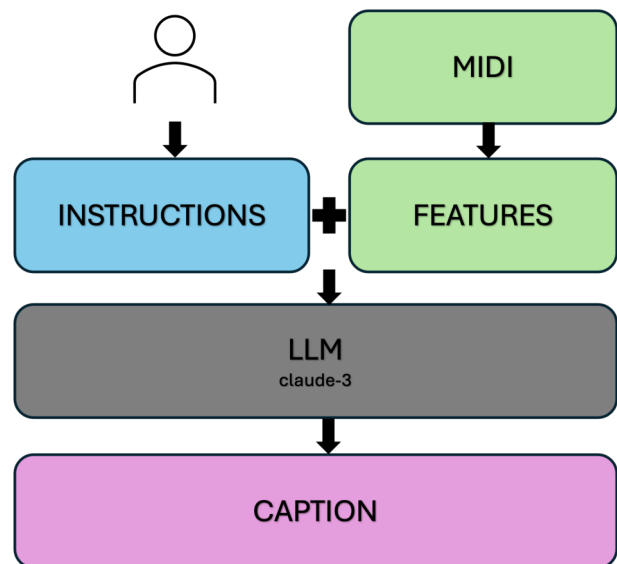


Figure 1: Overview of our approach. We extract meaningful and relevant features from MIDI files. These features are then added as tags to the human instructions that are sent to an LLM (Claude-3) to generate meaningful text captions of MIDI files.

In the field of Music Information Retrieval (MIR), MIDI plays a crucial role as a symbolic and musically meaningful representation of music. The format is often used by music producers and composers working in Digital Audio Workstations (DAWs). It is also a useful format for the computational analysis of music and related tasks such as music transcription, genre classification, similarity measurement, and music recommendation [4]. Furthermore, due to the symbolic nature of music, it has long been used by music generation algorithms [5]. In recent years, we have seen a surge of interest in music generation from free-flow text instructions [3, 6–9]. These studies leverage the expressive capabilities of LLMs to translate textual representation of musical attributes into actual music audio. This necessitates a meticulous alignment between the textual and musical feature spaces to ensure that the generated music closely follows text instructions. To validate and establish benchmarks for this text-to-music mapping, large-scale datasets with text captions have been developed [3, 10].

No such efforts, however, have yet been made for the MIDI format, despite its widespread use by musicians and its obvious, historically supported usage in music generation. This lack of text-MIDI datasets, in turn, has inhibited

researchers from exploring interesting and novel tasks such as MIDI generation from free-flow text prompts. In this work, we identify this shortcoming and develop a robust solution in the form of a large-scale curated MIDI dataset accompanied by text captions. Our goal is to obtain captions that are i) large in volume, ii) contain accurate information about the musical contents, and iii) feature a rich and refined vocabulary. We posit that such a dataset-level approach opens up further opportunities for researchers in MIDI-LLMs-related tasks.

To address the first goal, we identify an open source large-scale MIDI dataset in the form of Lakh MIDI dataset [11], that contains over 170K MIDI examples. Second, to attain the musical contents in each MIDI file, we extract meaningful features encompassing tempo, chord progression, time signature, instruments present, genre and mood. Each of the features are extracted using state-of-the-art MIR tools that ensure the quality and accuracy of the features extracted. After feature extraction, we are still left with the task of caption generation. Relying on traditional human annotation is tedious, time-consuming, and costly. Instead, motivated by the recent success of LLMs, we utilize in-context learning – a model’s ability to temporarily learn from human-provided instructions [12]. Our decision is motivated by Melechovsky et al. [3], who have demonstrated the efficacy of in-context learning in generating captions that are accurate, rich in description as well as grammatically coherent. In our approach, we furnish the LLM with instructions to generate captions based on the extracted music features, supplemented by a small set of feature-caption pairs created by expert annotators. Given the current absence of freely available MIDI-caption datasets, we anticipate that the provision of a substantial volume of detailed and informative captions will inspire the research community to delve further into tasks related to MIDI and Large Language Models (LLMs). The main contributions of this work can be summarized as follows:

- We introduce the first curated large-scale open dataset of MIDI-caption pairs, termed **MidiCaps**¹.
- Furthermore, we present a comprehensive set of music-specific features extracted from MIDI files. These features succinctly characterize the musical content, encompassing tempo, chord progression, time signature, instrument presence, genre, and mood.
- Finally, we provide a text caption annotation framework tailored specifically for MIDI data (see Figure 1). Leveraging the in-context learning capability of large language models (LLMs), we enable the generation of captions using only a small number of feature-caption training pairs. This framework, a first of its kind, is made freely accessible to users², facilitating the generation of MIDI-caption pairs for their individual MIDI files.

2. RELATED WORK

To the best of our knowledge, there are no publicly available MIDI caption datasets. In this section, we briefly mention various publicly available MIDI datasets and discuss the closely related topic of caption generation from audio and music.

Despite the scarcity of MIDI caption datasets, existing repositories offer potential resources that could be adapted for this purpose. Among these, the Lakh MIDI Dataset [11] stands out, comprising a vast collection of MIDI files. While primarily tailored for MIR tasks such as melody extraction and chord estimation, its volume and diversity present an opportunity for repurposing towards captioning tasks, albeit requiring appropriate preprocessing. The MAESTRO Dataset [13] offers aligned pairs of MIDI and audio files, primarily for piano music generation. The MuseGAN Dataset [14] focuses on multi-track songs, and the MAPS Dataset [15], contains recordings of classical piano pieces alongside aligned MIDI files and thus also present potential avenues for MIDI captioning research. Additionally, the Wikifonia Dataset [16] features a substantial collection of lead sheets accompanied by MIDI files. Closest to our proposed MIDI-caption dataset is the WikiMusicText (WikiMT) dataset [17], which includes lead sheets in ABC notation with metadata including text descriptions. These descriptions, however, pertain to general information about the music piece rather than detailed descriptions of musical contents provided in MIDI files within our captions.

In the last three years, several models were released for automatic caption generation from music `audio` files. One of the earlier models, MusCaps [18], uses an architecture based on recurrent and convolutional layers as well as a multimodal encoder. Recent research includes the use of large language models (LLMs) for captioning [3, 7, 10]. In [7], a pseudo labeling approach is used to label a large training dataset. First, existing captions from other datasets are curated, then the MuLaN [19] model, a joint music-text embedding model, evaluates the distance between captions and unlabeled audio files. The top caption candidates are selected based on their frequency to ensure balance among all samples. In [20], the focus is on capturing the full sentiment of classical music recordings through text descriptions, introducing a Group-Topology Preservation Loss to be used with their cross-modality translation model. A recent study by Doh et al. [10] targets pseudo labeling of audio data with the help of an LLM, utilizing the MusicCaps [6] dataset as ground truth and instructing GPT-3.5 Turbo [21] to generate full captions from these tags.

In [3], Melechovsky et al. curate a new dataset based on the MusicCaps dataset [6], called MusicBench. In MusicBench, the original captions are enhanced by including additional music descriptors such as chord sequence, musical key, time signature, and tempo. After performing audio and text augmentations to expand the dataset size, they use ChatGPT [22] for rephrasing captions to create more diverse captions. Furthermore, they employ in-context learning to guide ChatGPT using a small set

¹huggingface.co/datasets/amaai-lab/MidiCaps

²github.com/AMAAI-Lab/MidiCaps

of human-annotated examples, instructing it to generate diverse captions to create an evaluation dataset from extracted tags, named FMACaps. Inspired by their methodology, we adopt a similar approach and utilize in-context learning alongside a large-language model to generate captions from MIDI features. In the subsequent section, we offer an in-depth description of our proposed framework for MIDI captioning.

3. METHOD

In this section, we discuss details regarding the music-specific features we extract from MIDI files.

3.1 Feature extraction

In a first step, as per Figure 2, we extract various musical features from the MIDI files. This is achieved in two ways: a number of features are extracted directly from the MIDI files, and others are extracted from the synthesized MIDI files. The details of our approach are described below.

3.1.1 Preprocessing

We preprocess all files to remove faulty files. For instance, we found multiple files that had never-ending notes. Using Mido [23], we further exclude files of duration shorter than 3 seconds and longer than 15 minutes.

3.1.2 MIDI feature extraction

We use Music21 [24] and Mido [23] libraries to extract the following features from MIDI: Musical Key (Music21), Time Signature (Music21), Tempo (Mido), Duration of the MIDI file (Mido), and a list of Instruments (Mido).

The **Key** and **Time Signature** features are obtained through `music21.midi.analyze('keys')` and `music21.midi.getTimeSignatures()` functions, respectively. To calculate the **Tempo**, we first look for the `set_tempo` MIDI message to get the MIDI tempo. Then, the `mido.tempo2bpm()` function is used to convert this MIDI tempo to beats per minute (bpm). For MIDI file **Duration**, we retrieve the `length` attribute of a `mido.MidiFile` object.

To extract a list of **instruments**, we filter the MIDI messages based on channel number and their associated instrument program obtained from the program change message. To treat ambiguity given by some faulty files, we always take the last assigned program number as the definite instrument number for each MIDI channel. For channel 10, which is reserved for drums, we always consider the assigned instrument to be drums, unless there is another percussion instrument specified.

We further process the extracted instruments in three steps to identify the most prominent instruments. First, we extract total note duration for each of the instruments by scraping through note-on and note-off messages, and rank them by this duration. Second, we map the program numbers to their respective instrument names, grouping similar variations (e.g., both nylon and steel string acoustic guitars

as ‘acoustic guitar’). Third, we reduce the list of instruments to only include one instance of the same instrument name (in the previous example, the two acoustic guitars would merge into one), and then take top five instruments sorted by their total note duration.

3.1.3 Synthesized audio feature extraction

We use the Midi2Audio library [25] that utilizes FluidSynth [26, 27] to synthesize audio from MIDI with the Fluid Release 3 General-MIDI sound font. Then, we use these audio files to extract genre, mood, and chord features.

To extract **genre** and **mood**, we use Essentia [28], specifically the MTG-Jamendo genre and mood/theme discogs effnet models³. We keep the top two genre tags with the highest confidence score, and the top five mood/theme tags, also based on their confidence score. The confidence scores for each tag are also stored.

Next, we extract the single most occurring **chord sequence** of length 3 to 5. To obtain this, we first extract all chords from the audio using Chordino [29]. To obtain the most frequent short chord sequence, we first iterate through the chord list to find the most frequent patterns consisting of 3, 4, and 5 consequent chords. We do not allow these patterns to have the same first and last chord, e.g., [A, B, C, A] for a pattern of length 4 is not allowed, as this is likely an [A, B, C] pattern of length 3. Then, we decide on which pattern to keep through a set of rules described in Algorithm 1. In the below algorithm n_i represents the occurrence count of the most frequent pattern (p_i) of length i . We save the final selected pattern along with a number denoting how many times it occurred. Once we have extracted all of the features extracted, we move on to caption generation, described in the next subsection.

Algorithm 1 Selecting the most frequent chord pattern.

```

▷  $p_i$ : most frequent pattern of length  $i$ 
▷  $n_i$ : occurrence count of  $p_i$ 
▷  $p$ : final selected most frequent pattern
 $n = n_3 + n_4 + n_5$ 
if ( $n_5 \geq 0.8 \cdot n_4$ ) & ( $n_5 \geq 0.25 \cdot n$ ) then
     $p \leftarrow p_5$ 
else if ( $n_4 \geq 0.8 \cdot n_3$ ) & ( $n_4 \geq 0.3 \cdot n$ ) then
     $p \leftarrow p_4$ 
else if ( $n_3 == 0$ ) then
    if ( $n_4 == 0$ ) then
        if ( $n_5 == 0$ ) then
             $p \leftarrow \text{None}$ 
        else
             $p \leftarrow p_5$ 
        end if
    else
         $p \leftarrow p_4$ 
    end if
else
     $p \leftarrow p_3$ 
end if
    
```

³essentia.upf.edu/models.html#discogs-effnet

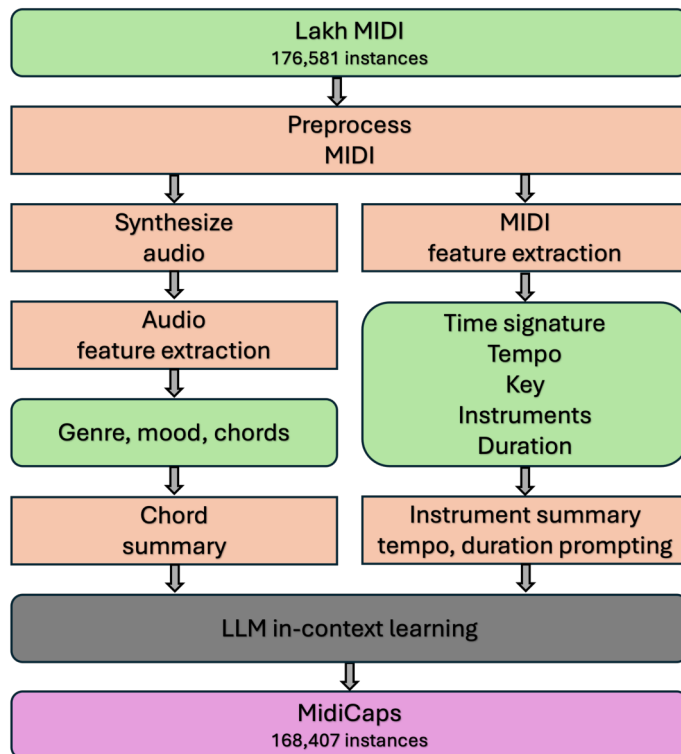


Figure 2: Detailed overview of our proposed captioning framework.

3.2 Caption generation

In this step, we take the extracted features and execute caption generation. To harness the expressive power and few-shot capability of a Large Language Model (LLM), we refer to a recent benchmarking study on LLMs [30], and ultimately selected Claude 3 Opus [31] due to its superior performance compared to other LLMs such as GPT4. Employing in-context learning (a task in which the LLM is given example data of paired input-output to serve as ‘context’, and is expected to continue producing outputs for new unpaired inputs in a similar manner), we begin by selecting 17⁴ diverse examples from the extracted features and request a human annotator to craft appropriate text captions for each of these based solely on the extracted features. This approach aims to prevent any auditory influence on human captioning, as Claude 3 (or any LLM, for that matter) will subsequently only process text inputs, not audio files. Once the 17 examples are prepared, we construct a text prompt instructing Claude 3 to analyze the human-prepared feature-caption pairs and generate suitable captions for new sets of features. To maintain clarity, we specify that the generated captions should be between three to seven sentences. Before generating captions for all 168K MIDI files, we conduct a sanity check on ten examples to evaluate Claude 3’s response to in-context learning, ensuring our prompt does not produce unrelated output or “hallucinate.” Please note, this check differs from the quality evaluation of the generated captions reported in the next section. In our study, a single round of sanity checks sufficed, obviating the need to modify prompts or alter the feature-caption pairs for in-context learning. Finally, us-

⁴Optimized based on limit on input tokens in Claude 3 text prompts.

ing the features extracted from each MIDI file, we generate corresponding captions, creating our proposed **MidiCaps** dataset, which we describe in detail in the next section.

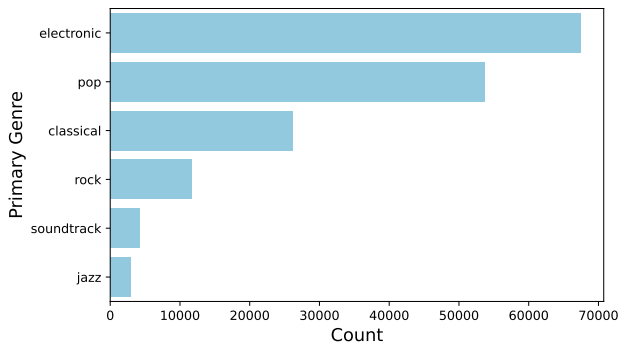
4. EVALUATION AND STATISTICS

In this section, we first introduce the **MidiCaps** dataset and subsequently detail subjective evaluation in form of listening study.

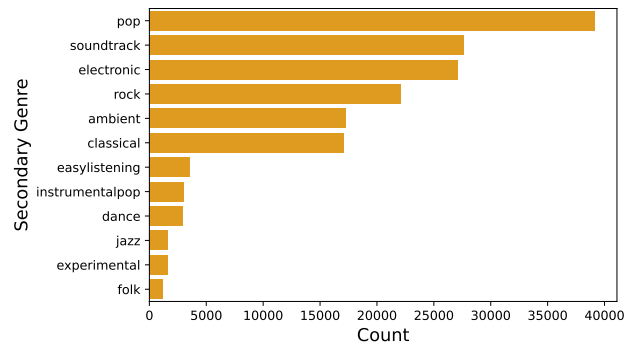
4.1 MidiCaps dataset

To generate our **MidiCaps** dataset, we start with MIDI files provided in the Lakh MIDI dataset [11], comprised of a collection of 176,581 unique MIDI files, designed to facilitate large-scale music information retrieval. Additionally, the dataset is distributed under a CC-BY 4.0 license, enabling us to expand the dataset without encountering copyright constraints. Subsequently, we process the raw MIDI files and extract musical features as described in Section 3.1, which we used in the captioning process Section 3.2 to create our final **MidiCaps** dataset consisting of 168,407 MIDI files with matching text caption. A couple of examples of captions generated are provided below. They encapsulate key information regarding the music contents while infusing a fluid human touch:

1. “A melodic and happy rock and pop song featuring a string ensemble, piano, clean electric guitar, slap bass, and drums. The song is in the key of F major with a 4/4 time signature and a tempo of 120 BPM. The chord progression alternates between Bb and F throughout the song, creating a motivational and energetic corporate vibe.”

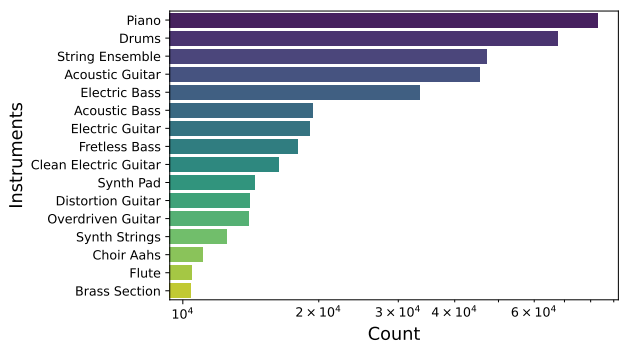


(a) Distribution of primary genre.

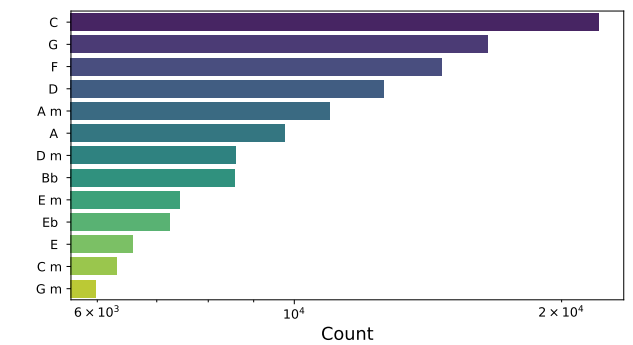


(b) Distribution of secondary genre.

Figure 3: Genre distributions.

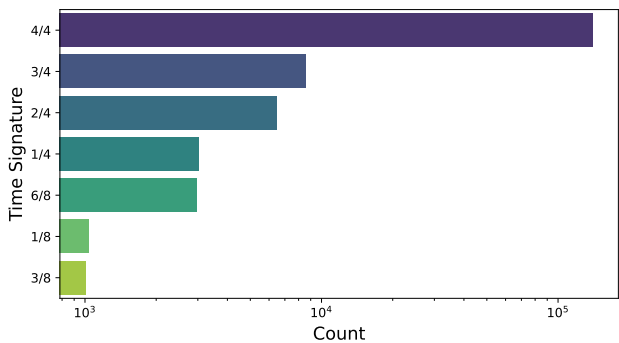


(a) Distribution of instruments present in the summary.

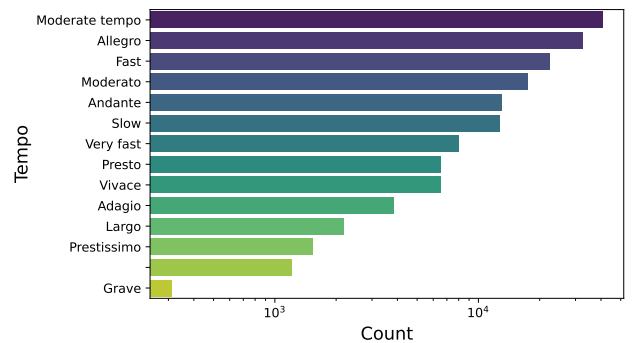


(b) Distribution of key.

Figure 4: Instrument and key distributions (in log scale).



(a) Distribution of time signature.



(b) Distribution of tempo.

Figure 5: Time signature and tempo distributions (in log scale).

2. “A melodic and happy pop song with a Christmas vibe, featuring piano, clean electric guitar, acoustic guitar, and overdriven guitar. The song is in the key of A major with a 4/4 time signature and a moderate tempo. The chord progression revolves around D, E6, D, and E, creating a motivational and loving atmosphere throughout the piece.”

Moreover, we provide a summary of some of the extracted features below to gain further insight into the diversity within the dataset. In Figure 3a, we illustrate the distribution of the primary (highest confidence score) and secondary (second highest confidence score) genres present in the dataset. In both cases, electronic and pop genres are

most prominent in the dataset. The secondary genre exhibits more variation, such as folk, instrumental pop, and easy listening, which have more occurrences as secondary genre but do not appear in the primary genre figure. This means that they can be used by the captioning system to further specify and narrow down the broad primary genres (e.g. classical) into more specific descriptions such as ‘ambient classical’ etc. Please note that only genres with more than 1,000 occurrences are displayed in the figures. Figure 4 summarizes the instruments and keys present in the dataset. Piano, drums, and various types of guitars are predominant in the instrument summary, corroborating the fact that a significant portion of the songs belongs to electronic, pop, and rock genres. Additionally, the keys

Audience: Annotated by:	General audience		Music experts	
	Human	AI	Human	AI
Question	Avg. rating (1-7)			
Overall matching	5.46	5.63	5.40	4.92
Human-like	5.21	5.32	5.09	4.98
Genre matching	5.80	5.63	5.54	4.73
Mood matching	5.50	5.62	5.43	4.82
Key matching	5.87	5.70	5.51	5.69
Chord matching	6.12	5.78	5.74	5.09
Tempo matching	5.71	5.86	5.37	5.77

Table 1: Results of the listening study. Each question is rated on a Likert scale from 1 (very bad) to 7 (very good). The table shows the average ratings per question for each group of participants.

of C, G, F, and D major have the highest occurrences in the dataset. Regarding time signature, 4/4 is significantly more common than any other (Figure 5a), while most songs follow a moderate tempo (Figure 5b).

4.2 Listening study

Since there is no ground truth or baseline model to compare our new dataset to, we conduct a listening study with the help of the PsyToolkit platform [32,33]. Listeners were asked to listen to 20 MIDI files, chosen at random, from which 15 are captioned by our framework and 5 are annotated by an expert human rater with absolute pitch. Then, listeners were asked to rate these captions in seven aspects, which are: 1) Overall matching of caption to audio, 2) How human-like the caption is, 3) Genre matching of caption with audio, 4) Mood matching, 5) Key matching, 6) Chord matching, and 7) Tempo matching. Those listeners who indicated that they do not have the ability to recognize chords/key were tagged as General audience. A total of 16 participants belong to this general audience, of which 25% has more than 1 year of musical training. Another 7 participants indicated that they can recognize chords and key or have absolute pitch. These were tagged as Music experts.

4.3 Results and discussion

Table 1 shows the results of the listening study. The average rating for overall matching of the text caption with the MIDI file for the general audience is even slightly higher (5.63) for the AI generated caption compared to the human-written caption (5.46). When it comes to the ratings by music experts, the overall matching rating is slightly lower, but still well above average (4.92). In term of how human-like the captions are, the general audience again provides high ratings, comparable to those given to the human-written captions (5.21). The music experts are slightly more critical and rate them at 4.98, which is still very close to their rating for human-written captions (5.09). A similar pattern can be seen for ratings of genre matching and mood matching. The ratings for tempo matching outperformed the human-written ones for both general audience and music experts.

In terms of key and chord matching, the general audience provide good ratings. For these questions the ratings from the music experts, however, are more reliable, as these participants have explicitly indicated that they are able to recognize chords and keys. Their rating for key matching (5.69) is on par with the rating for human-written captions (5.51), and confirm the high agreement that the musical key described in the caption matches the audio pieces. For chord matching, the music experts’ average rating of 5.09 falls below the rating for the human-written caption. Please note, however, that this particular question was not easy to answer. Extracting a single ‘main’ pattern (3-5 chords) from the entire list of extracted chords is challenging as there are many different cases, e.g., very short fragments of a few chords, and very long pieces with many chord patterns. Slight changes in chord patterns can also be intentional, e.g., a chord progression of C, G, D, C, G, D6 would likely be detected as a C, G, D, C, G pattern instead of a C, G, D variation. All this makes it hard to objectively judge a single-chord pattern in the text captions. Despite this, the chord matching rating of 5.09 provides support that our caption contains a matching chord summary. Overall, the results from the listening study support that our text captions provide a high-quality, human-like textual description that matches the MIDI files well.

The task of automatically labelling files of various length is difficult by nature as longer music pieces might require more text to be described precisely, while shorter pieces may need only a single sentence. This problem is further magnified when considering chord progressions and their summary as mentioned above. Additionally, extracting features from synthesized audio files is not optimal, as the choice of the sound font has an impact on the obtained results, which is likely to be most apparent in genre and mood features. Future research could focus on improving accuracy related to these features. In sum, we are confident that our **MidiCaps** dataset will facilitate the development of the first Text-to-MIDI generation models.

5. CONCLUSION

We present the first large-scale open MIDI captioned dataset, **MidiCaps**. This dataset also includes a comprehensive set of musical features such as chord patterns, genre, and mood. To facilitate the development of this dataset, we have developed a MIDI captioning framework. This approach includes music feature extraction and summarization from MIDI and the synthesized audio, as well as the use of the Claude-3 LLM to generate the final captions using in-context learning. To evaluate the final dataset, we have conducted two subjective listening studies, which confirm that the captions are natural and indeed contain a text description of the musical features contained in the accompanying MIDI file. The resulting new **MidiCaps** dataset contains 168,407 MIDI files with descriptive text captions and is available online⁵ under a Creative Commons licence.

⁵huggingface.co/datasets/amaai-lab/MidiCaps

6. ACKNOWLEDGMENTS

This project has received funding from the SUTD Kick-starter Initiative no. SKI 2021_04_06.

7. REFERENCES

- [1] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar *et al.*, “Llama: Open and efficient foundation language models,” *arXiv preprint arXiv:2302.13971*, 2023.
- [2] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, “Zero-shot text-to-image generation,” in *International conference on machine learning*. Pmlr, 2021, pp. 8821–8831.
- [3] J. Melechovsky, Z. Guo, D. Ghosal, N. Majumder, D. Herremans, and S. Poria, “Mustango: Toward controllable text-to-music generation,” *Proc. of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2024.
- [4] M. Schedl, E. Gómez, J. Urbano *et al.*, “Music information retrieval: Recent developments and applications,” *Foundations and Trends® in Information Retrieval*, vol. 8, no. 2-3, pp. 127–261, 2014.
- [5] D. Herremans, C.-H. Chuan, and E. Chew, “A functional taxonomy of music generation systems,” *ACM Computing Surveys (CSUR)*, vol. 50, no. 5, pp. 1–30, 2017.
- [6] A. Agostinelli, T. I. Denk, Z. Borsos, J. Engel, M. Verzetti, A. Caillon, Q. Huang, A. Jansen, A. Roberts, M. Tagliasacchi *et al.*, “Musiclm: Generating music from text,” *arXiv preprint arXiv:2301.11325*, 2023.
- [7] Q. Huang, D. S. Park, T. Wang, T. I. Denk, A. Ly, N. Chen, Z. Zhang, Z. Zhang, J. Yu, C. Frank *et al.*, “Noise2music: Text-conditioned music generation with diffusion models,” *arXiv preprint arXiv:2302.03917*, 2023.
- [8] H. Liu, Q. Tian, Y. Yuan, X. Liu, X. Mei, Q. Kong, Y. Wang, W. Wang, Y. Wang, and M. D. Plumbley, “Audioldm 2: Learning holistic audio generation with self-supervised pretraining,” *arXiv preprint arXiv:2308.05734*, 2023.
- [9] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Défossez, “Simple and controllable music generation,” *arXiv preprint arXiv:2306.05284*, 2023.
- [10] S. Doh, K. Choi, J. Lee, and J. Nam, “Lp-musiccaps: Llm-based pseudo music captioning,” *arXiv preprint arXiv:2307.16372*, 2023.
- [11] C. Raffel, *Learning-based methods for comparing sequences, with applications to audio-to-midi alignment and matching*. Columbia University, 2016.
- [12] J. Wei, Y. Tay, R. Bommasani, C. Raffel, B. Zoph, S. Borgeaud, D. Yogatama, M. Bosma, D. Zhou, D. Metzler *et al.*, “Emergent abilities of large language models,” *arXiv preprint arXiv:2206.07682*, 2022.
- [13] C. Hawthorne, A. Stasyuk, A. Roberts, I. Simon, C.-Z. A. Huang, S. Dieleman, E. Elsen, J. Engel, and D. Eck, “Enabling factorized piano music modeling and generation with the maestro dataset,” *arXiv preprint arXiv:1810.12247*, 2018.
- [14] H.-W. Dong, W.-Y. Hsiao, L.-C. Yang, and Y.-H. Yang, “Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.
- [15] A. Ycart, E. Benetos *et al.*, “A-maps: Augmented maps dataset with rhythm and key annotations,” 2018.
- [16] F. Simonetta, F. Carnovalini, N. Orio, and A. Rodà, “Symbolic music similarity through a graph-based representation,” in *Proceedings of the Audio Mostly 2018 on Sound in Immersion and Emotion*, 2018, pp. 1–7.
- [17] S. Wu, D. Yu, X. Tan, and M. Sun, “Clamp: Contrastive language-music pre-training for cross-modal symbolic music information retrieval,” *Proc. of ISMIR*, 2023.
- [18] I. Manco, E. Benetos, E. Quinton, and G. Fazekas, “Muscaps: Generating captions for music audio,” in *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2021, pp. 1–8.
- [19] Q. Huang, A. Jansen, J. Lee, R. Ganti, J. Y. Li, and D. P. W. Ellis, “Mulan: A joint embedding of music audio and natural language,” in *Proceedings of the 23rd International Society for Music Information Retrieval Conference, ISMIR 2022, Bengaluru, India, December 4-8, 2022*, P. Rao, H. A. Murthy, A. Srinivasamurthy, R. M. Bittner, R. C. Repetto, M. Goto, X. Serra, and M. Miron, Eds., 2022, pp. 559–566. [Online]. Available: <https://archives.ismir.net/ismir2022/paper/000067.pdf>
- [20] Z. Kuang, S. Zong, J. Zhang, J. Chen, and H. Liu, “Music-to-text synaesthesia: Generating descriptive text from music recordings,” 2022.
- [21] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray *et al.*, “Training language models to follow instructions with human feedback,” *Advances in neural information processing systems*, vol. 35, pp. 27 730–27 744, 2022.
- [22] OpenAI, “Introducing ChatGPT,” 2023. [Online]. Available: <https://openai.com/blog/chatgpt>
- [23] M. contributors, “Mido: MIDI objects for python,” 2024. [Online]. Available: <https://github.com/mido/mido>

- [24] M. S. Cuthbert and C. Ariza, “music21: A toolkit for computer-aided musicology and symbolic music data,” 2010.
- [25] B. Zamecnik, “midi2audio.” [Online]. Available: <https://github.com/bzamecnik/midi2audio>
- [26] J. Newmarch and J. Newmarch, “Fluidsynth,” *Linux Sound Programming*, pp. 351–353, 2017.
- [27] FluidSynth Contributors, “FluidSynth: A real-time software synthesizer,” 2024. [Online]. Available: <https://github.com/FluidSynth/fluidsynth>
- [28] D. Bogdanov, N. Wack, E. Gómez Gutiérrez, S. Gulati, H. Boyer, O. Mayor, G. Roma Trepát, J. Salamon, J. R. Zapata González, X. Serra *et al.*, “Essentia: An audio analysis library for music information retrieval,” in *Britto A, Gouyon F, Dixon S, editors. 14th Conference of the International Society for Music Information Retrieval (ISMIR); 2013 Nov 4-8; Curitiba, Brazil.[place unknown]: ISMIR; 2013. p. 493-8. International Society for Music Information Retrieval (ISMIR), 2013.*
- [29] M. Mauch and S. Dixon, “Approximate note transcription for the improved identification of difficult chords,” in *Proceedings of the 11th International Society for Music Information Retrieval Conference, ISMIR 2010, Utrecht, Netherlands, August 9-13, 2010*, J. S. Downie and R. C. Veltkamp, Eds. International Society for Music Information Retrieval, 2010, pp. 135–140. [Online]. Available: <http://ismir2010.ismir.net/proceedings/ismir2010-25.pdf>
- [30] D. Kevian, U. Syed, X. Guo, A. Havens, G. Dullerud, P. Seiler, L. Qin, and B. Hu, “Capabilities of large language models in control engineering: A benchmark study on gpt-4, claude 3 opus, and gemini 1.0 ultra,” *arXiv preprint arXiv:2404.03647*, 2024.
- [31] Anthropic, “Claude 3 opus,” 2024. [Online]. Available: <https://www.anthropic.com/claude>
- [32] G. Stoet, “Psytoolkit: A software package for programming psychological experiments using linux,” *Behavior research methods*, vol. 42, pp. 1096–1104, 2010.
- [33] —, “Psytoolkit: A novel web-based method for running online questionnaires and reaction-time experiments,” *Teaching of Psychology*, vol. 44, no. 1, pp. 24–31, 2017.

A NEW DATASET, NOTATION SOFTWARE, AND REPRESENTATION FOR COMPUTATIONAL SCHENKERIAN ANALYSIS

Stephen Ni-Hahn¹ Weihan Xu¹ Jerry Yin¹ Rico Zhu¹
Simon Mak¹ Yue Jiang¹ Cynthia Rudin¹

¹Duke University, USA

stephen.hahn@duke.edu

ABSTRACT

Schenkerian Analysis (SchA) is a uniquely expressive method of music analysis, combining elements of melody, harmony, counterpoint, and form to describe the hierarchical structure supporting a work of music. However, despite its powerful analytical utility and potential to improve music understanding and generation, SchA has rarely been utilized by the computer music community. This is in large part due to the paucity of available high-quality data in a computer-readable format. With a larger corpus of Schenkerian data, it may be possible to infuse machine learning models with a deeper understanding of musical structure, thus leading to more “human” results. To encourage further research in Schenkerian analysis and its potential benefits for music informatics and generation, this paper presents three main contributions: 1) a new and growing dataset of SchAs, the largest in human- and computer-readable formats to date (>140 excerpts), 2) a novel software for visualization and collection of SchA data, and 3) a novel, flexible representation of SchA as a heterogeneous-edge graph data structure.

1. INTRODUCTION

With the continuously growing availability of “big data,” machine learning models and algorithms have made enormous strides in many fields, such as computer vision and language modeling. Recent approaches to music information retrieval (MIR) and music generation tasks are increasingly fueled by massive datasets as well, particularly when working with raw audio. For instance, for generation tasks, Meta’s *MusicGen* is trained on approximately 20,000 hours of licensed music [1], OpenAI’s *Jukebox* on 1.2 million songs [2], and Google’s *Noise2Music* on 340,000 hours of music [3]. Castellon et al. show how these large generation models produce useful representations for downstream MIR tasks [4]. Won et al. perform

multimodal metric learning for tag-based music retrieval using approximately 158,000 tracks [5].

Despite this promising body of work, many areas of music research do not have access to such data and are therefore under-researched and underappreciated, particularly in the realm of symbolic music or Schenkerian Analysis (SchA). By infusing an understanding of Schenkerian musical structure, generative machine learning models may be able to learn more artistic, theoretically-informed structural features beyond simple form and metric features when making inference. Unfortunately, there is currently only one sizeable publicly available dataset for SchA in computer-readable format, and it is relatively small with 41 excerpts [6].

Schenkerian analysis provides a powerful, flexible, and broadly-used analytical framework for understanding musical melodic-harmonic structure in a sensitive, “human” way. Rather than viewing a piece of music as a series of vertical chunks or horizontal melodies, SchA instead analyzes music as an artistic “unfolding” of harmony through time, taking into account elements of melody, harmony, form, and counterpoint. Schenker’s theories have inspired numerous performers and composers [7–9], helping them to understand their own interpretations of musical structure, which in turn may inform their own performance and/or composition. An understanding of Schenkerian structure helps performers determine what notes deserve emphasis and which may be more transient. A composer can learn to imitate and develop structures they have seen in other pieces of music they admire.

Because Schenkerian theory requires a significant amount of background knowledge in music theory and practice and has a difficult learning curve, it is often overlooked or misunderstood. For instance, SchA is often deemed too narrow due to its origins in repertoire of Western common practice tonal music. However, aspects of Schenkerian theory have shown strong analytical power in works of popular, rock, jazz, and even African folk music, Chinese opera, and 20th century western atonal music [10–13]. To be clear, we see SchA as a broad and evolving field with various analytical tools that can be applied to a wide array of musical genres, not as a static theory solely designed for common practice tonality.

It is our belief that research in computational SchA can enable performers and composers to more easily analyze music and guide the process of understanding mu-



© S. Ni-Hahn, W. Xu, J. Yin, R. Zhu, S. Mak, Y. Jiang, and C. Rudin. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** S. Ni-Hahn, W. Xu, J. Yin, R. Zhu, S. Mak, Y. Jiang, and C. Rudin, “A New Dataset, Notation Software, and Representation for Computational Schenkerian Analysis”, in *Proc. of the 25th Int. Society for Music Information Retrieval Conf.*, San Francisco, United States, 2024.

sical structure. Computational SchA can also aid the expert human analyst by offering several potential solutions, speeding up their ability to parse through a piece of music or potentially unveiling unusual and interesting analytical insights. The computer would not replace the human expert; rather, it would help the analyst find reasonable solutions more quickly, which would be immensely helpful when conducting large-scale corpus studies. Furthermore, inclusion of SchA in MIR and generation tasks may greatly improve results. This injection of computational models with musical theory and/or structure has shown benefits in numerous MIR and generation tasks [14–17].

This paper introduces three main contributions. The first is a new and growing **SchA dataset**, the largest in human- and computer-readable format to date (>140 excerpts). Second, we present a novel **notation software for SchA** in an effort to ease data collection and visualization. Lastly, we describe a **representation of SchA** as a graphical data structure and graph pooling problem.

The following subsections describe SchA in more detail, as well as the relevant history of computational SchA. Section 2 describes our novel dataset and data collection tool. Finally, Section 3 describes how SchA may be represented as a graph data structure.

1.1 Hierarchical Music Analysis

Music is often composed and understood in terms of hierarchical structures such as phrase and rhythmic structure [18, 19], form structure [20, 21], and linear/harmonic structure [22, 23]. In this paper, we focus on the Schenkerian model of harmonic-melodic structure. As mentioned earlier, SchA aims to reveal how harmonies are “unfolded” through linear motion on various levels of structure. Figure 1 shows the relationship between a fugue’s subject melody and its underlying harmony, as well as the hierarchy of melodic tones.

Figure 1: The primary author’s analysis of J.S. Bach’s F major fugue subject from *Das Wohltemperierte Klavier I*.

The annotated score on the upper line shows how notes relate on various levels of structure, forming two theoretical outer voices. Longer stems indicate deeper levels of structure. The reduction on the bottom line exemplifies the underlying harmony that is unfolded by the subject melody. Green-stemmed notes correspond with the deep outer voices of the reduction.

SchA has shown that similar harmonic and motivic features often exist on multiple levels of hierarchy, revealing music’s “fractal” nature [24]. For instance, in Figure 1, the foreground melody within the first full measure (D4-C4-Bb3) can be seen as a *parallelism* of the deep level melody spanning the entire excerpt (C4-Bb3-A3); the two melodies have a similar motivic descending third in step-wise motion. One can also see the first full measure leading into the second measure as a $V^b - I$ motion in the key of V, paralleling the deep level $V^b - I$ shown in the reduction. While these examples are on a very small scale, one can see more complex harmonic and motivic structures unfolded through entire pieces. For instance, see Example 12 in [24] describing Schubert’s *Erlkönig* or Example 2 in [25] describing The Beatles’ *Something*.

Because these same music-theoretical ideas and motifs permeate multiple levels of structure, the use of a carefully-designed machine learning model may reveal such structure in a layered approach. With the rise of machine learning in data science, this calls into importance the need for computer-readable SchA datasets for model training.

1.2 Previous Work and Data for Computational Schenkerian Analysis

The majority of past attempts at computational SchA [26–30] were based on heuristics and rule-based algorithms, and therefore did not require a true computer-readable dataset for SchA. To our knowledge, Marsden [31] was the first to venture towards a machine learning approach, using a humble corpus of six Mozart analyses. He developed a “goodness metric” based on these six analyses to find the best candidate analyses within a massive search space.

More recently, Kirilin designed a probabilistic model for SchA that understands SchAs as maximal outerplanar graphs (MOPs) and learns how likely certain notes *prolong* others using random forests [32, 33]. He defines prolongation as “a situation where an analyst determines that a group of notes is elaborating a group of more structurally fundamental notes.” For instance, the syntax follows the pattern $X(Y)Z$, where the note(s) of Y prolong the motion from note X to note Z .

One potential drawback of Kirilin’s model is that it always considers one musical voice as one theoretical voice. Looking back at Figure 1, for example, we see there is clearly a deep level bass motion from E3 to F3, supporting the upper voice, which follows the motion C4-Bb3-A3. The sixteenth notes of m. 2 act to fill the gap between the lower and upper theoretical voices. An MOP reduction of the melody would force all notes to be understood as a single theoretical voice, thus obscuring the underlying counterpoint of the passage. For this reason, we represent SchA as a more general graph data structure, described further in Section 3.

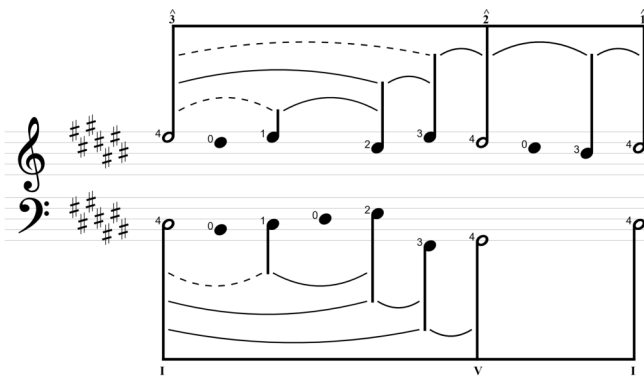
For his model, Kirilin released the first large-scale computer-readable dataset of symbolic music with corresponding expert Schenkerian analyses, *Schenker41* [6]. This collection consists of 41 excerpts from the common practice period of European art music, with analyses from

```

trebleNotes: {
  pitchNames: [n.E5, n.D5, n.E5, n._, n.C5, n.E5, n.D5, n.C5, n.B4, n.C5],
  depths: [4, 0, 1, 0, 2, 3, 4, 0, 3, 4],
  selected: [],
  ursatz: [0, 6, 9],
  scaleDegree: [3, 0, 0, 0, 0, 0, 2, 0, 0, 1],
  flagged: [],
  sharps: [],
  flats: [],
  naturals: [],
  parenthetical: []
},
bassNotes: {
  pitchNames: [n.C3, n.B2, n.C3, n.D3, n.E3, n.F2, n.G2, n._, n._, n.C3],
  depths: [4, 0, 1, 0, 2, 3, 4, 0, 0, 4],
  selected: [],
  ursatz: [0, 6, 9],
  scaleDegree: [1, 0, 0, 0, 0, 0, 5, 0, 0, 1],
  flagged: [],
  sharps: [],
  flats: [],
  naturals: [],
  parenthetical: []
},

```

(a) JSON representation.



(b) Graphical representation.

Figure 2: Screenshots of a toy Schenkerian analysis in JSON and graphical form as generated by our notation software.

three textbooks [23, 34–36] and an independent, anonymous expert in Schenkerian analysis. Kirilin also created the first computer-readable format for Schenkerian analysis, which describes all prolongations present in an analysis. The text-based format can also encode linear progressions, omitted repetitions, and harmonic context.

The Schenker41 dataset is an important first step towards broader musical-hierarchical research in the MIR community; however, there are some limitations. First of all, the quality of the excerpts chosen are questionable. Kirilin and Jensen recruited three expert Schenkerian analysts to grade textbook analyses as well as their machine learning model’s analyses in their 2015 paper (see Figure 8 in [37]). One would expect the textbook analyses to receive grades of “A-” or greater, allowing wiggle room for differences of opinion. However, many excerpts score lower marks; some were even given failing grades. Given the high proportion of dubious quality “ground truth” data, it is necessary to produce a greater quantity of quality data before successful, generalizable models can be trained.

There are also several Schenkerian symbols and concepts that are not currently represented in the text-based notation. For instance, unfoldings, voice exchanges, and other hierarchical harmonic function information are ignored. Concerning larger pieces, it is vital to understand the harmonic structure in several layers; an F major triad may stand as a local tonic “I” harmony in the foreground that serves to expand a deeper subdominant “IV” of the background, global key of C. Furthermore, more abstract concepts, such as motivic parallelisms, implied tones, and written commentary are eschewed for the sake of simplicity.

2. DATASET AND NOTATION SOFTWARE FOR SCHENKERIAN ANALYSIS

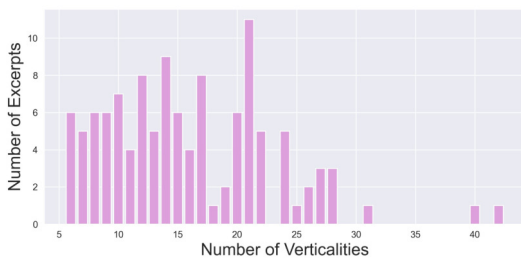
We thus introduce a new large-scale dataset of Schenkerian analyses in human- and computer-readable formats. As of the writing of this paper, the dataset contains 145 analyses from four analysts for a broad range of composers including J.S. Bach, Mendelssohn, Brahms, Bartók, Shostakovich, Gentle Giant, and more. The majority of

analyses were created by the first author (Stephen Ni-Hahn) with consultation from one of the other analysts, who wishes to remain anonymous at this time. Ni-Hahn currently has nearly a decade of experience with SchA including a graduate degree in music theory. The other three analysts are veteran Schenkerian scholars with several decades of experience in the field. The dataset is not static and aims to grow over time. Please contact Stephen Ni-Hahn (stephen.hahn@duke.edu) for questions regarding, and access to the dataset and notation software described in this paper.

Currently, the vast majority of analyses in the dataset describe the hierarchical relationships within fugue subjects by Bach and Pachelbel. Fugue subjects are ideal for preliminary trials with machine learning models since subjects are generally brief, consist of a single instrumental line (which may consist of multiple theoretical voices), generally have clear functional relationships, and each have a definite sense of closure by their end.

Rather than writing out each prolongation explicitly, we produce prolongations as a by-product when assigning a hierarchical *depth* to each note. For example, Figure 2 shows a toy example of an analysis in which the numbers to the left of the note heads indicate depth. Higher depth indicates deeper structure. To retrieve the prolongations, we simply traverse the graph at each depth level (greater than 0), connecting consecutive notes that are at the same level or higher. Custom prolongations that do not occur within this system may be added in a similar fashion to Kirilin’s text format by describing the voice and index of the start, middle, and end notes.

Figures 3 and 4 present simple statistics about our dataset. Figure 3a shows the distribution of excerpt lengths in terms of *verticalities*. A verticality is defined as a point in time where one or both of a treble and bass note exist. Note that this does not measure length of time or number of measures; rather, the number of verticalities describes the number of potentially unique depths in an excerpt. Figure 4 shows the distribution of intervals between consecutive notes in the treble and bass voices at various depths. We see that as depth increases, the distribution of treble in-



(a) Distribution of excerpt length.

Depth	Notes per Depth		Avg. Verticalities per Depth		Num Pieces
	Inclusive	Literal	Inclusive	Literal	Max Depth
0	1815	646	15.6	4.7	0
1	1280	324	11.0	2.8	1
2	957	265	8.3	2.3	8
3	694	258	6.5	2.4	26
4	439	232	5.4	2.9	36
5	208	161	4.6	3.5	33
6	51	41	4.3	3.4	10
7	10	6	5.0	3.0	1

(b) Dataset statistics regarding note depth.

Figure 3: Dataset statistics. Verticality is defined as a point in time where one or both of a treble and bass note exist. “Inclusive” includes notes of higher depth when counting notes of lower depths. “Literal” counts the note depths as they are defined. The final column describes the distribution of max depths over all excerpts. See Section 2 for more details.

Intervals moves from smaller to larger intervals, while bass intervals increasingly concentrate around 0 and 5. These statistics suggest that surface level treble motions in our dataset are mostly stepwise and span larger intervals at deeper levels of structure. Furthermore, deep bass structures tend to hold steady and support the upper voice or move along the circle of fifths by jumping 5 or 7 half steps. Table 3b describes various statistics regarding the notes and depths of our dataset. Columns labeled “inclusive” mean that notes of higher depth are included when counting notes of lower depths. For instance, a depth 4 note is counted in the number of depth 0 notes, while the depth 4 note would not count towards the number of depth 5 notes. The “literal” label counts the note depths as they are defined. The final column describes the distribution of max depths over all excerpts.

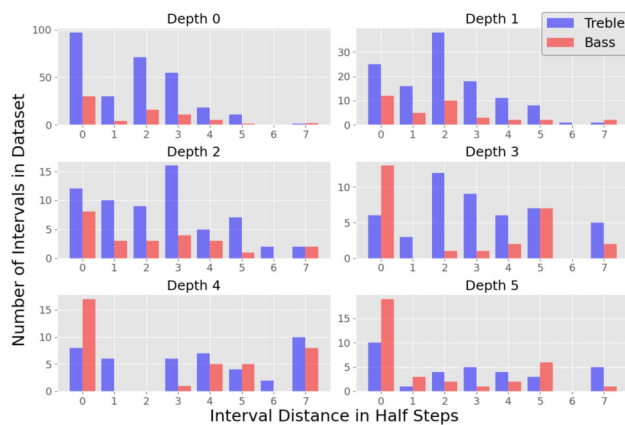


Figure 4: Distribution of intervals between consecutive notes at each depth.

2.1 Data Collection Tool

To facilitate easy collection and visualization of Schenkerian data, we introduce a new computer notation system for Schenkerian analyses (see Figure 2 for a screenshot).

As of the writing of this paper, our software is capable of notating up to four voice structures of any length. Simple commands allow the user to adjust the pitches, note depths, harmonic/scale-degree label, notes considered part

of the *Ursatz*, etc. Slurs and beams of the outer theoretical voices are automatically generated based on the depths of the notes. We are currently working on ways to render custom markings, such as voice exchanges, unfoldings, and linear progression beams.

Behind the scenes, the Schenkerian analysis is a simple standardized object in JavaScript Object Notation (JSON), which is *highly generalizable, lightweight, and simple to parse*, and is capable of describing any obscurities within a particular analysis. Our JSON object contains metadata about the analysis, key information, and information on each of four theoretical voices. Metadata describes the analyst, composer, title, subtitle, and any associated written description of the analysis. Furthermore, each theoretical voice is encoded as a list of pitch names, depths, *Ursatz* indices, scale degree/Roman numerals, flagged note indices, sharp/flat/natural indices, and parenthetical indices. Additionally, the JSON object stores “cross voice” symbols such as voice exchange lines and lines indicating related tones across larger spans of time.

Note that it is straightforward to translate between Kirlin’s OPC text notation and our JSON notation. To translate from text to JSON, the notes can be parsed from the musicxml and placed in their appropriate voice. Then note depths may be determined by the location and relative length of their prolongation. Translating from JSON to text is simpler, as one can traverse each depth and retrieve the prolongations.

The software is constructed using languages Javascript/Typescript and the Vue web framework. It is packaged using Electron Forge. Software access can be requested by emailing the first author.

3. SCHENKERIAN ANALYSIS AS A HETEROGENEOUS GRAPH DATA STRUCTURE

As mentioned in Section 1.2, Kirlin’s model simplifies the difficult problem of performing SchA, using a limited version of Yust’s MOP representation for SchA. With a greater amount of data, less compromising representations may be used for modeling. The following section describes how a musical score may be represented as a

algo2e 1 JSON to Clusters

Definitions

$parts \leftarrow \{sop, alto, ten, bass\}$
 $n_v \leftarrow$ the number of verticalities v (indexed by i) in an analysis
 $p_i \leftarrow$ note of part $p \in parts$ within v_i
 $d_i^{(p)} \leftarrow$ depth of note p_i
 $\forall p \in parts, \text{len}(p) = \text{len}(d^{(p)}) = n_v.$

Procedure CLUSTER(p, i)

```

if  $\exists j < i$  s.t.  $d_j^p > 0$  then
     $j \leftarrow \underset{j}{\text{argmin}} |i - j|$  s.t.  $j < i$  and  $d_j^{(p)} > 0$ 
    return  $\{(p, j)\}$  // Note in the same voice to the left
else if  $\exists j > i$  s.t.  $d_j^p > 0$  then
     $j \leftarrow \underset{j}{\text{argmin}} |i - j|$  s.t.  $j > i$  and  $d_j^{(p)} > 0$ 
    return  $\{(p, j)\}$  // Note in the same voice to the right
else
     $j_1, j_2 \leftarrow \underset{j_1, j_2}{\text{argmin}} \min(|i - j_1|, |i - j_2|)$  s.t.  $(i - j_1) \cdot (i - j_2) \leq 0$  and  $d_{j_1}^{(sop)} > 0$  and  $d_{j_2}^{(bass)} > 0$ 
    return  $\{(sop, j_1), (bass, j_2)\}$  // Closest two notes in outer voices in opposite directions to the inner voice note
end if
    
```

heterogeneous-edge directed graph data structure and how SchA may be conceptualized as a graph clustering problem.

3.1 Graph Music Representation

In what follows, we represent music as a heterogeneous directed graph G , where each node describes a note, and various types of edges describe the relationships between notes. Concretely, G is represented as (\mathbb{A}, X) , where $\mathbb{A} \in \{0, 1\}^{h \times n \times n}$ describes the set of h adjacency matrices (one for each edge type) over n nodes, and $X \in \mathbb{R}^{n \times d}$ is the node feature matrix with d as the number of features. These d features may be learned by a neural network, for instance, to correspond with categorical and numerical musical features.

We adapt the encoding scheme proposed by Jeong et al. [38] for the purpose of Schenkerian analysis. Nodes may be encoded with any musical feature present in the score, such as pitch class, octave, absolute duration, position (absolute or relative), metric strength, etc. We suggest the use of five main edge types: (i) forward edges connect two consecutive notes within a voice, (ii) onset edges connect notes that begin at the same time, (iii) sustain edges connect notes that are played while the source note is held, (iv) rest edges are like forward edges, but imply a rest occurs between the two related notes, and (v) linear edges connect each note with the next notes that occur at specific intervals from the source.

3.2 Schenkerian Analysis as Hierarchical Clustering

With this graphical representation of music, the process of Schenkerian analysis may then be posed as a hierarchical graph clustering problem. Figure 5 presents a toy example of how Schenkerian analysis may be represented as a series

of hierarchical clusters. The clustering between two subsequent levels of Schenkerian analysis is expressed through a *clustering matrix*, $S^{(l)} \in \mathbb{R}^{n_l \times n_{l+1}}$, where n_l is the number of nodes in *clustering* layer l and $n_{l+1} < n_l$ is the number of nodes after one iteration of clustering. We define n_0 to be the total number of notes in the music.

Note that we can understand a clustering between *any* two layers as a single matrix, denoted as $S^{(l_i) \rightarrow (l_j)} \in \mathbb{R}^{n_{l_i} \times n_{l_j}}$; $i < j$, where i and j are the index of the source and destination layers respectively. This single matrix is obtained by simply multiplying all sequential clustering matrices. For example, in Figure 5, to retrieve the matrix describing how all five nodes of the original score are clustered into the two nodes of the final middleground layer, we can multiply each clustering matrix together:

$$S^{(0) \rightarrow (2)} = S^{(0)} \cdot S^{(1)} \cdot S^{(2)} = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}^\top.$$

3.3 Converting Schenkerian Analyses from JSON to Matrix Notation

Schenkerian analysis JSON data (collected using our tool described in Section 2.1) requires extra processing to be represented as hierarchical clusters. Here, we provide an algorithm to convert our JSON data into a series of progressively smaller clustering matrices (see Algorithm 1).

Essentially, we first traverse the outer voices of the JSON file, clustering notes of depth 0 into the closest note of higher depth to the left in the same voice. If that note does not exist, it defaults to the closest note of a higher depth to the right. For inner voices, if they do not describe hierarchical depth (all 0 depth), they are clustered 50%-50% between the nearest bass and soprano below and above or left to right, in that order. If the inner voice has

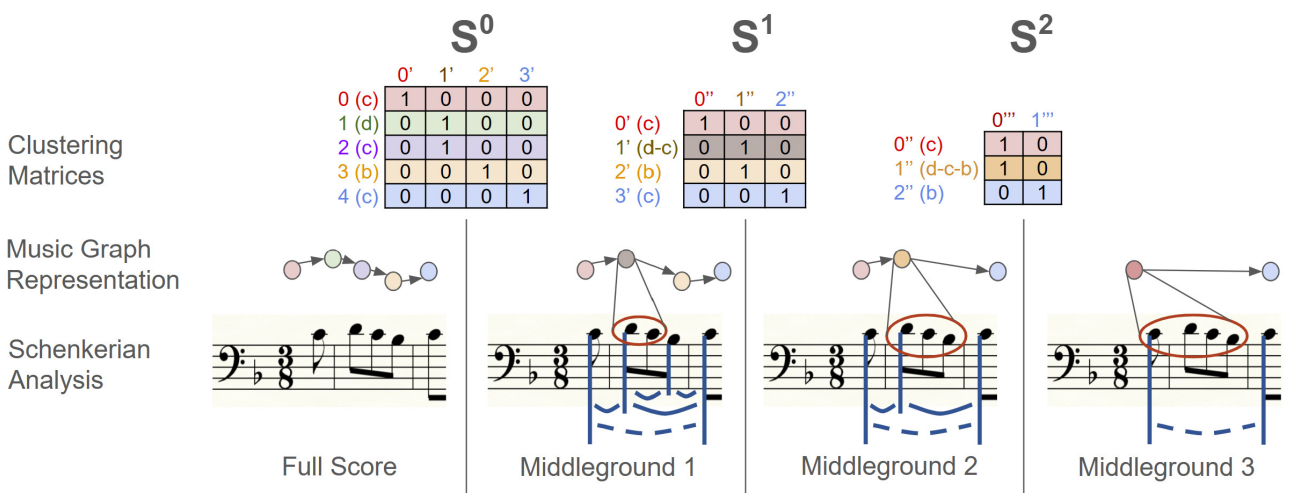


Figure 5: Visualization of Schenkerian analysis as a series of clustering matrices. The bottom row shows a simple score with Schenkerian annotation moving from all notes in the score to more abstracted versions of the score from left to right. The middle row visualizes the music as a graph. The top row shows the ground truth cluster matrices that relate one layer to the next; rows describe nodes before clustering, while columns describe nodes after clustering.

specified depth, it is treated similarly to the outer voices. All depths are then decremented and the process begins again for the next clustering matrix.

3.4 Implications of SchA Graph Clustering

The above formulation of SchA as a graph clustering problem facilitates more generalizable analysis. Whereas Kirilin’s MOP-based model focuses on a single melody as one theoretical voice, a fuller graph representation allows for greater flexibility via any number of theoretical voices. There are, however, several drawbacks with this new approach. Because the clustering works with the notes of the score, it is unclear how to handle cases where multiple theoretical voices converge on a single note. This issue may also be present when handling inner voices of unspecified depth. In our algorithm, we suggest splitting unspecified inner voices 50%-50% between the outer voices, but other approaches may also be reasonable.

Another advantage that the proposed graph clustering representation has over the MOP representation is its ability to cluster multiple notes into one in a single layer. This is particularly common when there are several repeated notes. In a MOP, repeated notes must be given detailed hierarchy, whereas a human expert would generally think of such repetitions as structurally redundant. There are also instances of prolongations that span more than one child, where having only one child would not properly reflect the music. For instance, if the melody over a C major tonic triad (CEG) quickly plays out the upper tetrachord of the scale, G-A-B-C, then the A and B are structurally equal; they both bridge the gap from G to C. On the other hand, allowing multiple children for every prolongation makes the search space for potential solutions orders of magnitude larger.

As the amount of labeled SchA data grows and computational power improves, there is great potential for learning complex relationships via machine learning that may

be unattainable in previous analyses. Deep learning has enjoyed considerable success on analyzing the Bach chorale dataset [39–41], thus we are optimistic that SchA can also be learned for broad datasets from different genres. The proposed dataset, notation software and graph representation provides a promising step towards this goal.

4. CONCLUSION

In this paper, we introduce the largest corpus of Schenkerian analyses in computer-readable format to date. This was largely made possible using our novel SchA notation software, which is natural, interpretable, and enables easy data collection and visualization. Finally, we describe and discuss a novel representation for SchA as a graph clustering problem that allows representation of any possible Schenkerian analysis, avoiding the limitations of MOPs. It is our hope that the growing amount of data and ease of its collection will enable broader research into SchA’s applications.

5. REFERENCES

- [1] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Défossez, “Simple and controllable music generation,” *Advances in Neural Information Processing Systems*, vol. 36, 2024.
- [2] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, and I. Sutskever, “Jukebox: A generative model for music,” *arXiv preprint arXiv:2005.00341*, 2020.
- [3] Q. Huang, D. S. Park, T. Wang, T. I. Denk, A. Ly, N. Chen, Z. Zhang, Z. Zhang, J. Yu, C. Frank *et al.*, “Noise2music: Text-conditioned music generation with diffusion models,” *arXiv preprint arXiv:2302.03917*, 2023.

- [4] R. Castellon, C. Donahue, and P. Liang, “Codified audio language modeling learns useful representations for music information retrieval,” *arXiv preprint arXiv:2107.05677*, 2021.
- [5] M. Won, S. Oramas, O. Nieto, F. Gouyon, and X. Serra, “Multimodal metric learning for tag-based music retrieval,” in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 591–595.
- [6] P. B. Kirilin, “A data set for computational studies of schenkerian analysis,” in *ISMIR*, 2014, pp. 213–218.
- [7] B. Finane, “The humanist - murray perahia - steinway & sons.” [Online]. Available: <https://www.steinway.com/news/features/the-humanist-murray-perahia>
- [8] H. Schenker, *The art of performance*. Oxford University Press, 2000.
- [9] T. L. Jackson, “Heinrich schenker as composition teacher: The schenker-oppel exchange,” *Music Analysis*, vol. 20, no. 1, pp. 1–115, 2001.
- [10] D. F. Nobile, “A structural approach to the analysis of rock music,” Ph.D. dissertation, 2014, copyright - Database copyright ProQuest LLC; ProQuest does not claim copyright in the individual underlying works; Last updated - 2023-03-03. [Online]. Available: <https://login.proxy.lib.duke.edu/login?url=https://www.proquest.com/dissertations-theses/structural-approach-analysis-rock-music/docview/1506971540/se-2>
- [11] J. Stock, “The application of schenkerian analysis to ethnomusicology: problems and possibilities,” *Music Analysis*, vol. 12, no. 2, pp. 215–240, 1993.
- [12] S. Larson, *Analyzing Jazz: A Schenkerian Approach*, ser. ACLS Humanities E-Book. Pendragon, 2009. [Online]. Available: <https://books.google.com/books?id=CmMJAQAAMAAJ>
- [13] A. Didier, “Form and tonal spectrum in 12-tone music: Approaches to analysis in schoenberg, walker, and webern,” Ph.D. dissertation, University of Oregon, 2022.
- [14] H. Fong, V. Kumar, and K. Sudhir, “A theory-based interpretable deep learning architecture for music emotion,” *Available at SSRN 4025386*, 2023.
- [15] J. Wu, C. Hu, Y. Wang, X. Hu, and J. Zhu, “A hierarchical recurrent neural network for symbolic melody generation,” *IEEE transactions on cybernetics*, vol. 50, no. 6, pp. 2749–2757, 2019.
- [16] S. Hahn, R. Zhu, S. Mak, C. Rudin, and Y. Jiang, “An interpretable, flexible, and interactive probabilistic framework for melody generation,” in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023, pp. 4089–4099.
- [17] X. Zhang, J. Zhang, Y. Qiu, L. Wang, and J. Zhou, “Structure-enhanced pop music generation via harmony-aware learning,” in *Proceedings of the 30th ACM International Conference on Multimedia*, 2022, pp. 1204–1213.
- [18] W. N. Rothstein, “Phrase rhythm in tonal music,” (*No Title*), 1989.
- [19] F. Lerdahl and R. S. Jackendoff, *A Generative Theory of Tonal Music, reissue, with a new preface*. MIT press, 1996.
- [20] J. Hepokoski and W. Darcy, *Elements of sonata theory: Norms, types, and deformations in the late-eighteenth-century sonata*. Oxford University Press, 2006.
- [21] W. E. Caplin, *Analyzing classical form: An approach for the classroom*. Oxford University Press, USA, 2013.
- [22] H. Schenker, *Free Composition: Volume III of new musical theories and fantasies*. Pendragon Press, 2001, vol. 1.
- [23] A. C. Cadwallader, D. Gagné, and F. Samarotto, “Analysis of tonal music: a schenkerian approach,” (*No Title*), 1998.
- [24] C. Burkhardt, “Schenker’s” motivic parallelisms,” *Journal of Music Theory*, vol. 22, no. 2, pp. 145–175, 1978.
- [25] R. Gauldin, “Beethoven, tristan, and the beatles,” in *College Music Symposium*, vol. 30, no. 1. JSTOR, 1990, pp. 142–152.
- [26] M. Kassler, *Proving musical theorems I: The middleground of Heinrich Schenker’s theory of tonality*. Basser Department of Computer Science, School of Physics, University of Sydney, 1975, no. 103.
- [27] R. E. Frankel, S. J. Rosenschein, and S. W. Smoliar, “A lisp-based system for the study of schenkerian analysis,” *Computers and the Humanities*, pp. 21–32, 1976.
- [28] S. W. Smoliar, “A computer aid for schenkerian analysis,” in *Proceedings of the 1979 annual conference*, 1979, pp. 110–115.
- [29] P. Mavromatis and M. Brown, “Parsing context-free grammars for music: A computational model of schenkerian analysis,” in *Proceedings of the 8th International Conference on Music Perception & Cognition*, 2004, pp. 414–415.
- [30] É. Gilbert and D. Conklin, “A probabilistic context-free grammar for melodic reduction,” in *Proceedings of the International Workshop on Artificial Intelligence and Music, 20th International Joint Conference on Artificial Intelligence*. Citeseer, 2007, pp. 83–94.
- [31] A. Marsden, “Schenkerian analysis by computer: A proof of concept,” *Journal of New Music Research*, vol. 39, no. 3, pp. 269–289, 2010.

- [32] P. B. Kirlin, “A probabilistic model of hierarchical music analysis,” PhD thesis, University of Massachusetts Amherst, Amherst, MA, February 2014, available at <https://www.cs.rhodes.edu/~kirlin/p/diss.html>.
- [33] J. D. Yust, *Formal models of prolongation*. ProQuest, 2006.
- [34] A. Forte and S. E. Gilbert, “Instructor’s manual for introduction to schenkerian analysis,” (*No Title*), 1982.
- [35] A. Forte and S. Gilbert, “Introduction to schenkerian analysis: instructor’s manual,” 1982.
- [36] T. Pankhurst, *SchenkerGUIDE: a brief handbook and website for Schenkerian analysis*. Routledge, 2008.
- [37] P. Kirlin and D. Jensen, “Learning to uncover deep musical structure,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 29, no. 1, 2015.
- [38] D. Jeong, T. Kwon, Y. Kim, and J. Nam, “Graph neural network for music score data and modeling expressive piano performance,” in *International conference on machine learning*. PMLR, 2019, pp. 3060–3070.
- [39] G. Hadjeres, F. Pachet, and F. Nielsen, “Deepbach: a steerable model for bach chorales generation,” in *International conference on machine learning*. PMLR, 2017, pp. 1362–1371.
- [40] C.-Z. A. Huang, A. Vaswani, J. Uszkoreit, N. Shazeer, C. Hawthorne, A. M. Dai, M. D. Hoffman, and D. Eck, “An improved relative self-attention mechanism for transformer with application to music generation,” *arXiv preprint arXiv:1809.04281*, vol. 2, 2018.
- [41] F. T. Liang, M. Gotham, M. Johnson, and J. Shotton, “Automatic stylistic composition of bach chorales with deep lstm.” in *ISMIR*, 2017, pp. 449–456.

6. ETHICS STATEMENT

This particular work has no direct negative ethical implications.

We acknowledge that Heinrich Schenker (the inventor of Schenkerian analysis) was racist and nationalist. His sociopolitical views are not condoned by authors of the present work. As originally designed, his style of analysis did not extend far beyond German composers of the common practice era. As we address in Section 1, we do not see Schenkerian analysis as a static analysis defined by Schenker; rather, we see it as a growing and developing set of tools that may be applied to various musical genres, detached from Schenker’s personal views.

DITTO-2: DISTILLED DIFFUSION INFERENCE-TIME T-OPTIMIZATION FOR MUSIC GENERATION

Zachary Novack¹

Julian McAuley¹

Taylor Berg-Kirkpatrick¹

Nicholas J. Bryan²

¹University of California – San Diego

²Adobe Research

znovack@ucsd.edu, njb@ieee.org

ABSTRACT

Controllable music generation methods are critical for human-centered AI-based music creation, but are currently limited by speed, quality, and control design trade-offs. Diffusion inference-time T-optimization (DITTO), in particular, offers state-of-the-art results, but is over 10x slower than real-time, limiting practical use. We propose **Distilled Diffusion Inference-Time T-Optimization** (or DITTO-2), a new method to speed up inference-time optimization-based control and unlock faster-than-real-time generation for a wide-variety of applications such as music inpainting, outpainting, intensity, melody, and musical structure control. Our method works by (1) distilling a pre-trained diffusion model for fast sampling via an efficient, modified consistency or consistency trajectory distillation process (2) performing inference-time optimization using our distilled model with one-step sampling as an efficient surrogate optimization task and (3) running a final multi-step sampling generation (decoding) using our estimated noise latents for best-quality, fast, controllable generation. Through thorough evaluation, we find our method not only speeds up generation over 10-20x, but simultaneously improves control adherence and generation quality all at once. Furthermore, we apply our approach to a new application of maximizing text adherence (CLAP score) and show we can convert an unconditional diffusion model without text inputs into a model that yields state-of-the-art text control. Sound examples can be found at <https://ditto-music.github.io/ditto2/>.

1. INTRODUCTION

Audio-domain text-to-music (TTM) methods [1–6] have seen rapid development in recent years and show great promise for music creation. Such progress has been made possible through the development of diffusion models [7–9], language models [1, 2], latent representations of audio [10–13] and text-based control [4, 14]. Such control, however, can be limiting for creative human-centered AI

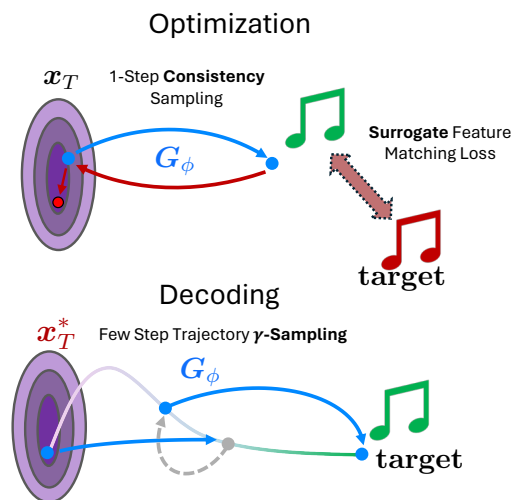


Figure 1: DITTO-2: Distilled Diffusion Inference-Time T-Optimization. We speed up diffusion inference-time optimization-based music generation by 10-20x while improving control and audio quality. (Top) We use diffusion distillation to speed up performance (optimize with 1-step sampling). (Bottom) We then run multi-step sampling for final higher-quality generation (decoding).

music applications, motivating more diverse and advanced control (e.g., melody) that target’s fine-grained aspects of musical composition.

Recent control methods that go beyond text-control fall into training-based and training-free methods. Training-based methods like Music-ControlNet [15] fine-tune DMs with additional adaptor modules that can add time-dependent controls over melody, harmony, and rhythm, offering strong control at the cost of hundreds of GPU hours of fine-tuning for each control. With training-free methods, in particular the class of inference-time *guidance* methods [16, 17], the diffusion sampling process is guided at each step using the gradients of a target control $\nabla_{x_t} \mathcal{L}(\hat{x}_0(x_t))$, where $\hat{x}_0(x_t)$ is a 1-step approximation of the final output. While training-free, the reliance on approximate gradients limits performance [18]. Finally, inference-time optimization (ITO) methods [19, 20] like DITTO [18] offer state-of-the-art (SOTA) control without the need for large-scale fine-tuning via optimizing for noise latents, but suffer from slow inference speeds (10-20x slower than real-time) [18].

In this work, we propose **Distilled Diffusion Inference-Time T-Optimization** (or DITTO-2), a new method for speeding up ITO-based methods by over an order of magni-



© Z. Novack, J. McAuley, T. Berg-Kirkpatrick, and N.J. Bryan. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Z. Novack, J. McAuley, T. Berg-Kirkpatrick, and N.J. Bryan, “DITTO-2: Distilled Diffusion Inference-Time T-Optimization for Music Generation”, in *Proc. of the 25th Int. Society for Music Information Retrieval Conf.*, San Francisco, United States, 2024.

tude for faster-than-real-time generation for a wide-variety of controllable generation tasks including inpainting, outpainting, intensity, melody, and musical structure control. Our method works via 1) distilling a pre-trained diffusion model for fast sampling via an efficient, modified consistency model (CM) [21] or consistency trajectory model (CTM) [22] distillation process (only 32 GPU hours on a 40GB A100), (2) performing inference-time optimization using our distilled model with a 1-step *surrogate* objective, and (3) running a final multi-step sampling generation (decoding) using our estimated noise latents for final best-quality results as shown in Fig. 1. We find our approach accelerates optimization 10-20x, improves control, and improves audio quality at all once. Furthermore, we apply our approach to maximize text adherence (CLAP score) and show how an unconditional diffusion model trained without text inputs can yield SOTA text control.

2. BACKGROUND

2.1 Diffusion-Based Music Generation

Audio-domain music generation has become tractable through diffusion-based methods, popularized with models such as Riffusion [3], MusicLDM [23], and Stable Audio [4]. Diffusion Models (DMs) [8, 24] are defined using a closed-form forward process, where input audio is iteratively noised according to a Gaussian Markov Chain. DMs then learn to approximate the score of the probability distribution of the reverse process $\nabla_{x_t} \log q(x_t)$ using a noise prediction model ϵ_θ , which progressively denoises a random initial latent $x_T \sim \mathcal{N}(0, I)$ to generate new data x_0 . For audio-domain DMs, diffusion is performed over spectrograms [15] or on the latent representations of an audio-based VAE [3, 4, 23, 25], with an external vocoder used to translate spectrograms back to the time domain. Though DMs are efficiently trained by a simple MSE score matching objective [8, 26], sampling from DMs typically requires running the denoising process for 100s of iterations (calls to ϵ_θ), and have slower inference than VAEs or GANs [27].

2.2 Fast Diffusion Sampling

Fast diffusion sampling is critical. DDIM [8] or DPM-Solver [28] accelerates DMs to sample in only 10-50 sampling steps. To truly increase speed, however, *distillation* can be used to produce a model that can sample in a *single* step [21, 22, 29, 30]. Two promising DM distillation methods include consistency models (CM) [21] and consistency trajectory models [22]. The goal of CMs is to distill a base DM ϵ_θ into a new 1-step network $x_0 = \mathbf{G}_\phi(x_t, c)$ that satisfies the consistency property $\forall t, t' \in [T, 0], \mathbf{G}_\phi(x_t, c) = \mathbf{G}_\phi(x_{t'}, c)$ or that every point along the diffusion trajectory maps to the same output. Formally, CMs are distilled by enforcing local consistency between the learnable \mathbf{G}_ϕ and an exponential moving average (EMA) copy \mathbf{G}_{ϕ^-} :

$$\mathbb{E}_{t \sim T} (\mathbb{E}_{x \sim \mathcal{D}} \|\mathbf{G}_\phi(x_t, c) - \mathbf{G}_{\phi^-}(\Theta(\epsilon_\theta, x_t, c), c)\|_2^2), \quad (1)$$

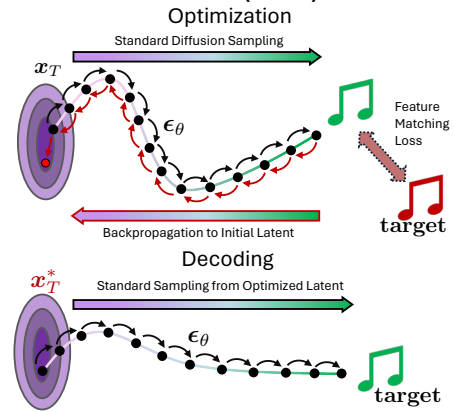


Figure 2: (Top) Baseline DITTO runs optimization over a multi-step sampling process to find an initial noise latent to achieve a desired stylized output, incurring a large speed cost. (Bottom) When generating the final output (decoding), the same multi-step diffusion sampling process is used.

where $\Theta(\epsilon_\theta, x_t, c)$ denotes one sampling step from x_t to x_{t-1} using the *frozen* teacher model ϵ_θ and some sampling algorithm (e.g. DDIM).

CMs are not perfect, however, and one-step performance lags behind DM quality [29]. Multi-step “ping-pong” sampling [21] also does not reliably increase quality due to compound approx. errors in each renoising step. CTMs [22], on the other hand, are designed to fix this problem. CTMs bridge the gap between CMs and DMs by distilling a model $x_s = \mathbf{G}_\phi(x_t, c, t, s)$ that can jump from *anywhere* t to *anywhere* s along the diffusion trajectory as shown in Fig. 3. CTMs then use γ -sampling to interpolate between few-step deterministic sampling along the trajectory ($\gamma = 0$) and CM’s “ping-pong” sampling ($\gamma = 1$), allowing a way to balance sampling stochasticity with overall quality. To our knowledge, CTM distillation is unexplored for audio, and CM distillation has only been applied to general audio [31].

2.3 Diffusion Inference-time Optimization

Diffusion inference-time optimization (DITTO) [18–20] is a general-purpose framework to control diffusion models at inference-time. The work is based on the observation that the *initial noise latent* x_T , traditionally thought of as a random seed, encodes a large proportion of the semantic content in the generation outputs [18, 32]. Thus, we can search for an initial noise latent of the diffusion generation process via optimization to achieve a desired stylized output as shown in Fig. 2. We do this by defining a differentiable feature extraction function (e.g. chroma-based melody extraction) $f(\cdot)$, a matching loss function \mathcal{L} (e.g. cross entropy), a target feature \mathbf{y} , and optimize x_T :

$$x_T^* = \arg \min_{x_T} \mathcal{L}(f(x_0), \mathbf{y}) \quad (2)$$

$$x_0 = \Theta_T(\epsilon_\theta, x_T, c), \quad (3)$$

where $\Theta_T(\epsilon_\theta, x_T, c)$ denotes T calls of the model using any sampler Θ . In practice, DITTO is run with a fixed budget of K optimization steps using a standard optimizer (i.e. Adam). This approach allows for any control that can

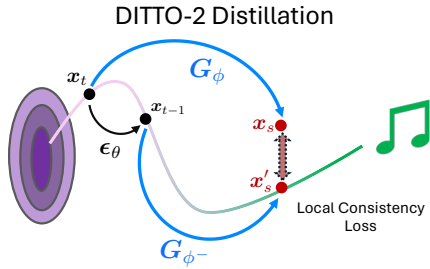


Figure 3: CTM Distillation for DITTO-2. We distill G_ϕ by minimizing the distance between the jump from x_t to x_s and x_{t-1} to x'_s , where x_{t-1} is generated by sampling with the base model ϵ_θ .

be parameterized differentially, including melody, intensity, and musical structure, as well as editing tasks like inpainting and outpainting. For brevity, we combine Eq. 2 and Eq. 3 into the shorthand $x_T^* = \arg \min_{x_T} \mathcal{L}_\theta^{(T)}(x_T)$.

The downside of DITTO, however, is that it is slow. We need to backpropagate through the *entire* sampling process for each of the K optimization steps and use memory management techniques like gradient checkpointing [33] or invertible networks [19] to handle large memory use that slows down generation. The overall cost of running a single ITO generation is on the order of $4KT$: T -step diffusion chain for K opt. steps, with a factor of 2 from gradient management and 2 from using classifier-free guidance (CFG) [34] to improve quality.

3. METHOD

3.1 Overview

We seek to dramatically speed up the diffusion ITO process to achieve controllable music generation for near-interactive rate music co-creation. To do so, we focus on three critical methodological improvements. First, we leverage **diffusion distillation** to significantly speed up diffusion sampling with an efficient, modified distillation process designed to be used together with ITO methods. Second, we introduce **surrogate optimization**, or the idea of decoupling the task of estimating noise latents from the task of rendering a final output or decoding, which allows us to leverage both fast sampling for optimization for control estimation and multi-step sampling for final, high-quality generation. Third, we combine diffusion distillation with surrogate optimization within the DITTO framework and produce a new, more efficient diffusion inference-optimization algorithm (no gradient checkpointing) as found in Section 3.4.

3.2 Acceleration through Diffusion Distillation

The clearest way to speed up ITO is to simply reduce the number of diffusion sampling steps T . From initial experiments, however, we found that (1) reducing the number of sampling steps T degrades overall generation quality [8], (2) quality degradation makes the optimization gradients weaker (as the outputs are less semantically coherent) leading to control degradation, and (3) achieving close to real-time performance requires < 4 sampling steps, which pro-

duce fully incoherent results on standard DMs. Thus, we employ distillation to speed up the diffusion process [29].

First, we develop CM distillation [21, 29, 31] for ITO-based controllable music generation. For CM distillation, we follow past work [29] for our training recipe, optimizing (1), and also learn an explicit embedding for the CFG scale w in the model $G_\phi(x_t, c, w)$ during distillation following [29]. By distilling CFG, we are able to half the number of total model calls per distilled diffusion sampling step. Once distilled, G_ϕ jumps from x_T to x_0 , allowing for deterministic 1-step sampling and stochastic multi-step-sampling by repeatedly *re-noising* with some $\epsilon \sim N(0, I)$ back to x_{t-1} .

Second, we develop CTM distillation [22] for ITO-based controllable music generation. CTM distillation offers more advantageous speed vs. quality design trade-offs, but comes at a cost of a more complex training procedure. In more detail, CTM distillation normally involves an expensive soft-consistency loss in the data domain with added GAN and score-matching loss terms. As we aim to distill our base model for *surrogate* optimization (see Sec. 3.3), we are able to simplify and speed up the CTM distillation process. First, we remove the image-domain GAN loss to reduce complexity of developing an audio-based GAN loss. Second, we use the consistency term from CTM in local-consistency form [35]:

$$\mathbb{E}_{t, s \sim T} \mathbb{E}_{(x, c) \sim \mathcal{D}} \|G_\phi(x_t, c, w, t, s) - G_{\phi^-}(\Theta(\epsilon_\theta, x_t, c), c, w, t-1, s)\|_2^2. \quad (4)$$

Third, we use the 1-step Euler parameterization of G_ϕ from [22]’s Appendix, which avoids explicitly learning additional parameters for the target step s in order to accelerate distillation. These changes reduce the number of per-training step model calls from 10-30 to 3, leading to a near order-of-magnitude speed up in wall clock time for performing the distillation process. Finally, we upgrade CTM’s unconditional framing for conditional diffusion by incorporating c into the distillation procedure, and adding w directly into the model to distill the CFG weight into an explicit parameter following past work [29], resulting in CFG control at inference but without double the complexity. In total, we perform distillation in as few as 32 GPU hours on an A100, the fastest trajectory-based distillation to our knowledge [22, 35].

3.3 Surrogate Optimization

Given a distilled CM or CTM model, we seek to minimize our inference runtime and maximize control adherence and audio quality. The obvious choice to minimize runtime is to use our distilled model with one-step sampling, but this results in limited audio quality and text-control. To solve this, we first split the ITO process into two separate phases: **optimization**, i.e. the nested loop of optimizing the initial latent over M -step multi-step sampling, and **decoding**, i.e. the final T -step sampling process from the optimized latent x_T^* , where $M = T$ in all past work [18, 19]. In this light, it is clear that the optimization phase is mostly respon-

Algorithm 1 Distilled Diffusion Inference-Time T -Optimization (DITTO-2)

input : G_ϕ , feature extractor f , loss \mathcal{L} , target \mathbf{y} , starting latent \mathbf{x}_T , text \mathbf{c} , optimization steps K , optimizer g , decoding steps $\{\tau_0, \dots, \tau_M\}$, γ , CFG weight w

- 1: // Optimization Loop
- 2: **for** K iterations **do**
- 3: $\mathbf{x}_0 = G_\phi(\mathbf{x}_T, \mathbf{c}, w, T, 0)$
- 4: $\hat{\mathbf{y}} = f(\mathbf{x}_0)$
- 5: $\mathbf{x}_T \leftarrow \mathbf{x}_T - g(\nabla_{\mathbf{x}_T} \mathcal{L}(\hat{\mathbf{y}}, \mathbf{y}))$
- 6: **end for**
- 7: // Decoding Loop
- 8: $\mathbf{x}_t \leftarrow \mathbf{x}_T$
- 9: **for** $t = M$ to 1 **do**
- 10: $\hat{\tau}_{t-1} = \sqrt{1 - \gamma^2 \tau_{t-1}}$
- 11: $\mathbf{x}_{t-1} = G_\phi(\mathbf{x}_t, \mathbf{c}, w, \tau_t, \hat{\tau}_{t-1}) + \gamma \tau_{t-1} \epsilon$
- 12: **end for**

output : \mathbf{x}_0

sible for the control strength and runtime, while decoding is generally responsible for final output quality.

Thus, we fix our final decoding process as multi-step sampling with T steps. Then, we perform control optimization over a **surrogate** objective $\hat{\mathbf{x}}_T^* = \arg \min_{\mathbf{x}_T} \mathcal{L}_\phi^{(M)}(\mathbf{x}_T)$ using some model ϵ_ϕ and $M \ll T$, where our surrogate is more efficient but yields approx. equal latents to our original objective

$$\arg \min_{\mathbf{x}_T} \mathcal{L}_\phi^{(M)}(\mathbf{x}_T) \approx \arg \min_{\mathbf{x}_T} \mathcal{L}_\theta^{(T)}(\mathbf{x}_T). \quad (5)$$

A natural candidate for a surrogate model would be the base DM ϵ_θ with fewer sampling steps. DM performance, however, becomes fully incoherent as $M \rightarrow 1$ [21, 22], causing a significant domain-gap when $M < T$. Alternatively, our distilled models are naturally strong surrogates:

- One-step outputs are generally coherent unlike in base DMs, resulting in more stable gradients when $M = 1$.
- Since distilled models excel at few-step sampling (i.e. < 8) [21, 22], the control domain gap between M and T can be reduced while ensuring coherent outputs.
- CTMs can increase quality with more sampling.

As a result, we use a CM or CTM G_ϕ as our surrogate, optimize with $M = 1$, and decode with $T \in [1, 8]$.

3.4 Complete Algorithm

Given our efficient CTM-based distillation process, and our surrogate objective, we propose a new ITO algorithm for controllable music generation in Alg. 1. Here, we run optimization to estimate control parameters using our surrogate 1-step objective. Then, we use the optimized latent \mathbf{x}_T^* and decode from our surrogate model with T steps using either multi-step CM Sampling (i.e. $\gamma = 1$) or CTM γ -sampling ($\gamma < 1$). Beyond decoupling optimization and decoding, we

also eliminate the need for gradient checkpointing found in the original DITTO method [18]. In total, we reduce the ITO speed from $4KT$ costly operations for DITTO to $K + T$.

4. EXPERIMENTS

To evaluate our proposed method, we follow the evaluation protocol used for DITTO [18] for intensity, melody, music structure, inpainting, and outpainting as described below. Before the full breadth of application tests, however, we explore our design space by comparing different distillation techniques and surrogate options on the task of intensity control. We further conclude with an experiment showing an adaptive sampling surrogate scheme as well a new experiment on maximizing text-adherence (CLAP score).

4.1 Controllable Generation Evaluation Protocol

We benchmark our method on five controllable music generation tasks from DITTO [18] including:

- **Intensity Control** [15, 18]: Here, we control the time-varying volume and overall semantic density to some target intensity curve \mathbf{y} using the extractor $f(\mathbf{x}_0) := \mathbf{w} * 20 \log_{10}(\text{RMS}(\mathbf{V}(\mathbf{x}_0)))$ (i.e. the RMS energy of the vocoder \mathbf{V} outputs smoothed with a Savitsky-Golay filter \mathbf{w}) and $\mathcal{L} = \|f(\mathbf{x}_0) - \mathbf{y}\|_2^2$.
- **Melody Control** [2, 15, 18]: We control the model outputs to match a given target melody $\mathbf{y} \in \{1, \dots, 12\}^{N \times 1}$ (where N is number of frames) using the chromagram of the model outputs $f(\mathbf{x}_0) = \log(\mathbf{C}(\mathbf{V}(\mathbf{x}_0)))$ and $\mathcal{L} = \text{NLLoss}(f(\mathbf{x}_0), \mathbf{y})$.
- **Musical Structure Control** [18]: We control the overall timbral structure of the model outputs by regressing the self-similarity (SS) matrix $f(\mathbf{x}_0) = \mathbf{T}(\mathbf{x}_0)\mathbf{T}(\mathbf{x}_0)^\top$ of Mel Frequency Cepstrum Coefficients (MFCC) \mathbf{T} against a target SS matrix \mathbf{y} like "ABA" form with $\mathcal{L} = \|f(\mathbf{x}_0) - \mathbf{y}\|_2^2$.
- **Inpainting and Outpainting** [16, 18]: Given music \mathbf{x}_{ref} , we can continue (outpainting) or infill (inpainting) \mathbf{x}_{ref} by matching the model outputs over o -length overlap regions $f(\mathbf{x}_0) := \mathbf{M}_{\text{gen}} \odot \mathbf{x}_0$ to the reference $\mathbf{y} = \mathbf{M}_{\text{ref}} \odot \mathbf{x}_{\text{ref}}$ (where \mathbf{M}_{ref} and \mathbf{M}_{gen} denote the overlap masks) and $\mathcal{L} = \|f(\mathbf{x}_0) - \mathbf{y}\|_2^2$.

For brevity, we focus on the $o = 1$ case for outpainting and inpainting (i.e. a gap of 4 seconds) and omit looping given its equivalence. See [18] for a more thorough description.

4.2 Pre-training and Distillation Details

For our base DM, we follow a similar setup and model design to DITTO [18], using the same base model and vocoder. Specifically, we train a 41M parameter Stable Diffusion-style 2D UNet directly over 6-second mel-spectrograms trained on ≈ 1800 hours of licensed music, and MusicHiFi [36] as the vocoder. The base model is trained with genre, mood, and tempo tags similar to [37] rather

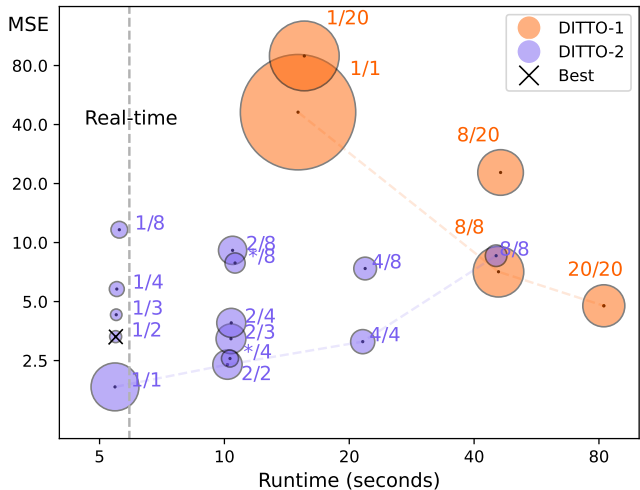


Figure 4: DITTO-2 inference speed vs. control MSE vs. audio quality (FAD, denoted by size, smaller is better). Dashed line denotes the cutoff for real-time performance, color denotes ITO method, and subscripts denote number of sampling steps during optimization / final decoding. Applied to intensity control. Trends also hold for CLAP score.

than full text descriptions. Both the CM and CTM surrogate models are distilled using a maximum of $T = 20$ sampling steps, evenly spaced across the trajectory for 4 hours across 8 A100 40GB GPUs on the same data. For DITTO-2, we use Adam. During CTM γ -sampling, we set $\gamma \in [0.05, 0.35]$ as empirically we found that using deterministic $\gamma = 0$ resulted in noticeable audio artifacts that degrade overall quality.

4.3 Metrics

For all tasks, we report the Fréchet Audio Distance (FAD) and CLAP Score with the CLAP [38] music backbone (as for FAD the standard VGGish backbone poorly correlates with human perception [39]), which measure overall audio quality and text relevance respectively across 2.5K generations. FAD is calculated with MusicCaps as the reference [1] dataset. Since our base model uses tags rather than captions, we convert each tag set into captions for CLAP Score calculation using the format “A [mood] [genre] song at [tempo] beats per minute.” Additionally, we report the MSE to the control target for intensity and structure control, and the overall accuracy for melody control.

5. RESULTS

5.1 Design Exploration Results

We study our design space for ITO via a case study on the task of intensity control. Notably, we compare DITTO with our proposed approach using CM and CTM distilled models in Fig. 4. We show runtime in seconds (x-axis), control MSE (y-axis), and FAD (point size) for an array of (M, T) combinations for our base DM, as well as our distilled CTM models (i.e. DM-8/20 corresponds to the base DM with $M = 8, T = 20$). We find that our distilled models are over 10x faster than the standard DITTO (20, 20)

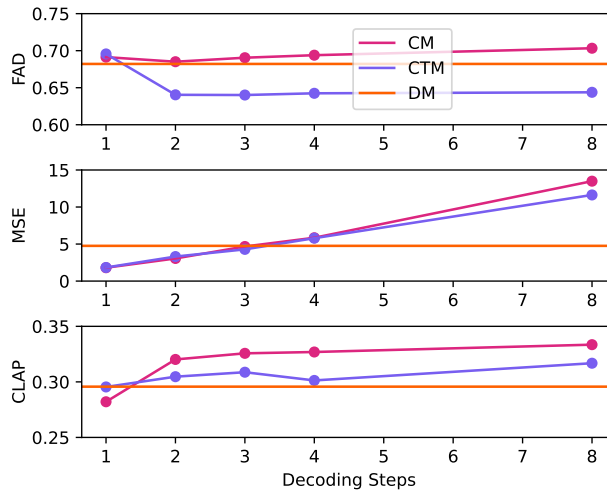


Figure 5: FAD, MSE, and CLAP results on Intensity Control for 1-step optimization, where orange lines denote baseline 20-step performance. MSE increases with more decoding steps for both CM/CTM given the domain gap though beats the baseline with < 4 steps. CM is unable to beat baseline quality due to accumulated errors in multi-step sampling, while multi-step CTM achieves SOTA quality.

configuration, while simultaneously achieving both better audio quality and control adherence. To understand these trends more in depth, specifically as we vary the number of decoding steps for our distilled models, we show both FAD (top), MSE (middle), and CLAP (bottom) in Fig. 5 as a function on number of decoding steps with $M = 1$, where the orange line denotes the baseline DITTO results with $M = T = 20$.

A few key points of DITTO-2 are visible here. Notably, both CM and CTM distilled models are able to achieve *better* control adherence than the base performance, as the shorter optimization process allows convergence to happen more effectively. Additionally, we find that CTM is clearly stronger than CM in terms of quality, as CTM is able to cleanly trade-off quality for control adherence (as sampling with more steps with $M = 1$ introduces a domain gap) and even *improve* baseline quality, while CM exhibits no real quality trend when sampling more due to its accumulated errors in multi-step sampling. In particular, CTM with $M = 1, T = 2$ achieves SOTA control adherence and FAD with faster than real-time. CM and CTM multi-step sampling also improves text relevance above the base DM.

5.2 Benchmark Results

We show full results on our suite of controllable music generation benchmarking results in Table 1. Here, we compare baseline DITTO with DITTO-2, where we display results for the best performing (M, T) setup in each experiment for CM and CTM, where best performance was chosen by finding the setup with the lowest latency (and thus best control adherence) while roughly matching the best overall FAD. As a whole, DITTO-2 achieves comparable or better performance than DITTO on all tasks with an 10-20x speedup, clearing the way for near real-time inference-time

Intensity	Time (s)	FAD	CLAP	MSE
DITTO	82.192	<u>0.682</u>	0.296	4.758
DITTO-2 (CM)	5.206	0.685	0.320	3.055
DITTO-2 (CTM)	<u>5.467</u>	0.640	<u>0.309</u>	<u>3.311</u>
Melody	Time (s)	FAD	CLAP	Acc.
DITTO	230.780	0.699	<u>0.283</u>	<u>82.625</u>
DITTO-2 (CM)	21.867	0.697	0.303	81.577
DITTO-2 (CTM)	<u>22.501</u>	<u>0.698</u>	0.273	85.226
Musical Structure	Time (s)	FAD	CLAP	MSE
DITTO	245.295	0.632	0.281	0.024
DITTO-2 (CM)	11.381	0.669	<u>0.234</u>	0.020
DITTO-2 (CTM)	<u>11.749</u>	<u>0.658</u>	<u>0.226</u>	<u>0.022</u>
Outpainting	Time (s)	FAD	CLAP	
DITTO	144.437	0.716	<u>0.343</u>	
DITTO-2 (CM)	6.658	<u>0.694</u>	0.319	
DITTO-2 (CTM)	<u>7.098</u>	0.680	0.347	
Inpainting	Time (s)	FAD	CLAP	
DITTO	145.486	0.690	<u>0.339</u>	
DITTO-2 (CM)	6.744	<u>0.689</u>	0.358	
DITTO-2 (CTM)	<u>6.814</u>	0.660	0.337	

Table 1: Controllable generation benchmark results. Best performing configuration for each DITTO-2 setup across five unique tasks. Both CM and CTM results yield excellent results with 10-20x speed ups.

M	T	Runtime	FAD	CLAP	MSE
1	1	5.447	0.696	0.295	1.835
1	2	5.467	0.640	0.307	3.311
1	4	5.502	0.643	0.301	5.792
2	2	10.171	0.659	0.281	2.384
2	4	10.387	0.658	0.296	3.894
Adaptive	4	10.315	0.644	0.296	2.561

Table 2: Intensity control results with various (M, T) options including an adaptive sampling during optimization.

controllable music generation. Specifically, we find that CTM outperforms CM, showing noticeably better quality with similar runtime and control adherence.

5.3 Variable Compute Budget Optimization

Though we are primarily interested in real-time performance (i.e. as fast as possible), we additionally investigated how we can use a varying compute budget during optimization (in terms of runtime). As simply increasing M predictably increases runtime by a multiplicative factor, we designed an *adaptive* schedule for M (denoted as $*$ in Fig. 4) in order to improve downstream decoding performance without increasing runtime significantly. Formally, for K optimization steps, we set the adaptive budget as using $M = 1$ for $\lfloor \frac{K}{2} \rfloor$ iterations, then $M = 2$ for $\lfloor \frac{3K}{8} \rfloor$, and finally $M = 4$ for $\lfloor \frac{K}{8} \rfloor$ iterations, thus allowing a coarse-to-fine optimization process. In Table 2, using the adaptive schedule exhibits the runtime of the $M = 2$ case yet achieves much better FAD and similar control adherence. This shows that given a more flexible compute budget, using an adaptive M schedule balances downstream performance better than simply modifying a fixed M , and allows smoother objective tradeoff between audio quality

Method	Condition	FAD	CLAP
Base TTM	Tags	0.488	0.167
DITTO-2	Tags	0.456	0.317
DITTO-2	N/A	0.440	<u>0.341</u>
U-DITTO-2	N/A	0.430	0.347
MusicGen (1.5B)	Caption	0.444	0.237
MusicGen (3.3B)	Caption	<u>0.437</u>	0.226

Table 3: Text similarity results. We use DITTO-2 to maximize CLAP similarity using a fully unconditional pre-trained diffusion model and yield a 54% relative improvement over past SOTA CLAP score (MusicGen).

and control strength.

5.4 Inference-time Optimization of Text-Control

Past ITO methods for music generation use simple feature extractors $f(\cdot)$ (i.e. chroma or RMS energy) [18] to minimize runtime speed. Given that our method is much faster, however, we can introduce new bespoke control applications with neural network-based feature extractors. Thus, we propose the task of inference-time **text similarity** control. We extract the normalized CLAP audio embedding [38] of our model outputs $f(\cdot) = \text{CLAP}(x_0)$ and, given some natural language caption y , calculate the cosine distance between the output and the normalized CLAP text embedding of the caption $\mathcal{L}(x_0) = 1 - f(x_0)^\top f(y)$.

Using FAD and CLAP score as metrics, we benchmark several configurations including our base DM model with tag inputs, DITTO-2 method with tag inputs, DITTO-2 method with null tag inputs, MusicGen w/melody (1.5B) [2], and MusicGen w.o./melody (3.5B) [2]. For models that take input text, we use captions from MusicCaps [1] as input and for models with tag inputs, we convert MusicCaps captions to tags via GPT-4 as done in past work [15] with tempo extracted from audio. Furthermore, to ablate whether any part of the tag-conditioned training process influences downstream DITTO-2 CLAP control, we retrain and distill our base model *without any text input*, which we denoted U-DITTO-2. In Table 3, we see that DITTO-2 enables SOTA text relevance compared to MusicGen by an over 54% relative improvement (large), thus showing the benefits of ITO-based approaches which allow us to directly optimize for desired downstream metrics, and notably enables fully-unconditional models to have text control *with no paired music-text training*.

6. CONCLUSION

We present DITTO-2: **Distilled Diffusion Inference-Time T-Optimization**, a new efficient method for accelerating inference-time optimization for fast controllable music generation. By utilizing a modified consistency or consistency trajectory distillation process and performing inference-time optimization on efficient surrogate objectives, we speed up past ITO methods by over 10-20x while simultaneously improving audio quality and text control. Furthermore, we find we can leverage the efficiency of our method on new, more complex tasks like text-adherence and show we can convert a fully unconditional diffusion model into a TTM model that yields SOTA results on evaluated metrics.

7. REFERENCES

- [1] A. Agostinelli, T. I. Denk, Z. Borsos, J. Engel, M. Verzetti, A. Caillon, Q. Huang, A. Jansen, A. Roberts, M. Tagliasacchi *et al.*, “MusicLM: Generating music from text,” *arXiv:2301.11325*, 2023.
- [2] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Défossez, “Simple and controllable music generation,” in *Neural Information Processing Systems (NeurIPS)*, 2023.
- [3] S. Forsgren and H. Martiros, “Riffusion: Stable diffusion for real-time music generation,” 2022. [Online]. Available: <https://riffusion.com/about>
- [4] Z. Evans, C. Carr, J. Taylor, S. H. Hawley, and J. Pons, “Fast timing-conditioned latent audio diffusion,” *International Conference on Machine Learning (ICML)*, 2024.
- [5] H. Liu, Z. Chen, Y. Yuan, X. Mei, X. Liu, D. Mandic, W. Wang, and M. D. Plumbley, “AudioLDM: Text-to-audio generation with latent diffusion models,” in *International Conference on Machine Learning (ICML)*, 2023.
- [6] H. Liu, Y. Yuan, X. Liu, X. Mei, Q. Kong, Q. Tian, Y. Wang, W. Wang, Y. Wang, and M. D. Plumbley, “Audioldm 2: Learning holistic audio generation with self-supervised pretraining,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2024.
- [7] J. Ho, A. Jain, and P. Abbeel, “Denoising diffusion probabilistic models,” *Neural Information Processing Systems (NeurIPS)*, 2020.
- [8] J. Song, C. Meng, and S. Ermon, “Denoising diffusion implicit models,” in *International Conference on Learning Representations (ICLR)*, 2020.
- [9] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, “High-resolution image synthesis with latent diffusion models,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.
- [10] K. Kumar, R. Kumar, T. De Boissiere, L. Gestin, W. Z. Teoh, J. Sotelo, A. De Brebisson, Y. Bengio, and A. C. Courville, “MelGAN: Generative adversarial networks for conditional waveform synthesis,” *Neural Information Processing Systems (NeurIPS)*, 2019.
- [11] N. Zeghidour, A. Luebs, A. Omran, J. Skoglund, and M. Tagliasacchi, “SoundStream: An end-to-end neural audio codec,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP)*, 2021.
- [12] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, “High fidelity neural audio compression,” *arXiv preprint arXiv:2210.13438*, 2022.
- [13] R. Kumar, P. Seetharaman, A. Luebs, I. Kumar, and K. Kumar, “High-fidelity audio compression with improved RVQGAN,” in *Neural Information Processing Systems (NeurIPS)*, 2023.
- [14] H. Manor and T. Michaeli, “Zero-shot unsupervised and text-based audio editing using ddpn inversion,” *International Conference on Machine Learning (ICML)*, 2024.
- [15] S.-L. Wu, C. Donahue, S. Watanabe, and N. J. Bryan, “Music ControlNet: Multiple time-varying controls for music generation,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP)*, 2024.
- [16] M. Levy, B. D. Giorgi, F. Weers, A. Katharopoulos, and T. Nickson, “Controllable music production with diffusion models and guidance gradients,” in *Diffusion Models Workshop at NeurIPS*, 2023.
- [17] J. Yu, Y. Wang, C. Zhao, B. Ghanem, and J. Zhang, “FreeDoM: Training-free energy-guided conditional diffusion model,” *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.
- [18] Z. Novack, J. McAuley, T. Berg-Kirkpatrick, and N. J. Bryan, “DITTO: Diffusion inference-time T-optimization for music generation,” in *International Conference on Machine Learning (ICML)*, 2024.
- [19] B. Wallace, A. Gokul, S. Ermon, and N. V. Naik, “End-to-end diffusion latent optimization improves classifier guidance,” *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.
- [20] K. Karunratanakul, K. Preechakul, E. Aksan, T. Beeler, S. Suwajanakorn, and S. Tang, “Optimizing diffusion noise can serve as universal motion priors,” *arXiv preprint arXiv:2312.11994*, 2023.
- [21] Y. Song, P. Dhariwal, M. Chen, and I. Sutskever, “Consistency models,” in *International Conference on Machine Learning (ICML)*, 2023.
- [22] D. Kim, C.-H. Lai, W.-H. Liao, N. Murata, Y. Takida, T. Uesaka, Y. He, Y. Mitsufuji, and S. Ermon, “Consistency trajectory models: Learning probability flow ODE trajectory of diffusion,” in *International Conference on Learning Representations (ICLR)*, 2023.
- [23] K. Chen, Y. Wu, H. Liu, M. Nezhurina, T. Berg-Kirkpatrick, and S. Dubnov, “MusicLDM: Enhancing novelty in text-to-music generation using beat-synchronous mixup strategies,” in *IEEE International Conference on Audio, Speech and Signal Processing (ICASSP)*, 2024.
- [24] P. Dhariwal and A. Nichol, “Diffusion models beat GANs on image synthesis,” *Neural Information Processing Systems (NeurIPS)*, 2021.
- [25] M. W. Y. Lam, Q. Tian, T.-C. Li, Z. Yin, S. Feng, M. Tu, Y. Ji, R. Xia, M. Ma, X. Song, J. Chen, Y. Wang, and Y. Wang, “Efficient neural music generation,” in *Neural Information Processing Systems (NeurIPS)*, 2023.

- [26] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, “Score-based generative modeling through stochastic differential equations,” in *International Conference on Learning Representations*, 2021.
- [27] M. Pasini and J. Schlüter, “Musika! fast infinite waveform music generation,” in *International Society for Music Information Retrieval Conference (ISMIR)*, 2022.
- [28] C. Lu, Y. Zhou, F. Bao, J. Chen, C. Li, and J. Zhu, “DPM-Solver++: Fast solver for guided sampling of diffusion probabilistic models,” *ArXiv*, vol. abs/2211.01095, 2022.
- [29] S. Luo, Y. Tan, L. Huang, J. Li, and H. Zhao, “Latent consistency models: Synthesizing high-resolution images with few-step inference,” *arXiv preprint arXiv:2310.04378*, 2023.
- [30] A. Sauer, D. Lorenz, A. Blattmann, and R. Rombach, “Adversarial diffusion distillation,” *ArXiv*, vol. abs/2311.17042, 2023. [Online]. Available: <https://api.semanticscholar.org/CorpusID:265466173>
- [31] Y. Bai, T. Dang, D. Tran, K. Koishida, and S. Sojoudi, “Accelerating diffusion-based text-to-audio generation with consistency distillation,” in *Interspeech*, 2024.
- [32] C. Si, Z. Huang, Y. Jiang, and Z. Liu, “FreeU: Free lunch in diffusion U-Net,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.
- [33] T. Chen, B. Xu, C. Zhang, and C. Guestrin, “Training deep nets with sublinear memory cost,” *arXiv preprint arXiv:1604.06174*, 2016.
- [34] J. Ho and T. Salimans, “Classifier-free diffusion guidance,” in *NeurIPS Workshop on Deep Gen. Models and Downstream Applications*, 2021.
- [35] J. Zheng, M. Hu, Z. Fan, C. Wang, C. Ding, D. Tao, and T.-J. Cham, “Trajectory consistency distillation: Improved latent consistency distillation by semi-linear consistency function with trajectory mapping,” 2024. [Online]. Available: <https://arxiv.org/abs/2402.19159>
- [36] G. Zhu, J.-P. Caceres, Z. Duan, and N. J. Bryan, “MusicHiFi: Fast high-fidelity stereo vocoding,” *IEEE Signal Processing Letters (SPL)*, 2024.
- [37] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, and I. Sutskever, “Jukebox: A generative model for music,” *arXiv:2005.00341*, 2020.
- [38] Y. Wu, K. Chen, T. Zhang, Y. Hui, T. Berg-Kirkpatrick, and S. Dubnov, “Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation,” in *IEEE International Conference on Audio, Speech and Signal Processing (ICASSP)*, 2023.
- [39] A. Gui, H. Gamper, S. Braun, and D. Emmanouilidou, “Adapting Frechet Audio Distance for generative music evaluation,” in *IEEE International Conference on Audio, Speech and Signal Processing (ICASSP)*, 2024.

THE CONCATENATOR: A BAYESIAN APPROACH TO REAL TIME CONCATENATIVE MUSAICING

Christopher J. Tralie

Ursinus College

Mathematics, Computer Science, And Statistics

Ben Cantil

DataMind Audio

ABSTRACT

We present “The Concatenator,” a real time system for audio-guided concatenative synthesis. Similarly to Driedger et al.’s “musaicing” (or “audio mosaicing”) technique, we concatenate a set number of windows within a corpus of audio to re-create the harmonic and percussive aspects of a target audio stream. Unlike Driedger’s NMF-based technique, however, we instead use an explicitly Bayesian point of view, where corpus window indices are hidden states and the target audio stream is an observation. We use a particle filter to infer the best hidden corpus states in real-time. Our transition model includes a tunable parameter to control the time-continuity of corpus grains, and our observation model allows users to prioritize how quickly windows change to match the target. Because the computational complexity of the system is independent of the corpus size, our system scales to corpora that are hours long, which is an important feature in the age of vast audio data collections. Within The Concatenator module itself, composers can vary grain length, fit to target, and pitch shift in real time while reacting to the sounds they hear, enabling them to rapidly iterate ideas. To conclude our work, we evaluate our system with extensive quantitative tests of the effects of parameters, as well as a qualitative evaluation with artistic insights. Based on the quality of the results, we believe the real-time capability unlocks new avenues for musical expression and control, suitable for live performance and modular synthesis integration, which furthermore represents an essential breakthrough in concatenative synthesis technology.

1. INTRODUCTION

Concatenative synthesis, or audio mosaicing, is a data-driven approach to arrange granular fragments of audio samples, particularly using data sourced from the spectral-temporal features of a target sound. While granular synthesis systems typically rely on combinations of aleatoric parameterization, deterministic automation, and traditional synthesis modulation to achieve complex and evolving textures from sound fragments [1], concatenative synthesis al-

gorithms utilize Music Information Retrieval technology to decide parameters such as the index, amplitude, and pitch of each sound fragment.

Modern music producers are inundated by audio data. Services like Splice offer hundreds of thousands of samples readily available on the cloud, and Kontakt multi-sample libraries can often take up over 10gb of disk space to capture a single instrument. Music Producers generate plenty of their own audio data as well: stems, multi-tracks, long-form recordings, and mix variations account for a large portion of many a music producer’s audio collection. Recent software such as XO by XLN Audio, Sononym, and Ableton Live 12 offer automatic organization of audio files based on various tags and descriptors, but these implementations of MIR technology are more utilitarian than creative in their design and application. Meanwhile, concatenative synthesis options remain sparse since its conceptual inception [2]: Reformer by Krotos is designed to create foley designs, apps like Samplebrain and CataRT [3, 4] are lacking in critical musical areas such as pitch tracking, with the more advanced options having limited accessibility for artists, requiring prior knowledge of Max (FluCoMa, MuBu) or Python (Audioguide).

The Concatenator advances concatenative synthesis in 3 major ways: 1) it is capable of accurately reproducing harmonic and percussive sounds using arbitrary corpora 2) in real-time at scale, 3) affording new levels of control and accessibility. Furthermore, unlike neural audio systems [5], it requires no training and can adapt to arbitrary corpora at runtime. The speed, ease, and scope of The Concatenator offers a fresh paradigm for music producers to interact creatively with their ever-expanding excess of audio data, leading to what we believe is a breakthrough in the field.

2. RELATED WORK

We build on important works in Bayesian inference, particle filters, concatenative synthesis, and applied nonnegative matrix factorization (NMF), which we briefly describe

Driedger’s Technique. From an artistic point of view, the most similar technique to ours is Driedger et al.’s 2015 “Let It Bee” concatenative musaicing technique [6], which uses NMF to learn activations of spectral window templates in a **corpus collection** so that their combination will match a **target** spectrogram. This technique was a fruitful innovation in sound design for electronic music production, as featured heavily on *Zero Point* by Rob Clouth [7],



© C.J. Tralie, B. Cantil. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** C.J. Tralie, B. Cantil, “The Concatenator: A Bayesian Approach To Real Time Concatenative Musaicing”, in *Proc. of the 25th Int. Society for Music Information Retrieval Conf.*, San Francisco, United States, 2024.

using custom software also authored by Clouth. The algorithm was also implemented in an open source python script in 2018 [8], and in Max by the FluCoMa project in 2021 (fluid.bufnmcross) [9], which made NMF-inspired audio mosaicing accessible enough to contribute towards the production of at least two more albums heavily featuring the technique: *Edenic Mosaics* by Encanti (2021) [10] and *Hate Devours Its Host* by Valance Drakes (2023) [11].

We now detail the mathematics of Driedger et al.’s technique, as we borrow a few ideas in our work. Driedger et al. learn H in the equation $V \approx WH$, where V is an $M \times T$ target spectrogram with M frequency bins and T times, W is an $M \times N$ set of N spectral corpus templates that are treated as fixed, and H is a matrix of $N \times T$ learned activations. For instance, W could be the windows of a collection of buzzing bees and V could be an excerpt from The Beatles’ “Let It Be” (hence the title). Driedger et al. use the Kullback-Liebler (KL) divergence loss, an instance of the more general β -divergence [12], to measure the goodness of fit of WH to V . This loss function is

$$D(V||WH) = \sum V \odot \log \left(\frac{V}{WH} \right) - V + WH \quad (1)$$

where \odot , $/$, $+$, and $-$ are all applied element-wise, and the sum is taken over all elements of the resulting matrix. As Lee/Seung show, choosing the right step size turns gradient descent of Equation 1, with respect to W and H , into *multiplicative update rules* that guarantee monotonic convergence. Driedger et al. keep W fixed to force the final audio to use exact copies of the templates, so only the update rule for H is relevant. At iteration ℓ , this is:

$$H_{kt}^\ell \leftarrow H_{kt}^{\ell-1} \left(\frac{\sum_m W_{mk} V_{mt} / (WH^{\ell-1})_{mt}}{\sum_m W_{mk}} \right) \quad (2)$$

Crucially, though, Driedger et al. note that the update rules in Equation 2 alone will lose the timbral character of the templates in W . They hence disrupt ordinary KL gradient descent by performing several increasingly impactful modifications to H before Equation 2 in each step, which are eventually set in stone after L total iterations. First, they avoid repeated windows to avoid a “jittering” effect, allowing a particular window k to only activate once in some r -length interval based on where it’s the strongest:

$$(H_r)_{kt}^\ell \leftarrow \left\{ \begin{array}{ll} H_{kt}^{\ell-1} & H_{kt}^{\ell-1} > H_{ks}^{\ell-1}, |t-s| \leq r \\ H_{kt}^{\ell-1} (1 - \frac{\ell+1}{L}) & \text{otherwise} \end{array} \right\} \quad (3)$$

They also promote sparsity similarly by shrinking all but the top p activations in each column of H_r to create H_p^ℓ . Finally, they encourage *time continuous activations* by doing “diagonal enhancement,” or by doing a windowed sum down each diagonal of H_p , assuming the columns of W are also in a time order.

$$(H_c)_{kt}^\ell = \sum_{i=-c}^c (H_p)_{k+i,t+i}^\ell \quad (4)$$

Since this encourages the algorithm to mash up chunks of W in a time order, it effectively encourages sound grains from the templates than the length of a single window that ordinary NMF would take. Finally, Driedger et al. apply Equation 2 to H_c^ℓ instead of $H^{\ell-1}$ to obtain H^ℓ .

These disruptions remove the guarantee that Equation 1 will be minimized, or that it will even monotonically decrease, but Driedger et al.’s key insight is that the loss function is merely a guide to choose reasonable activations; a suboptimal fit leaves room to better preserve timbral characteristics. We take a similar perspective.

Driedger Tweaks. The idea of spectrogram decomposition used for concatenative musaicing goes back to the work of [13]. Beyond that, the authors of [12] provide some improvements to Driedger et al.’s technique, including mixing corpus windows directly rather than performing phase retrieval on WH . One issue with Driedger et al.’s technique is the sources have to be augmented with pitch shifts to span additional pitches in the target, increasing memory consumption and runtime. The authors of [14, 15] avoid this by using 2D deconvolutional NMF [16] on the Constant-Q transform, whereby pitch shifts are modeled as constant shifts of the activations instead of the templates, saving memory. The other convolutional axis models time history and time shifts, avoiding the need for the diagonal enhancement of Equation 4. The authors apply 2D NMF to both the source and target, so they do not preserve the original sound grains. However, for our preferred style, we want to take the source grains exactly as they are.

Other Concatenative Techniques. Schwarz created an offline concatenative synthesis system dubbed “Caterpillar” that uses the Viterbi algorithm [2], which he later approximated with a real time system, “CataRT” that uses a greedy approach instead of the Viterbi algorithm [3, 4]. Simon’s “audio analogies” is quite similar [17], but instead of a user controlled traversal through timbral space, they use features from some source (e.g. midi audio) to guide synthesis to a target with a different timbre (e.g. real audio of someone playing a trumpet). Caterpillar and audio analogies are both *sequentially Bayesian in nature*, where the *hidden state* is the template to concatenate, and the “observation” is a user-controlled trajectory or features from a source timbre, respectively. The prior transition probabilities are based on temporal continuity. However, they use the Viterbi algorithm, which is computationally intensive and which needs all time history, so it cannot be applied in real time. By contrast, a **particle filter** is a scalable Monte Carlo method for sequential Bayesian inference [18–20]. It is less common in MIR, but it has found use in a few real time MIR applications such as multi-pitch tracking [21], tempo tracking [22, 23], and beat tracking [24].

3. THE CONCATENATOR

The NMF technique of Driedger is not suitable for real-time applications; the gradient update rules of Equation 2 scale linearly in the length of the corpus, leaving all but minutes long corpora usable (Section 3.3), and the equations to suppress repeated windows and promote time con-

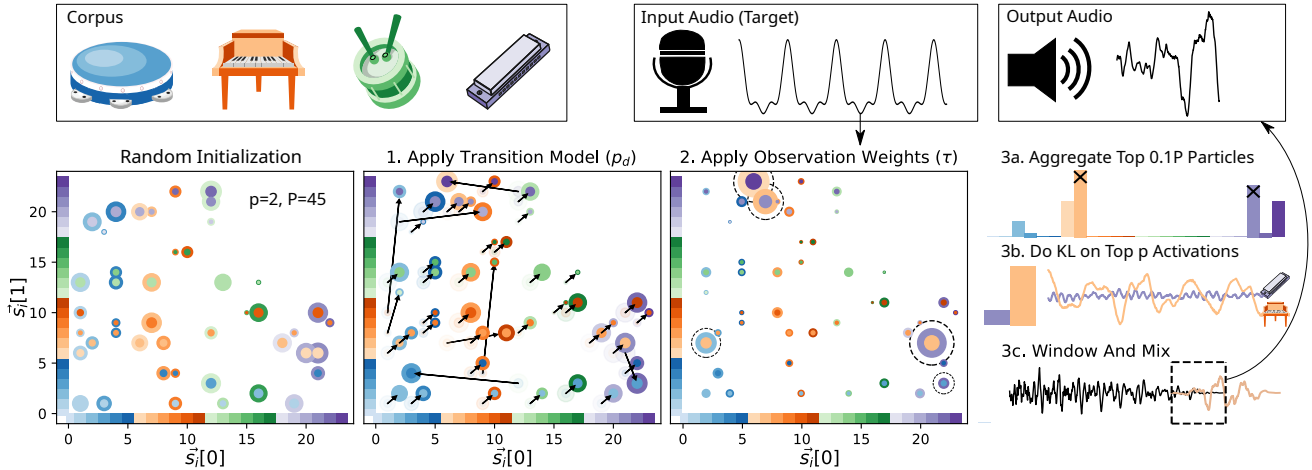


Figure 1. The Concatenator maintains P “particles,” each of which represents p specific corpus windows. Each window moves forward by 1 timestep in the corpus with probability p_d , or otherwise jumps randomly. Then, particles each mix their windows to best match the target, and particles with the top 10% best fits to the target vote on a final set of p windows.

tinuity at each entry of H require knowledge of all activations in H , including future activations. Instead, The Concatenator does many tiny KL-based NMF problems (Equation 2) online in “particles” based on random sampling at each timestep. The particles then vote on a final set of activations to use at that timestep (Figure 1)¹. The random sampling trades off historical context to choose longer grains, with fit to the target audio streaming in. We provide the mathematical and implementation specifics below.

3.1 Sequential Bayesian Formulation And State Space

Formally, The Concatenator uses a sequential Bayesian formulation, where the t^{th} column of the target spectrogram V is the “observation,” at time t . The hidden state indexes p out of N possible windows in the corpus spectrogram W . We use a particle filter to efficiently infer the the best such windows (Section 3.2). Henceforth, we refer to the observations as vectors \vec{v}_t to emphasize that the data is streaming, and we focus on one timestep t at a time.

State space. To keep the state space simple, we decouple which windows are active from their activation weights; we only model the former as the hidden state, while we infer the weights as a best fit under the KL-loss (Equation 1). To control for polyphony directly, we use a p -sparse nonnegative integer-valued vector $\vec{s}_t \in \mathbb{N}^p$ as the hidden state. This vector indexes the p corpus windows that are active at time t , where p is fixed ahead of time. For convenience of implementation, template indices can repeat and are in no particular order:

$$\vec{s}_t[k] \in \{0, 1, \dots, N-1\}, k = 0, 1, \dots, p-1 \quad (5)$$

We then infer the associated nonnegative weights $\vec{h}_t[k]$ for each activation to give the approximation $\vec{\Lambda}_t$ at time t :

$$\vec{\Lambda}_t[m] = \sum_{k=0}^{p-1} \vec{h}_t[k] W_{m, \vec{s}_t[k]} \quad (6)$$

¹ CC musical instrument images adapted from vectorportal.com

In particular, given W , \vec{s}_t , and \vec{v}_t , we apply the update rules of Equation 2 for a pre-specified number L of iterations, using the corresponding columns \vec{s}_t of W

$$\vec{h}_t^\ell[k] \leftarrow \vec{h}_t^{\ell-1}[k] \left(\frac{\sum_m (W_{m, \vec{s}_t[k]})(\vec{v}_t[m]) / (\vec{\Lambda}_t^{\ell-1}[m])}{\sum_m W_{m, \vec{s}_t[k]}} \right) \quad (7)$$

Transition Model. We use the KL-loss (Equation 1) to measure the spectral fit of Λ_t to \vec{v}_t . As in Driedger et al. [6] (Equation 4), however, we are willing to sacrifice fit to take longer grains from the corpus W . To that end, we define the prior **state transition probability** in the as a Factorial Hidden Markov Model (FHMM) [25]. Each \vec{s}_t satisfies the Markov property and is conditionally independent of all previous steps given \vec{s}_{t-1} , but *each component* k of $\vec{s}_t[k]$ also transitions independently of other components, leading to the following transition probability:

$$p_T(\vec{s}_t = \vec{b} | \vec{s}_{t-1} = \vec{a}) = \prod_{k=0}^{p-1} \left\{ \begin{array}{ll} p_d & \vec{b}[k] = \vec{a}[k] + 1 \\ \frac{1-p_d}{N-1} & \text{otherwise} \end{array} \right\} \quad (8)$$

where $p_d \in [0, 1]$ is the “probability of remaining time-continuous.” Intuitively, if $p_d > 0.5$, then we are more likely to continue to use a time-continuous activation than we are to jump to a new random activation, which promotes longer contiguous sound grains from the corpus, even at the expense of a lower fit to the spectral template². As such, p_d a parameter that can be tuned by the artist and set closer to 1 to promote longer grains. We generally find $p_d \in [0.9, 0.99]$ to be effective (Section 4.1).

We must also specify the *observation probability*, which pulls the states closer to matching \vec{v}_t , even if they have to jump away from time continuity; otherwise, the

² This has a similar effect to “extend matches” functionality in Sturm’s MatConcat [26] when a match isn’t found. In our Bayesian framework, such extensions happen on a continuum based on fit to target.

result would sound nothing like the target. Though each component transitions independently, they all contribute jointly to an observation, which makes inference trickier than it is for traditional HMMs.

3.2 Sampling, Observing, And Synthesizing

We now describe how to apply Bayesian inference to find the sequence of corpus windows \vec{s}_t and their activation weights \vec{h}_t that maximize the posterior probability given the transition model in Equation 8 and the observation model below. While the authors of [27] use a similar FHMM applied to multi-pitch tracking, inferring the hidden states via message passing algorithms known as “Max-Sum” [28] and “Junction Tree” [29], we need a faster technique which is also real-time, and which has tunable accuracy that degrades gracefully with restricted computational resources. To that end, we turn to a particle filter.

Our particle filter consists of P particles, each of which is a p -dimensional state vector (Equation 5) that we refer to as \vec{s}_i . The particles traverse the corpus over time, and they each have a weight w_i that keeps track of the posterior probability of its accumulated motion over all timesteps (we now dispense with the time index t on \vec{s}_i and w_i since t will be clear from context). Since each particle is its own estimate of a state that best describes what templates to choose, our goal is to sample them in such a way that (at least some of) the particles are close to capturing activations that maximize the posterior probability given all \vec{v}_t .

Tracking Weights. All particles begin with even weights $w_i = 1/P$. At the beginning of each time step, we sample new indices for each \vec{s}_i according to Equation 8. Then, we multiply each weight by the **observation probability** p_O . Given the KL loss d_i between the i^{th} particle’s spectral approximation $\vec{\Lambda}_i$ (Equation 6) and \vec{v}_t after L iterations of Equation 7, for each particle i , p_O is:

$$p_O[i] = \frac{e^{-\tau d_i}}{\sum_j e^{-\tau d_j}} \quad (9)$$

In other words, the observation probability is a softmax over KL-based goodness of fits of \vec{s}_i to \vec{v}_t , and the softmax has a “temperature” τ . We use a negative exponential since a larger d_i loss indicates a poorer fit using windows \vec{s}_i and hence, should be a lower probability. Intuitively, a higher τ will emphasize particles that fit the observation better, putting more importance on the observation relative than the transition probability. This is tunable and has a similar effect to varying p_d in the transition, as we will explore more in Section 4.1. After multiplying each w_i by $p_O[i]$, we normalize the weights so that they sum to 1.

Resampling. The above is a naive particle filter, but it suffers from “sample impoverishment,” where a few particles stand out with high weights and the rest are stuck with vanishing weights, leaving the system unable to adapt to new observations. To ameliorate this, we compute a standard definition of the “effective number of particles” $n_{\text{eff}} = 1/(\sum_i w_i^2)$, which is maximized when all particles have equal weight $1/P$. If n_{eff} goes below $0.1P$ at a particular time step, we resample the particles with

stochastic universal sampling [30,31], an $O(P)$ resampling technique, and reset all weights to $1/P$ before continuing. This leads to “survival of the fittest” where particles with a higher weight are more likely to be replicated and those with a lower weight are more likely to be eliminated.

Synthesizing audio. After updating the weights, we take a weighted average of the windows in the top $0.1P$ particles, with the option to further boost windows that follow continuously from those chosen in previous steps. We also ignore windows that would be repeated from up to r timesteps in the past (analogous to Driedger’s Equation 3). We then let \vec{s}_t be the top p such windows by weight, and we compute the corresponding activations \vec{h}_t . These steps can be done in $O(Pp)$ time with hash tables and linear time selection. Finally, we mix together the corresponding waveforms from the corpus (as in [12]) and apply a Hann window to overlap-add this audio to the output stream.

3.3 Computational Complexity

The dominant cost of both The Concatenator and of Driedger is computing activations via KL iterations. Given N corpus templates, T times in the target, and a spectral dimension of M , for L KL iterations, the time complexity of Driedger (Equation 2) is $O(LMNT)$. This is a *linear* dependency on the corpus length. So if, for example, Driedger’s technique takes a minute on a target sourcing a corpus that’s a minute in length, it will take 2 hours a 2-hour corpus on that same target. To improve this scaling, the authors of [12] do a greedy nearest neighbors search in the corpus, but this requires tuning and may miss important windows. In fact, our random sampling naturally scales in an even more favorable way. Specifically, given P particles and p windows per particle, the time complexity of our analogous Equation 7 is only $O(LPMpT)$, which does not scale with the corpus size N at all (though P may need to scale with N for the best results (Section 4.1)). As an example, for a 60 minute corpus a window length of 2048 ($M = 1025$, hop=1024) at a sample rate of 44.1khz, using $P = 1000$ and $p = 5$, this is a speedup of nearly 30x over Driedger. Moreover, propagating particles and applying the observation model are also embarrassingly parallelizable at the particle level, which we leverage in our implementation. Finally, while Driedger et al. use $L = 20$ [6], we find that $L = 10$ is sufficient in our context.

3.4 Bells And Whistles (Pun Intended)

Regularizing Quiet Moments in The Corpus. One pitfall using KL-based NMF is that if enough activations are near silence, Equation 7 becomes numerically unstable and the weights \vec{h}_i can approach ∞ . To address this, we modify the KL-loss to include a masked L_2 penalty for \vec{h}_i for the i^{th} particle for the target \vec{v}_t at time t . Given the corresponding approximation $\vec{\Lambda}_i$ (Equation 6), the modified loss is

$$D(\vec{v}_t || \vec{\Lambda}_i) = \left(\sum \vec{v}_t \odot \log \left(\frac{\vec{v}_t}{\vec{\Lambda}_i} \right) - \vec{v}_t + \vec{\Lambda}_i \right) + \frac{\|\alpha \odot \vec{h}_i\|_2^2}{2} \quad (10)$$

where, abusing notation, α is a mask that is a fixed value (we use 0.1) if the corresponding corpus window is less than -50dB and 0 otherwise. Equation 7 then turns into

$$\vec{h}_i^\ell[k] \leftarrow \vec{h}_i^{\ell-1}[k] \left(\frac{\sum_m (W_{m, \vec{s}_i[k]})(\vec{v}_t[m]) / (\Lambda_i^{\ell-1}[m])}{(\sum_m W_{m, \vec{s}_i[k]} + \alpha[k] \vec{h}_i^{\ell-1}[k])} \right) \quad (11)$$

Intuitively, if $s_i[k]$ is a quiet corpus window, $\alpha[k] = 0.1$, which shrinks $\vec{h}_i^{\ell-1}[k]$ down³.

Pitch Shifting. Though we don't use this in Section 4, we implemented Driedger et al.'s technique to increase the pitch coverage of the corpus; that is, we can replicate the corpus in its entirety for different pitch shifts that are chosen up-front. This only incurs a preprocessing cost since the complexity of The Concatenator is independent of corpus length (Section 3.3), which does not impact real-time performance once the system starts. However, our system could choose a different trade-off of space and time complexity by augmenting the state space as the Cartesian product of window indices and pitch shifts. Pitch shifts could be computed on the corpus audio *on demand* whenever a state with a nonzero pitch shift is chosen.

Finally, for a fixed corpus with or without pitch shifts, the user can control a slider that pitch shifts the *target* in real time, so that the chosen windows move relatively to the audio input. This could be used, for example, to harmonize to singing in an interval that's a fifth away.

3.5 How Many Particles?

In practice, few particles are surprisingly effective at capturing windows that fit the target, which we explain with a simple probabilistic argument. Given a corpus with N sound grains (including pitch shifts) and P particles that each capture p windows, suppose also that we have a hypothetical "ideal particle" \vec{s}_t with the p best windows at time t , which are completely disjoint from all current particles; the only way to jump to the best windows is to randomly resample with probability $(1 - p_d)$. Since we use a small hop length relative to the sample rate ($1024/44100 \approx 23$ ms), we have a few timesteps to jump without a large effect on the final audio. Also, there are usually several windows in the corpus that sound acceptably similar to windows in \vec{s}_t . Let δ be the maximum tolerable offset before or after in time for choosing the best windows, and let w be a factor of acceptable windows (e.g. $w = 11$ would consider each window in \vec{x} and its ten most similar in the corpus). Assuming all offsets of acceptable windows are disjoint, then the probability of jumping to at least one of the top k windows of \vec{s}_t , or to one of their acceptably close corresponding offsets, is:

$$1 - \left(p_d + (1 - p_d) \frac{(N - 1 - wk)}{N - 1} \right)^{(2\delta+1)pP} \quad (12)$$

For example, for $p_d = 0.95$, $\delta = 2$ and $w = 11$, and $N = 10000$ (≈ 4 min corpus), the probabilities are 0.747,

³ For a derivation of similar additive constraints on NMF, refer to [32]

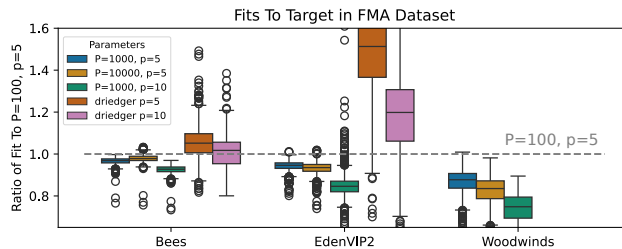


Figure 2. Increasing polyphony leads to a better fit (ratios < 1), and increasing particles leads to a better fit, especially for larger corpora like the Woodwinds (≈ 1.6 hrs).

0.936, 0.983 for $k = 1, 2, 3$, respectively. These probabilities all degrade when N gets larger for a larger corpus, but in that case, it is likely that the acceptable w is also larger.

Furthermore, once one of the particles catches on to a good window in the corpus, it is promoted with a high weight and gets carried on to a longer grain. This is similar to how the "patch match" technique in computer graphics [33, 34] computes nearest neighbors of many nearby patches by starting with a random initialization of nearest neighbors, and then well-matched to patches correct the nearest neighbors of spatially adjacent patches [33].

4. EVALUATION

4.1 Quantitative Evaluation

To empirically assess reliability, we do an extensive MIR-style evaluation, which is much more comprehensive than standard evaluation in other concatenative synthesis works.

Effect of Parameters. First, to complement our analysis in Section 3.5, we want to empirically examine how many particles are needed for different sized corpora. We also want guarantee the impact of important parameters in our system for artistic control. We select 3 corpora: Driedger's buzzing bees (small, 66 seconds), a corpus used in *Edenic Mosaics* [10] known as "EdenVIP2," which consists of various real-world percussive sounds (medium, 10.5 minutes), and all Woodwind clips from the pre-2012 UIowa MIS dataset [35] (large, ≈ 1.6 hours). Then, we randomly subsample 1000 30 second clips from the Free Music Archive (FMA)-small dataset [36], each of which we use as a target for the three different corpora for various parameter choices. We use a sample rate of 44.1khz for all corpora, we use stereo audio for the bees and EdenVIP2, and we use mono audio for the Woodwinds.

First, we assess the effect of particles on fit; we fix $p_d = 0.95$, temperature $\tau = 10$, and $r = 3$, using $L = 10$ iterations for all KL operations, and we take $P \in \{100, 1000, 10000\}$. We also compare to Driedger et al.'s technique with $c = 3$ and $r = 3$ using $L = 50$ iterations, though we omit comparisons with Woodwinds due to computational cost (Section 3.3). In all cases, we use frequencies from 0 to 8000hz with a sample rate of 44100hz, a window length of 2048 samples, and a hop length of 1024 samples. Since the spectral similarity of different targets to a particular corpus varies widely, we report the *ratio* of the

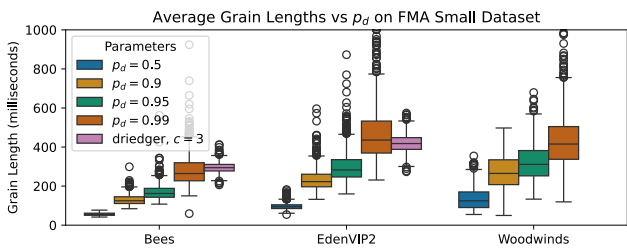


Figure 3. Increasing p_d increases the average grain length since windows are less likely to jump at each timestep.

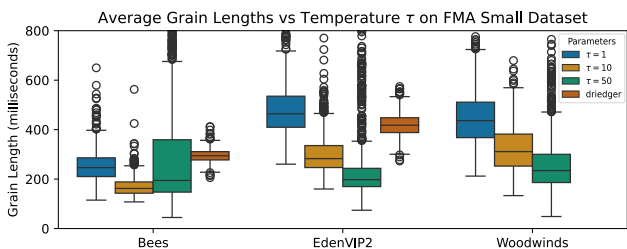


Figure 4. Increasing τ decreases the average grain length since this prioritizes the observation probability.

KL loss in Equation 1 to the KL loss for The Concatenator with $P = 100, p = 5$. Figure 2 shows the results. As expected, an increased polyphony leads to a better fit, as does increasing particles for all but the Bees, though the effect of increased particles is most pronounced for the largest corpus of Woodwinds, which makes sense by Equation 12.

As we noted in Section 2, however, a very good fit may lose the timbral characteristics of the corpus. A lower p helps, but we also need to ensure that grains are long enough. Therefore, we also examine mean grain length for various parameters. Figure 3 shows the result of varying p_d for a fixed temperature $\tau = 10$ and $p = 5$, and Figure 4 shows the result of varying the temperature τ for $p = 5$ and $p_d = 0.95$. As expected, grain length goes up with increased p_d and down with increased τ . In practice, lowering τ and raising p_d will lead to especially long grain lengths, albeit with a lower target fit.

Reproducing Pitch. In addition to fits and grain lengths, we quantify how well The Concatenator reproduces target pitch. Using the Woodwinds corpus, we create targets out of all stems in the MDB-stem-synth dataset [37]. We compare ground truth pitch annotations of the stems to the pitches estimated with CREPE [38] on both

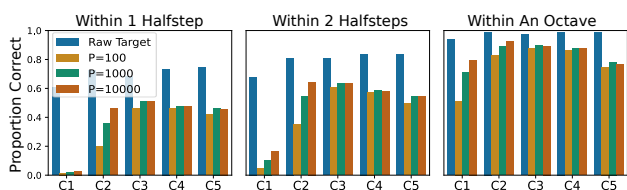


Figure 5. The Concatenator reproduces reasonably correct pitches in the 1.6 hour Woodwinds corpus with targets in MDB-stem-synth, in real time (at $P = 100, 1000$), at all but the lowest octave C1.

the raw target and the synthesized audio for various P , and we break the results down by octave. Figure 5 reports the proportion of pitches correctly identified at each 23ms hop length to within different tolerances, over all stems. Even though CREPE was not trained on concatenated audio, it reports pitch nearly as clearly as on the raw target for most octaves except for C1, which makes sense since the spectral resolution is only 21.5hz. We can mitigate this in the current system by increasing the window, at the expense of temporal resolution. In the future, though, we would like to try a streaming CQT that can better capture lower frequencies. Finally, since the bassoon is the only instrument out of 10 in the Woodwinds that has notes in the C2 octave, additional particles are needed for precise pitch in that octave, which can be explained by w in Equation 12.

4.2 Qualitative Evaluation

This algorithm was tested in a variety of contexts to assess its performance and accuracy for applications in music and sound design. Our Corpora contained audio samples that fell into the following categories: Test Tones, Percussion, Full Mixes, Sample Libraries, Foley, and Driedger Comparisons. Our Targets were single audio files that were designed to test how the Concatenator re-created varying kinds of melody, counterpoint, full mixes, basses, drums, vocals, noise, and prior examples used with the Driedger algorithm. Our tests reveal that the Concatenator performs highly accurately in pitch reproduction for most melodies, two-part harmonies, and full mixes that contain prominent melodic features, while struggling with accurate reproduction of more complex three-part harmony. Given the nature of the particle filter, which rotates through new temporal positions in the corpus at random, some notes are more accurate than others, and some notes are dropped all together, as expected from our quantitative analysis. While this tendency might make the Concatenator unfit for replacing the role of large multi-sample instruments, the vast majority of pitches remain wholly accurate while the aleatoric variation of off-color audio grains may represent an entirely desirable aesthetic quality of its own. Similarly for drums, sometimes transients are incredibly accurate, while other times they sound a little smeared. This tendency is due to the particle filter’s random positioning, and can be improved by increasing the particle amount.

4.3 Supplementary Material / Discussion

We include supplementary material at <https://www.centralie.com/TheConcatenator>. This includes a python prototype for the real-time system that uses port audio [39], audio examples for all corpus/target pairings in Section 4.2, and a video showing artistic examples of what the real time system enables in the loop with Ableton Live.

This is only the beginning. Since The Concatenator exists feedback loop, we expect artists will go much deeper, likely well beyond the “obstacle course” we put it through.

5. REFERENCES

- [1] C. Roads, “Automated granular synthesis of sound,” in *Computer Music Journal*, 1978, pp. vol.2, p.61.
- [2] D. Schwarz, “A system for data-driven concatenative sound synthesis,” in *3rd International Conference on Digital Audio Effects (DAFx)*, 2000, pp. 97–102.
- [3] D. Schwarz, G. Beller, B. Verbrugghe, and S. Britton, “Real-time corpus-based concatenative synthesis with catart,” in *9th International Conference on Digital Audio Effects (DAFx)*, 2006, pp. 279–282.
- [4] D. Schwarz, R. Cahen, and S. Britton, “Principles and applications of interactive corpus-based concatenative synthesis,” in *Journées d’Informatique Musicale (JIM)*, 2008, pp. 1–1.
- [5] A. Bitton, P. Esling, and T. Harada, “Neural granular sound synthesis,” in *International Computer Music Conference*, 2020.
- [6] J. Driedger, T. Prätzlich, and M. Müller, “Let it bee-towards nmf-inspired audio mosaicing,” in *Proceedings of 16th International Society for Music Information Retrieval (ISMIR)*, 2015, pp. 350–356.
- [7] R. Clouth, “Zero point,” <https://robclouth.com/zero-point>, 2020.
- [8] C. Tralie, “Let it bee,” <https://github.com/ctrlalief/LetItBee>, 2018.
- [9] O. Green, G. Roma, P. A. Tremblay, J. Bradbury, F. Cameli, A. Harker, and T. Moore, “Fluid corpus manipulation: Max objects library,” <https://github.com/flucoma/flucoma-max>, 2021.
- [10] B. Cantil, “Edenic mosaics,” 2021.
- [11] V. Drakes, “Hate devours its host,” <https://amekcollective.bandcamp.com/album/hate-devours-its-host>, 2023.
- [12] M. Buch, E. Quinton, and B. L. Sturm, “Nichtnegative-matrixfaktorisierungnutzendesklangsynthesensystem (nimfks): Extensions of nmf-based concatenative sound synthesis,” in *Proceedings of the 20th International Conference on Digital Audio Effects*, 2017, p. 7.
- [13] J. J. Burred, “Cross-synthesis based on spectrogram factorization,” in *ICMC*, 2013.
- [14] H. Foughmand Aarabi and G. Peeters, “Multi-source musaicing using non-negative matrix factor 2-d deconvolution,” in *18th International Society for Music Information Retrieval (ISMIR) Late-Breaking Demo Session*, 2017.
- [15] —, “Music retiler: Using nmf2d source separation for audio mosaicing,” in *Proceedings of the Audio Mostly 2018 on Sound in Immersion and Emotion*, 2018, pp. 1–7.
- [16] M. N. Schmidt and M. Mørup, “Nonnegative matrix factor 2-d deconvolution for blind single channel source separation,” in *International Conference on Independent Component Analysis and Signal Separation*. Springer, 2006, pp. 700–707.
- [17] I. Simon, S. Basu, D. Salesin, and M. Agrawala, “Audio analogies: Creating new music from an existing performance by concatenative synthesis,” in *ICMC*. Citeseer, 2005.
- [18] N. Metropolis and S. Ulam, “The monte carlo method,” *Journal of the American statistical association*, vol. 44, no. 247, pp. 335–341, 1949.
- [19] A. Doucet, S. Godsill, and C. Andrieu, “On sequential monte carlo sampling methods for bayesian filtering,” *Statistics and computing*, vol. 10, pp. 197–208, 2000.
- [20] S. Thrun, “Probabilistic robotics,” *Communications of the ACM*, vol. 45, no. 3, pp. 52–57, 2002.
- [21] Z. Duan and B. Pardo, “A state space model for online polyphonic audio-score alignment,” in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2011, pp. 197–200.
- [22] A. T. Cemgil and B. Kappen, “Monte carlo methods for tempo tracking and rhythm quantization,” *Journal of artificial intelligence research*, vol. 18, pp. 45–81, 2003.
- [23] S. W. Hainsworth and M. D. Macleod, “Particle filtering applied to musical tempo tracking,” *EURASIP Journal on Advances in Signal Processing*, vol. 2004, pp. 1–11, 2004.
- [24] M. Heydari and Z. Duan, “Don’t look back: An online beat tracking method using rnn and enhanced particle filtering,” in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 236–240.
- [25] Z. Ghahramani and M. Jordan, “Factorial hidden markov models,” *Advances in neural information processing systems*, vol. 8, 1995.
- [26] B. L. Sturm, “Matconcat: An application for exploring concatenative sound synthesis using matlab,” in *7th International Conference on Digital Audio Effects (DAFx)*, 2004.
- [27] M. Wohlmayr, M. Stark, and F. Pernkopf, “A probabilistic interaction model for multipitch tracking with factorial hidden markov models,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 799–810, 2010.
- [28] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, “Factor graphs and the sum-product algorithm,” *IEEE Transactions on information theory*, vol. 47, no. 2, pp. 498–519, 2001.

- [29] F. V. Jensen *et al.*, *An introduction to Bayesian networks*. UCL press London, 1996, vol. 210.
- [30] G. Kitagawa, “Monte carlo filter and smoother for non-gaussian nonlinear state space models,” *Journal of computational and graphical statistics*, vol. 5, no. 1, pp. 1–25, 1996.
- [31] J. Carpenter, P. Clifford, and P. Fearnhead, “Improved particle filter for nonlinear problems,” *IEE Proceedings-Radar, Sonar and Navigation*, vol. 146, no. 1, pp. 2–7, 1999.
- [32] T. Virtanen, “Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria,” *IEEE Transactions on Audio, Speech, And Language Processing*, vol. 15, no. 3, 2007.
- [33] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, “PatchMatch: A randomized correspondence algorithm for structural image editing,” *ACM Transactions on Graphics (Proc. SIGGRAPH)*, vol. 28, no. 3, Aug. 2009.
- [34] C. Barnes, E. Shechtman, D. B. Goldman, and A. Finkelstein, “The generalized PatchMatch correspondence algorithm,” in *European Conference on Computer Vision*, Sep. 2010.
- [35] “The university of iowa musical instrument samples,” <https://theremin.music.uiowa.edu>, last Accessed: 2024-04-05.
- [36] M. Defferrard, K. Benzi, P. Vandergheynst, and X. Bresson, “FMA: A dataset for music analysis,” in *18th International Society for Music Information Retrieval Conference (ISMIR)*, 2017. [Online]. Available: <https://arxiv.org/abs/1612.01840>
- [37] J. Salamon, R. M. Bittner, J. Bonada, J. J. Bosch, E. Gómez Gutiérrez, and J. P. Bello, “An analysis/synthesis framework for automatic f0 annotation of multitrack datasets,” in *Hu X, Cunningham SJ, Turnbull D, Duan Z. ISMIR 2017 Proceedings of the 18th International Society for Music Information Retrieval Conference; 2017 Oct 23-27; Suzhou, China.[Suzhou]: ISMIR; 2017. International Society for Music Information Retrieval (ISMIR), 2017.*
- [38] J. W. Kim, J. Salamon, P. Li, and J. P. Bello, “Crepe: A convolutional representation for pitch estimation,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 161–165.
- [39] R. Bencina and P. Burk, “Portaudio-an open source cross platform audio api,” in *ICMC*, 2001.

DEEP RECOMBINANT TRANSFORMER: ENHANCING LOOP COMPATIBILITY IN DIGITAL MUSIC PRODUCTION

Muhammad Taimoor Haseeb*
MBZUAI

Ahmad Hammoudeh*
MBZUAI

Gus Xia
MBZUAI

ABSTRACT

The widespread availability of music loops has revolutionized music production. However, combining loops requires a nuanced understanding of musical compatibility that can be difficult to learn and time-consuming. This study concentrates on the 'vertical problem' of music loop compatibility, which pertains to layering different loops to create a harmonious blend. The main limitation to applying deep learning in this domain is the absence of a large, high-quality, labeled dataset containing both positive and negative pairs. To address this, we synthesize high-quality audio from multi-track MIDI datasets containing independent instrument stems, and then extract loops to serve as positive pairs. This provides models with instrument-level information when learning compatibility. Moreover, we improve the generation of negative examples by matching the key and tempo of candidate loops, and then employing AutoMashUpper [1] to identify incompatible loops. Creating a large dataset allows us to introduce and examine the application of Transformer architectures for addressing vertical loop compatibility. Experimental results show that our method outperforms the previous state-of-the-art, achieving an 18.6% higher accuracy across multiple genres. Subjective assessments rate our model higher in seamlessly and creatively combining loops, underscoring our method's effectiveness. We name our approach the Deep Recombinant Transformer and provide audio samples¹.

1. INTRODUCTION

The widespread availability of music loops used in Digital Audio Workstations (DAWs) has revolutionized music production. For example, *Umbrella* by Rihanna, composed using the "Vintage Funk Kit 03" GarageBand loop, transformed a royalty-free sample into a Grammy-winning global hit [2]. However, combining loops requires a nuanced understanding of musical compatibility and mostly relies on manual selection. Furthermore, the vast number of available loops presents a daunting challenge in deciding which loops pair well, leading to a combinatorial prob-

* The first two authors contributed equally.

¹ Samples available at: <https://conference-demo-2024.github.io/demo/>

lem. Finding compatible loops was recognized as one of the grand challenges in MIR research [3].

The loop compatibility problem can be broken down into two sub-problems: the *vertical* problem and the *horizontal* problem. The vertical problem pertains to the layering of different loops — and understanding how rhythm, melody, and harmony interact within a single moment of music — to create a harmonious blend. Conversely, the horizontal problem addresses the sequencing of loops over time, ensuring that transitions between different loops are smooth and maintain the overall coherence of the musical piece. This research focuses on the vertical problem.

A major limitation to applying deep learning to this domain has been the absence of high-quality, labeled, datasets. Previous works propose source separating existing music, extracting loops from each stem, and creating positive pairs [4,5]. Source separation models produce these four stems: *vocal*, *bass*, *drum*, and *other*. The outputs of source separation models are not perfect and often suffer from noise and distortion. Moreover, the "other" category can include a wide range of sounds — for example, entire string sections — and can be too *noisy* for the model to learn what makes two loops compatible. To generate negative samples, loop reversal, beat shifting, or key and tempo modifications are made to a loop in a positive pair. Altering loop characteristics to generate negative samples risks misleading models to distinguish these superficial differences rather than learning true musical incompatibility.

Our proposed solution to the above mentioned problems is to generate positive examples using MIDI datasets containing independent stems for each instrument, synthesizing them into audio, and extracting loops. This provides models with more granular information about *each* instrument when learning loop compatibility. Similarly, we find that while AutoMashUpper (AMU) demonstrates modest success in identifying compatible loops, its strength lies in accurately identifying incompatible loops after we match the tempo, key, and phase of query and target loops — thereby providing more realistic negative samples for model training [1]. Obtaining a large, high-quality, labeled dataset allows us to introduce and examine the application of Transformer-based architectures for addressing the vertical loop compatibility problem.

Our method outperforms the previous state-of-the-art for loop compatibility by 18.6% higher accuracy, proving its versatility and robustness across 13 genres through rigorous evaluations. Our contributions are as follows:

1. **A novel method to generate a large, high-quality, labeled dataset** for models to learn musical compatibility by providing positive and negative loop pairs that



share identical keys, tempos, and phases.

2. **Transformer-based architectures** to enhance accuracy for instrument-level music loop compatibility.
3. **Extensive objective and subjective assessments** demonstrating our method’s effectiveness.

2. RELATED WORK

Two approaches exist in the literature: rule-based and learning-based. Rule-based methods establish a set of rules to generate a compatibility score. In contrast, learning-based methods require positive and negative examples to train models for compatibility prediction. We review both.

2.1 Rule-Based Methods

Davies et al. set the groundwork for loop compatibility [6]. Their model, AutoMashUpper, computes mashability estimation by evaluating a weighted average of harmonic and rhythmic compatibility, and spectral balance across key-adjusted sections within a loop database. Best matching loops are aligned through time stretching and pitch shifting to match the query loop. Davies et al. introduce further improvements in a subsequent study, enhancing their system’s capabilities [1]. Key improvements include the development of a faster algorithm for assessing harmonic similarity, integration of rhythm and loudness for mashability evaluation, and a subjective evaluation to assess the overall mashability of music pieces. Later works use this as a baseline to compare loop compatibility performance.

Lee et al. introduce a framework incorporating both vertical and horizontal dimensions of musical segments to create harmonious mashups [7]. Features include tempo, beat-synchronous chromagram, chord signatures, Mel Frequency Cepstral Coefficients, and volume levels. The system uses a Group of Background Units (GBU) from a specific track, typically comprising multiple background units that adhere to prevalent structures found in popular music genres, forming the foundational layer of a mashup. It evaluates potential lead units to layer atop the established GBUs, which pivots on three factors: Harmonic Matching which determines the harmonic compatibility between lead and background units, Harmonic Change Balance monitors the rate of harmonic transitions between to reduce monotony, while Volume Weighting calibrates the audibility of lead units. The framework computes a vertical mashability score for each candidate pair and selects those with the highest compatibility. Tsuzuki et al. overlay vocal tracks from other artists who have performed the same piece, aligning them with the instrumental track [8].

Bernardes et al. assess the harmonic compatibility of musical tracks through small- and large-scale structures [9]. Small-scale compatibility is determined by blending dissonance and perceptual relatedness, derived from the Tonal Interval Space [10], resilient to instrumental timbral variations. Large-scale compatibility is based on key estimations, aiding in overarching harmonic planning. Software showcases these metrics through interactive visualization to aid in finding harmonically compatible tracks. Maças et al. present MixMash by building on this method [9, 11]. MixMash enhances user interaction through a

force-directed graph that visualizes multidimensional musical attributes like hierarchical harmonic compatibility, onset density, spectral region, and timbral similarity. The visualization represents tracks as nodes with varying distances and connections indicating their compatibility.

2.2 Learning-Based Methods

A major limitation in using Deep Neural Networks to evaluate the compatibility of musical loops has been the lack of adequately labeled datasets. Chen et al. are the first to use neural network models [4]. First, they propose an innovative pipeline to generate a labeled dataset using the Free Music Archive. To create positive samples they employ an unsupervised MSS algorithm that isolates looped content [12]. Negative samples are created by editing a loop in a positive loop pair by doing one of three things: reversing, randomly shifting beats, or rearranging beats of one of the loops. They propose using two architectures, a Convolution Network (CNN) and a Siamese Network (SNN), to learn compatibility between two loops. While both models outperform traditional rule-based systems, the CNN model demonstrates superior performance. Subsequent studies have identified limitations in the proposed data acquisition process. Specifically, it employs multiple heuristics for source separation and does not ensure the outputs consist of distinct instruments, e.g. a positive training example could comprise two similar drum loops [5]. In addition, they restrict their work to hip-hop without exploring how well this approach generalizes to other genres. Finally, they use a two-second input which may not capture the complexity and variability of longer musical pieces.

Huang et al. introduce an alternative method for assembling a training dataset by developing their own supervised music source separation model, which splits tracks into four distinct stems: *vocal*, *bass*, *drum*, and *other*. While it is an innovative approach, it leaves serious gaps in dataset quality. The outputs of source separation models are not perfect and often suffer from noise and distortion. In addition, the "other" category can include a wide range of sounds — e.g. guitars, pianos, trumpets, saxophones, violins, cellos, ambient sounds, synthesizers, reverb, choirs, flutes, clarinets, and even entire string sections. Since professional-grade musical loops contain distinct sounds — a guitar riff, a saxophone lick, a violin jig, etc. — using "other" may be too *noisy* for the model to learn what makes two instrument loops compatible. Similarly, as observed by Chen et al., bass and drum loops seamlessly adapt across most genres and styles when matched for tempo and key, and are somewhat trivial to learn for the model. On the other hand, they generate negative samples by varying the basic characteristics of a loop — key, tempo, or phase shifts. Even though it guarantees incompatibility, simply altering the basic loop characteristics to generate negative examples risks misleading the model to learn to distinguish these superficial differences rather than understanding true musical incompatibility. Instead, we propose training models to determine compatibility between loops sharing identical tempo, key, and phase to mirror choices made in actual music production — a significantly more challenging task. While there are some similarities, Huang

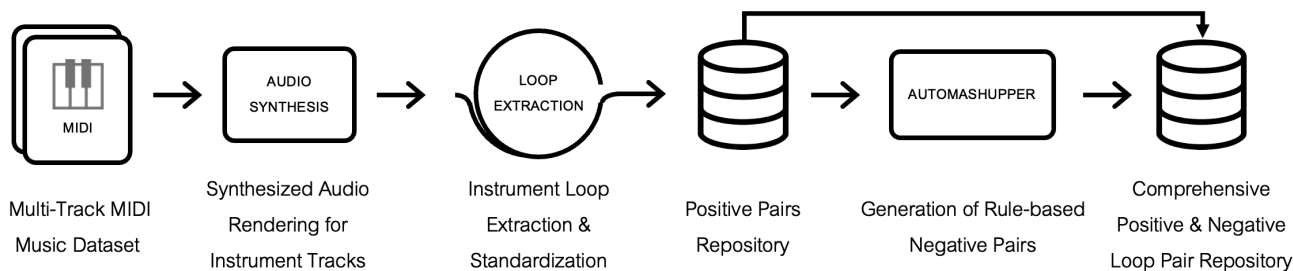


Figure 1. Our dataset generation pipeline takes multi-track MIDI music as input and generates a labeled loop dataset.

et al. do not extract loops and instead use complete stems to train a model on mash-ups involving combinations of vocal and backing track stems; whereas our research delves into the compatibility of instrumental music loops. Therefore, we use Chen et al. as a baseline for our work.

Broadly, despite their utility, these methods do not capture the complex interplay of musical elements, underscoring the necessity for more advanced methods. With the availability of a larger dataset, we introduce and examine the application of Transformer-based architectures for addressing the vertical loop compatibility problem.

3. DATA GENERATION PIPELINE

We introduce a novel self-supervised method to create a large, high-quality, labeled dataset to provide instrument-level granularity. Our method also ensures identical basic attributes — such as tempo, key, and phase — amongst incompatible pairs to compel models to learn compatibility and not focus on such superficial differences.

3.1 Generating Positive Examples

Synthesized datasets have shown promise in enhancing model performance across various music information retrieval tasks, including transcription, understanding compositional semantics, sound synthesis, and instrument recognition [13]. In the absence of an instrument-level labeled dataset, we modify the data collection pipeline proposed by Chen et al. to instead create loops from synthesized data [4]. Similar to Flakh, we generate our dataset by taking songs from the Lakh dataset, rendering MIDI files using sample-based synthesizer and then extracting loops [13, 14]. For this task, we used FluidSynth².

The Lakh MIDI dataset, with over 175,000 unique MIDI files, provides detailed musical score data for various instruments that can be synthesized due to distinct track segmentation. We chose files with significant parts for piano, bass, guitar, and drums. A total of 20,371 files are identified, of which 15,000 were taken at random and rendered [13]. Each MIDI file is split into individual instrument tracks, matched with appropriate patches based on program numbers, and rendered into audio. As observed by Chen et al., when adjusted for tempo and key, drum and bass loops tend to be universally compatible. Therefore, all drum and bass MIDI tracks were removed from synthesis and subsequent creation of positive and negative

pairs [4]. The collected set of 15,000 songs spans 13 genres and 47 instruments. We then use the same method as Chen et al. to extract loops from each rendered audio [4]. Of the 15,000 songs, 12,193 songs have at least one valid loop pair. Specifically, of these 12,193 songs, we obtained 126,746 loops and 90,376 valid positive pairs of loops. This provides our training models with more granular information about *each* instrument loop while learning what constitutes compatibility. Files were separated into training (72,301 loop pairs), validation (9,037 loop pairs), and testing (9,038 loop pairs) — leaving us with a total of 251 hours of positive examples, with roughly equal representation of instruments and genres in each set. To ensure consistency, we standardize the duration of each loop to 10 seconds by either repeating or trimming the loops.

3.2 Generating Negative Examples

Generating pairs of negative loops is a difficult task. One naive approach could be to randomly select combinations from our loop set. However, this does not guarantee incompatibility. Unlike what’s been proposed in similar works, we argue that simply altering the basic loop characteristics to guarantee the generation of negative examples risks misleading the model to learn to distinguish these superficial differences rather than understanding true musical incompatibility. Instead, to reflect real-life music production choices, we train models to determine incompatibility between loops sharing identical tempo, key, and phase. We find that while AutoMashUpper demonstrates modest success in identifying compatible loops, its strength lies in identifying incompatible loops within the same tempo, key, and phase, thus furnishing reliable negative labels for compatibility modeling. Inversely applying the original method focuses on *least* compatible pairs. Harmonic incompatibility finds significant chord progression clashes, rhythmic incompatibility leads to off-sync combinations, and spectral imbalance points to lopsided energy distributions, cultivating disturbances and noise.

We adopt AMU to a subset of loops by drawing, without replacement, 1,500 positive pairs (3,000 loops) from varied genres and instruments. For each loop in this collection, we calculate its incompatibility against every other loop by adjusting the target loop’s keys and tempos to match the query loop and then calculating weighted sums of harmonic, rhythmic, and spectral compatibility between the source and target. For this task, we use a Python im-

² Available at: <http://www.fluidsynth.org/>

plementation³ of the Krumhansl-Schmuckler key-finding algorithm [15], Rubber Band⁴ for sound stretching and pitch-shifting, and weights proposed by Bernardo — 0.4 for both harmonic and rhythmic, and 0.2 for spectral compatibility — to derive an overall compatibility score between loop pairs [16]. After obtaining all scores, the 35 least compatible loops are paired with each loop in the set. We exclude any duplicate pairs, culminating in 95,281 unique negative pairs. More than 1,000 negative pairs are tested at random by the research team to confirm incompatibility. The final pairs are then partitioned into training (76,225 loop pairs), validation (9,528 loop pairs), and testing segments (9,528 loop pairs), leaving us with a total of 264 hours of negative examples. The negative set can be significantly expanded by drawing more pairs at the start.

Data Type	# Loops	# Loop Pairs	# Hours
Training	101,397	148,526	412
Validation	12,674	18,565	51.5
Test	12,675	18,566	51.5
In Total	126,746	185,657	515

Table 1. Overview of data, including positive and negative examples, across training, validation, and test subsets.

4. MODEL ARCHITECTURE

Recent advancements in self-attention networks, particularly the Transformer architecture, provide a new perspective for solving the vertical loop compatibility challenge [17]. In this study, a large labeled dataset allows us to introduce and examine the application of Transformer-based architectures. Specifically, we use the same model architecture as MusicTaggingTransformer (MTT), proposed by Won et al., for its robustness on other MIR tasks [18]. We refer to this adapted Transformer architecture as the Deep Recombinant Transformer (DRT). Initial pre-processing employs MelSpectrogram transformation and AmplitudeToDB conversion of the input, which comprises the summed audio signals of two candidate loops. This is followed by Res2DMaxPoolModule for downsampling, with subsequent convolutional layers and max-pooling operations for detailed feature extraction. The core Transformer architecture is equipped with 256-dimensional attention vectors across four layers and eight heads, PreNorm, Residual structures, and GELU-activated Feed Forward networks for processing. A unique class ([CLS]) token, alongside positional embedding, is added to the feature set for sequence analysis. The output from the Transformer is directed through a sigmoid function, mapping the high-dimensional feature vectors to a binary outcome space, and delineating the likelihood of each audio sample belonging to a specific category. Then, we compute the binary cross entropy loss (BCELoss) to update the parameters of the whole model. Model’s output is between 0 and 1, with values closer to 1 indicating a higher probability that the pair of loops are compatible, and closer to 0 when they are not. Therefore, we can use its output

³ <https://pypi.org/project/pymusickit/>

⁴ <https://breakfastquay.com/rubberband/>

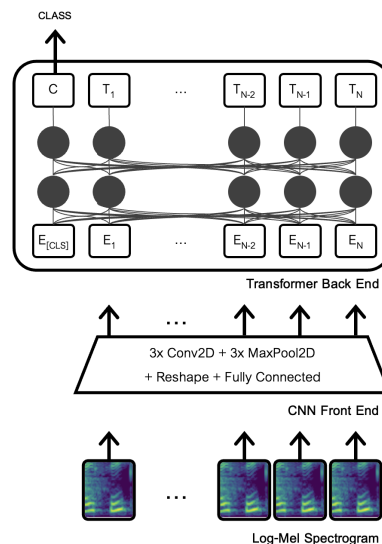


Figure 2. Architecture of Deep Recombinant Transformer.

to estimate the compatibility of any two loops. Dropout (0.1) and batch normalization strategies are implemented to mitigate over-fitting and ensure robust model generalization. This integration of convolutional and Transformer elements captures both local and global audio features for deep and context-aware analysis. To investigate the adaptability and performance of the Transformer architecture for this task, our study explores two distinct configurations: one variant employs two-encoder layers, while the other utilizes four-encoder layers. This enables us to evaluate the impact of architectural depth on model performance.

5. EXPERIMENT SETUP AND EVALUATION

We evaluate the performance of Transformer architectures in identifying compatible loops against the state-of-the-art. Following this, we focus on understanding the impact of our new dataset on model performance. Finally, we conduct a subjective assessment.

5.1 Effect of Using a Transformer

First, we compare the performance of Transformers against CNN-based architectures. We train and evaluate two configurations for the Transformer architecture — two and four encoder layers. For comparison, we explore the performance of the original CNN-based NLC⁵ model proposed by Chen et al. Initially, we adhere to the original NLC specification, applying it to two-second audio segments extracted from our 10-second dataset. However, we also train a modified NLC on a 10-second long input for a fair comparison. Moreover, acknowledging that extended audio contexts may require a deeper CNN architecture, we also train Short-chunk CNN Res [19], a deeper CNN architecture, due to its strong performance on MIR-related classification tasks. We perform hyperparameter optimization for each architecture using unseen validation sets.

⁵ <https://github.com/mir-aidj/neural-loop-combiner>

The first type of evaluation entails a classification task. It assesses a model’s ability to distinguish compatible loops from incompatible ones. We report accuracy and F1 scores for each model. Table 2 summarizes these results.

Model	Accuracy ↑	F1 Score ↑
NLC (2 seconds)	62.25	68.76
NLC (10 seconds)	60.50	70.76
Short Chunk CNN Res	70.50	77.13
DRT (2 Attn Layers)	78.60	82.02
DRT (4 Attn Layers)	80.90	83.66

Table 2. Comparative performance on loop compatibility classification task for models trained using our dataset.

Model	Avg. ↓ Rank	Top ↑ 10	Top ↑ 30	Top ↑ 50
NLC (2 seconds)	43.4	0.25	0.44	0.56
NLC (10 seconds)	51.2	0.13	0.25	0.52
Short Chunk CNN Res	38.3	0.07	0.46	0.77
DRT (2 Attn Layers)	25.7	0.15	0.69	1.00
DRT (4 Attn Layers)	16.2	0.44	0.75	1.00

Table 3. Comparative performance on loop ranking task for models trained using our dataset, using average rank and accuracy in the top-k positions across 100 queries.

Another performance evaluation reported in the research involves ranking candidate loops by compatibility with a particular query loop [4, 5]. This assessment is especially important for a model’s practical use, which seeks to find loops that match a specific query from a large collection of loops. Using AMU, and the unseen test set, we create a collection of candidate loops for each query loop, ensuring that precisely one of these candidates pairs positively with the query. The model’s performance is measured by where the "target loop" ranks in the list, with a higher position indicating better performance. Following the benchmark set by Chen et al., we also assess the compatibility of exactly 100 candidate loops balanced across genres and instruments. Each model is evaluated for accuracy within the top 10, 30, and 50 positions, as well as the mean rank. Table 3 shows these aggregated averages.

The results indicate that the four attention layer Music-TaggingTransformer demonstrates superior performance across loop compatibility classification and ranking tasks. We also observe that models, though not explicitly trained for it, perform well in identifying compatible drum and bass loops, confirming these are relatively trivial to learn.

5.2 Effect of Using Our Dataset

Generating negative examples by altering loop characteristics can mislead models toward learning superficial differences instead of true musical incompatibility. To objectively evaluate this, we create a control dataset using the negative sampling methodology proposed by Chen et al. In this control dataset, the positive pairs remain the same, while for negative pairs we reverse, randomly shift beats, or re-arrange beats of one of the loops. Although reversing

performed best in the original study, the performance differences across the three strategies were small. To account for potential non-transferability to our dataset, we include an equal representation of all three methods in our control set. We retrain the three best-performing architectures from Table 2, and evaluate them, on this control dataset. The classification results are summarized in Table 4 while the retrieval results are summarized in Table 5.

Model	Accuracy ↑	F1 Score ↑
NLC (2 seconds)	66.4	71.4
Short Chunk CNN Res	88.9	89.3
DRT (4 Attn Layers)	88.2	88.7

Table 4. Comparative performance on loop compatibility classification task for models trained using control dataset.

Model	Avg. ↓ Rank	Top ↑ 10	Top ↑ 30	Top ↑ 50
NLC (2 seconds)	13.25	0.57	0.75	1.00
Short Chunk CNN Res	1.0	1.00	1.00	1.00
DRT (4 Attn Layers)	1.0	1.00	1.00	1.00

Table 5. Comparative performance on loop ranking task for models trained using control dataset, using average rank and accuracy in the top-k positions across 100 queries

While all models show improved performance across both tasks, we perform another set of evaluations to determine if these on-paper performance gains are transferable to real-life production scenarios. Here, we evaluate these models, trained on the control set, against the test set generated by our proposed method — where pairs sharing the same tempo, key, and phases are analyzed for compatibility. These results are presented in Tables 6 and 7.

Model	Accuracy ↑	F1 Score ↑
NLC (2 seconds)	50.95	62.49
Short Chunk CNN Res	53.30	67.52
DRT (4 Attn Layers)	54.15	67.93

Table 6. Classification performance of models trained on control set (Table 4), but evaluated on our original test (Table 1) containing loops pairs with identical tempo and keys.

Model	Avg. ↓ Rank	Top ↑ 10	Top ↑ 30	Top ↑ 50
NLC (2 seconds)	50.7	0.13	0.31	0.50
Short Chunk CNN Res	39.4	0.00	0.30	0.85
DRT (4 Attn Layers)	35.8	0.14	0.40	0.67

Table 7. Ranking performance on our original test set using average rank and accuracy in the top-k positions.

While the models trained on the controlled dataset have better performance (Tables 4 and 5) than models trained on our dataset (Tables 2 and 3), they do not generalize for negative samples that are more in line with real-world production choices (Tables 6 and 7). This is because real-life comparison involves two loops that are identical in tempo,

key, and phase, without being reversed or subjected to random beat shifts. Since these models have not encountered such incompatible samples during training, their performance tends to degrade in the production setting.

5.3 Subjective Assessment

Study Methodology: We perform a subjective analysis using Apple loops from GarageBand to evaluate the effectiveness of our proposed method and demonstrate its applicability to high-quality production loops. Based on superior objective performance, three models are selected for this user study: NLC (2 seconds), Short Chunk CNN Res, and DRT (4 Attn Layers). Two variants of each model were included, the first trained on our proposed dataset, and the second trained on the control dataset. For this subjective assessment, we employed a methodology similar to that used by Zhao et al. [20]. We paired one query loop against 99 candidates, within the query loop’s genre, and formulated audio test clips by combining the query loop with the highest-ranked match. Each set contained eight query-target pairs: top matches proposed by each variant of the three models, a human musician-generated pair, and a randomly selected target loop to serve as a control group. Each audio sample was of equal length (10 seconds). A total of 6 such sets were created. The subjects were asked to rate each sample on a 5-point Likert scale according to the following criteria:

- **Seamlessness:** Naturalness of the loop combination.
- **Creativity:** Originality and inventive quality.

The study engaged a total of 37 participants. To qualify, participants were required to have a baseline engagement with music, defined as listening to at least five hours of music per week, to ensure sufficient exposure to music to provide informed feedback. Each survey participant listened to exactly three of the sets chosen at random (24 audio combinations, or 240 seconds of audio). To ensure diverse and representative survey respondents, we employed demographic filtering to include different ages, genders, and cultural backgrounds. The sequence of the presentation was randomized to eliminate any potential bias, and the origins of the pairs were not disclosed to participants.

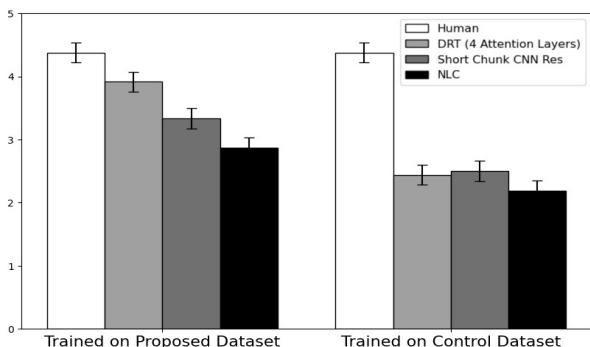


Figure 3. Subjective evaluation results for composition seamlessness computed using within-subject ANOVA.

Results and Analysis: Figures 3 and 4 display our findings from the subjective evaluation. The y-axis represents

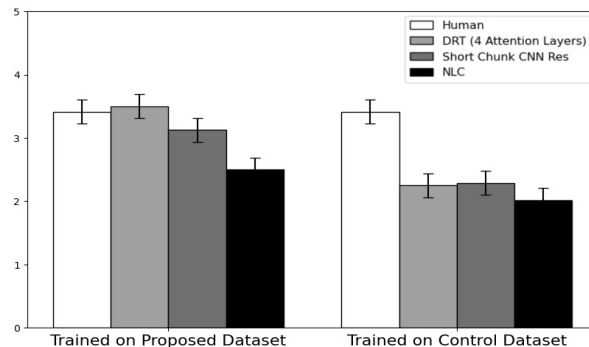


Figure 4. Subjective evaluation results for composition creativity computed using within-subject ANOVA.

the mean scores and the error bars denote the Standard Deviation calculated through a within-subject ANOVA [21]. Our model demonstrated superior performance over the control, achieving statistical significance ($p < 0.05$) for both measures. The proposed DRT architecture, trained on our dataset, surpassed other models by a significant performance difference ($p < 0.001$). The participant responses in the survey demonstrated high reliability, as evidenced by a Cronbach’s α of 0.812 [22]. Overall, the scores for our approach were on par with human music compositions.

6. CONCLUSION

We explored the vertical loop compatibility problem in music production. One major limitation to applying deep neural networks in this domain has been the absence of labeled datasets. We presented a novel self-supervised method for generating a large, high-quality, labeled dataset from a multi-track MIDI dataset, containing separate instrument tracks, and synthesizing them into audio to extract loops. This provides our training models with more granular information about each instrument across different genres and provides negative pairs with matching tempo, key, and phase to force models to learn true musical compatibility. A large dataset allows us to introduce and examine Transformer-based architectures. Our architecture employs a larger context window of ten seconds allowing a holistic input representation and consequently better compatibility prediction. Experimental results show that our method outperforms the previous state-of-the-art, achieving an 18.6% higher accuracy across multiple genres. Subjective assessments rate our model higher in seamlessly and creatively combining music loops.

Nevertheless, implementing Transformer architectures demands significant computational resources. Also, while AMU performs well in identifying incompatible pairs, it does not guarantee incompatibility and may contain leakage. Finally, synthesized audio from MIDI may not fully capture the richness of professionally recorded music. This could limit the model’s learning scope, especially regarding timbral and expressive nuances which otherwise may be important to learn. Future work may involve experimenting with more efficient architectures, collecting human-labeled datasets, and synthesizing MIDI using professional virtual instruments for better dataset quality.

7. REFERENCES

- [1] M. E. Davies, P. Hamel, K. Yoshii, and M. Goto, "Automashupper: Automatic creation of multi-song music mashups," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 12, pp. 1726–1737, August 2014.
- [2] G. Sorcinelli, "From garageband loop to grammy award: A look back at rihanna's "umbrella"," *Micro-Chop*, Oct 2016.
- [3] M. Goto, "Grand challenges in music information research," in *Multimodal Music Processing*, ser. Dagstuhl Follow-Ups, M. Müller, M. Goto, and M. Schedl, Eds. Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2012, vol. 3, pp. 217–226. [Online]. Available: <https://drops-dev.dagstuhl.de/entities/document/10.4230/DFU.Vol3.11041.217>
- [4] B.-Y. Chen, J. B. L. Smith, and Y.-H. Yang, "Neural loop combiner: Neural network models for assessing the compatibility of loops," *arXiv preprint arXiv:2008.02011*, 2020.
- [5] J. Huang, J. C. Wang, J. B. Smith, X. Song, and Y. Wang, "Modeling the compatibility of stem tracks to generate music mashups," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 1, May 2021, pp. 187–195.
- [6] M. E. P. Davies, P. Hamel, K. Yoshii, and M. Goto, "Automashupper: An automatic multi-song mashup system," in *International Society for Music Information Retrieval Conference*, 2013. [Online]. Available: <https://api.semanticscholar.org/CorpusID:28117>
- [7] C.-L. Lee, Y.-T. Lin, Z.-R. Yao, F.-Y. Lee, and J.-L. Wu, "Automatic mashup creation by considering both vertical and horizontal mashabilities," in *International Society for Music Information Retrieval Conference*, 2015. [Online]. Available: <https://api.semanticscholar.org/CorpusID:17802326>
- [8] K. Tsuzuki, T. Nakano, M. Goto, T. Yamada, and S. Makino, "Unisoner: An interactive interface for derivative chorus creation from various singing voices on the web," in *ICMC*, 2014.
- [9] G. Bernardes, M. Davies, and C. Guedes, "A hierarchical harmonic mixing method," in *Music Technology with Swing*, ser. Lecture Notes in Computer Science, M. Aramaki, M. Davies, R. Kronland-Martinet, and S. Ystad, Eds., vol. 11265. Cham: Springer, Cham, 2018, cMMR 2017. [Online]. Available: https://doi.org/10.1007/978-3-030-01692-0_11
- [10] G. Bernardes, D. Cocharro, M. Caetano, C. Guedes, and M. Davies, "A multi-level tonal interval space for modelling pitch relatedness and musical consonance," *Journal of New Music Research*, vol. 45, pp. 1–14, May 2016.
- [11] C. Maças, A. Rodrigues, G. Bernardes, and P. Machado, "Mixmash: A visualisation system for musical mashup creation," in *2018 22nd International Conference Information Visualisation (IV)*, 2018, pp. 471–477.
- [12] J. B. L. Smith, Y. Kawasaki, and M. Goto, "Unmixer: An interface for extracting and remixing loops," in *ISMIR*, 2019, pp. 824–831.
- [13] E. Manilow, G. Wichern, P. Seetharaman, and J. L. Roux, "Cutting music source separation some slakh: A dataset to study the impact of training data quality and quantity," in *2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2019, pp. 45–49.
- [14] C. Raffel, "Learning-based methods for comparing sequences, with applications to audio-to-midi alignment and matching," Ph.D. dissertation, PhD Thesis, 2016.
- [15] C. L. Krumhansl, "Cognitive foundations of musical pitch," 2001.
- [16] G. Bernardo and G. Bernardes, "Leveraging compatibility and diversity in computer-aided music mashup creation," *Personal and Ubiquitous Computing*, vol. 27, no. 5, pp. 1793–1809, 2023.
- [17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.
- [18] M. Won, K. Choi, and X. Serra, "Semi-supervised music tagging transformer," *arXiv preprint arXiv:2111.13457*, 2021.
- [19] M. Won, A. Ferraro, D. Bogdanov, and X. Serra, "Evaluation of cnn-based automatic music tagging models," *arXiv preprint arXiv:2006.00751*, 2020.
- [20] J. Zhao and G. Xia, "Accomontage: Accompaniment arrangement via phrase selection and style transfer," 2021.
- [21] H. Scheffe, *The Analysis of Variance*. John Wiley Sons, 1999, vol. 72.
- [22] L. J. Cronbach, "Coefficient alpha and the internal structure of tests," *Psychometrika*, vol. 16, no. 3, pp. 297–334, Sep 1951.

I CAN LISTEN BUT CANNOT READ: AN EVALUATION OF TWO-TOWER MULTIMODAL SYSTEMS FOR INSTRUMENT RECOGNITION

Yannis Vasilakis

Queen Mary University of London

i.vasilakis@qmul.ac.uk

Rachel Bittner

Spotify

rachelbittner@spotify.com

Johan Pauwels

Queen Mary University of London

j.pauwels@qmul.ac.uk

ABSTRACT

Music two-tower multimodal systems integrate audio and text modalities into a joint audio-text space, enabling direct comparison between songs and their corresponding labels. These systems enable new approaches for classification and retrieval, leveraging both modalities. Despite the promising results they have shown for zero-shot classification and retrieval tasks, closer inspection of the embeddings is needed. This paper evaluates the inherent zero-shot properties of joint audio-text spaces for the case-study of instrument recognition. We present an evaluation and analysis of two-tower systems for zero-shot instrument recognition and a detailed analysis of the properties of the pre-joint and joint embedding spaces. Our findings suggest that audio encoders alone demonstrate good quality, while challenges remain within the text encoder or joint space projection. Specifically, two-tower systems exhibit sensitivity towards specific words, favoring generic prompts over musically informed ones. Despite the large size of textual encoders, they do not yet leverage additional textual context or infer instruments accurately from their descriptions. Lastly, a novel approach for quantifying the semantic meaningfulness of the textual space leveraging an instrument ontology is proposed. This method reveals deficiencies in the systems' understanding of instruments and provides evidence of the need for fine-tuning text encoders on musical data.

1. INTRODUCTION

Multiclass classification has been a heavily researched topic in Music Information Retrieval (MIR) with many concrete applications such as genre, instrument and emotion recognition [1–4]. Despite the success of Deep (DL) systems for such tasks, recurring deficiencies persist among these problems. These are: (1) the limited availability of large-scale annotated datasets curated by experts, (2) the restricted capability of these systems to infer only a set of predefined classes.

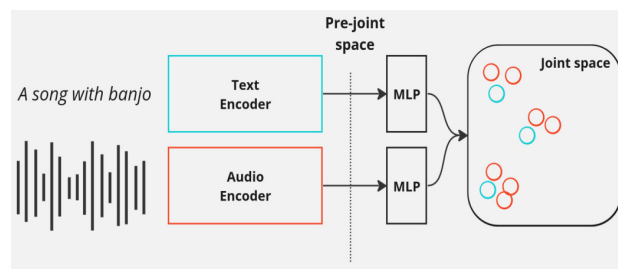


Figure 1: Figure of a pipeline for two-tower multimodal systems. A separate model for each modality is used and their individual representations are projected to a joint audio-text space through a Multi-Layer Perceptron (MLP). This enables direct comparison between audio and textual data. We refer to embeddings obtained before joint-space projection as pre-joint space embeddings.

Music is an ever-evolving art form and as a result, there is an inherent need to make these systems adaptable to new terms/classes [5–7] and infer task-agnostic representations that can be useful for a plethora of downstream tasks with representation learning [8–10]. Zero-shot learning (ZSL) is focused on estimating a classifier capable of inferring unseen, new classes without annotated examples [11–14]. ZSL is often achieved in either of two ways: (1) by decomposing each class into attributes and inferring unseen classes through their related attributes (e.g. genres decomposed into presence or absence of instruments [12]) or (2) using word embeddings from Language Models (LM) [12–14]. The success of contextualized Large Language Models (LLM) has driven the research community predominantly toward the second solution, as it doesn't require experts to define attributes and the mapping between classes and attributes [15–17].

As a result, ZSL for audio classification is primarily focused on connecting the audio and semantic representation spaces. This interconnection can happen in 2 main ways: (1) mapping the audio representations to text space [12, 13], or (2) mapping both of the spaces to a new, joint audio-text space [14, 18–20]. The systems of the second category are named two-tower multimodal systems, where pre-trained audio models and LMs are used as the audio and text encoders respectively. Representations obtained from each modality are then mapped to a joint audio-text space and systems are jointly optimized such



that the audio and text representation are close in the joint space (e.g. the phrase “A rock song track” is similar to the recording of a “rock” song). We will call such representations as embeddings from here on.

This work aims to better understand the properties of existing two-tower systems. We use instrument classification as a case-study to provide insights into the presence (or absence) of semantic properties in the audio, text or joint spaces in addition to reporting classification metrics. Concretely, we consider 3 systems: MusCALL [18], a CLAP [20,21] model trained on speech and music datasets, and a CLAP model trained on music data [21]. We evaluate the performance of these systems on instrument classification using the TinySOL dataset [22]. We would like to highlight that multimodal DL models typically excel at simple tasks and datasets like this.

Furthermore, a novel approach for quantifying the semantic meaningfulness of textual encoders for instrument recognition is proposed.

For reproducibility, our experiments are performed on open-source datasets and the code of our experiments is made publicly available¹, such that they can be reproduced.

2. RELATED WORK

2.1 Zero-shot transfer

ZSL focuses on estimating classifiers for novel, unseen classes without annotated examples. Two-tower systems are not primarily optimized for ZSL but due to the pre-trained textual encoder, novel words or phrases can be interpreted during inference. This property is known as zero-shot transfer (ZST) [23].

Side-information can be used in multiple ways that fall into two categories: (1) decomposing classes into shared attributes and (2) using LMs to represent this information as a text embedding. Despite its success, the first solution requires experts to effectively estimate the relevance of attributes and several classes and is a costly activity. As a result and due to the remarkable results obtained through contextualized LLMs, research has focused on the second option.

Generally, the methodology can be broken down into 3 components: (1) an audio encoder, (2) a textual encoder and (3) a projection to a common space. General purpose audio DL models that have been used as the audio encoder include VGGish [24], PANN [8], HTS-AT [9] and Audio Spectrogram Transformers [25]. For the textual encoder, distributional LMs like GloVe [26], Word2Vec [27] and contextualized LMs like BERT [28] have been thoroughly tested. As each modality produces heterogeneous representations, different methods of establishing comparability have been tested. This is predominantly achieved through projecting audio to text/semantic space [11–13] or a novel, joint audio-text space [18–20].

¹ https://github.com/YannisBilly/i_can_listen_but_cannot_read

2.2 Two-tower multimodal systems

Multimodal systems aim to represent data with additional knowledge from multiple modalities. Examples are audio combined with images [29], text [12, 30] or a combination thereof.

Two-tower multimodal systems focus on combining the textual and audio modalities by projecting them in a joint audio-text space. In that space, words that are relevant to a specific song will produce embeddings that will be close in terms of some similarity metric. An illustration of a two-tower system is presented in Figure 1. Information flowing through the audio encoder or textual encoder is referred to as the audio and textual branch respectively. We are also interested in the embeddings obtained through the encoders before projecting them into the joint audio-text space. We will call these the pre-joint spaces from now on.

The text used during training is usually a description of a song and will be referred to as a caption. The text used during inference will be referred to as a prompt.

MusCALL [18] combined a ResNET-50 for audio [31] with a Transformer for text encoding [32], optimized jointly over InfoNCE contrastive loss [33]. Additionally, a weighing mechanism based on caption-caption similarity was incorporated between negative audio-caption pairs. This is based on the premise that similar captions will be given to similar audio. The audio used is private but the training code is publicly available and used for this work.

MuLan [19] experimented with ResNET-50 as well as Audio Spectrogram Transformers [25] for audio encoding and a pretrained BERT model for the text branch. Both were jointly optimized over the Contrastive Multi-view Coding loss [34], which is a cross-modal extension of InfoNCE. Neither the data nor the code is available.

LAION-CLAP tested 6 different combinations of audio and text embedding models, the best one of which was HTS-AT with RoBERTa [35]. The latter is the one that will be used in this paper. The LAION-Audio-630k dataset was formed by combining AudioCaps, CLotho and Audioset.

Generally, research for two-tower systems is limited to testing different combinations of audio and text encoders, optimized jointly over a form of contrastive loss and modality fusion. We believe that closer inspection of their embeddings and evaluation protocol is needed.

3. EVALUATION OF TWO-TOWER SYSTEMS

3.1 Dataset and models

We use the TinySOL dataset which contains 2913 audio clips with a single note played from a single instrument out of a set of 14 instrument classes. This dataset has been chosen as it has consistent recording settings without noise, it is a simple dataset for instrument recognition and finally, confounding factors (compression, sampling rate etc) are minimized. We consider 3 models in total:

1. **Music/Speech CLAP:** [20, 21] A CLAP-based model trained on music/speech data²

² music_speech_epoch_15_esc_89.25.pt

2. **Music CLAP**: [21] A CLAP-based model trained on music data³
3. **MusCALL**: A version of [18], retrained on music data⁴

We use the two pretrained CLAP systems provided by LAION⁵. For this work, the original MusCALL implementation was retrained from scratch, as both the data and trained models used in the original paper are not publicly available. Instead, we train on the LPMusicCaps-MTT [36] dataset, which is built by leveraging the audio and 188 tags from Magna Tag A Tune [37] to artificially generate captions through a GPT-3.5 model. The audio is resampled to 44.1 and 16 KHz for CLAP and MusCALL respectively, and the pre-processing steps described in their respective code repositories are followed.

3.2 Zero-shot transfer for instrument classification

Given an unseen audio segment x^* , a text label l^* and a two-tower system $f(x)$, we want to model the likelihood $P(l^*|x^*)$ based on the embeddings provided by $f(x)$. In the general case, $f \mapsto \mathbb{R}^F$ is a function that represents a two-tower system and maps audio or text information to a joint audio-text space, where F is the dimension of the joint space. Also, let $\delta : (\mathbb{R}^F \times \mathbb{R}^F) \rightarrow \mathbb{R}$ be a function that measures similarity between joint space embeddings. In this approach, we model the $P(l^*|x^*)$ based on δ , as in:

$$P(l^*|x^*) \propto \delta(f(x^*), f(l^*)) \quad (1)$$

Multiclass classification attributes the most probable class to each recording and equivalently, the one that has the maximum likelihood. Given our approximation, the output class for each recording is:

$$c^* = \operatorname{argmax}_{c \in C} \delta(f(x^*), f(c)) \quad (2)$$

for $c \in C$, where $C = \{c_1, c_2, \dots, c_N\}$ is the set of classes that we are interested in.

In our work, the embedding similarity function δ is the cosine similarity:

$$\delta(e_1, e_2) = \frac{e_1 \cdot e_2}{\|e_1\| \cdot \|e_2\|} \quad (3)$$

where $\|\cdot\|$ is the L_2 norm and $e_i \in \mathbb{R}^F$ are embeddings.

3.3 Experiment 1: Are two-tower systems context dependent?

Two-tower systems are typically not trained on single words, but rather longer prompts. As a result, the embedding produced for a single-word text label (e.g. “guitar”) can be very different from the embedding for a longer prompt, with additional context (e.g. “a guitar track”). When using two-tower models for classification, the class label can be wrapped in a prompt such as “A <label>

track” [18], to better match the training distribution. Methods to introduce stochasticity in the prompts used during training have been empirically proven to lead to more robust results [38]. Retraining the systems and testing different ways of augmenting captions used for training is left for future work, but works in image-text [39] and video-text [40] two-tower systems provide some evidence for their usefulness.

The impact of different approaches for giving additional context to a single-word text label during inference has not been well-explored. We explore the prompt sensitivity of each system by slightly changing the text prompt used for zero-shot classification in order to better understand to which extent these systems leverage contextual information. As far as we are aware, we are the first to evaluate the use of different types of prompts for two-tower systems during inference. Specifically, we evaluate 3 systems against 6 different prompts:

1. MusCALL prompt: “A <label> track”
2. Generated definition: “The <label> is a ...”
3. Generated definitions without label words: “The <removed> is a ...”
4. Label word with random context: “<label> <randomly selected lorem ipsum segment>”
5. Musically informed #1: “This is a recording of a <label>”
6. Musically informed #2: “Solo musical instrument sound of a <label>”

The first prompt proposed is the prompt that was used in MusCALL. The second prompt is generated using GPT-3.5 [41]. The third prompt is the same as the second but we removed all instances of the label itself to evaluate the influence of the context on its own. To evaluate if the systems are sensitive to specific words and to further evaluate if the context is useful, the fourth prompt adds random words alongside the label. Lastly, we test 2 musically informed prompts.

As a first metric, we consider Top-k accuracy with $k = \{1, 2, 3\}$. We calculate the cosine similarity between each recording and instrument prompt and sort them. We assign zero-shot class labels as described in Section 3.2 and check if the true label is present in the top-k assigned class labels. Furthermore, we calculate Receiver Operating Characteristic and Precision-Recall Area Under the Curve (ROC-AUC and PR-AUC respectively) following [42].

Figure 2 presents the zero-shot instrument recognition results for the three models across the 6 prompts, as well as the audio-only alternatives that will be described in Section 3.4. Despite the focus on music data, Music CLAP doesn’t display very different results from Music/Speech CLAP. While music-specific systems are generally expected to perform better, this is not the case for two-tower systems. This might be an indication that music requires special treatment, as the metrics approach the state of the art in audio-text [21] and image-text [43] two-tower systems.

³ music_audioset_epoch_15_esc_90.14.pt

⁴ <https://github.com/ilaria-manco/muscall>

⁵ <https://github.com/LAION-AI/CLAP>

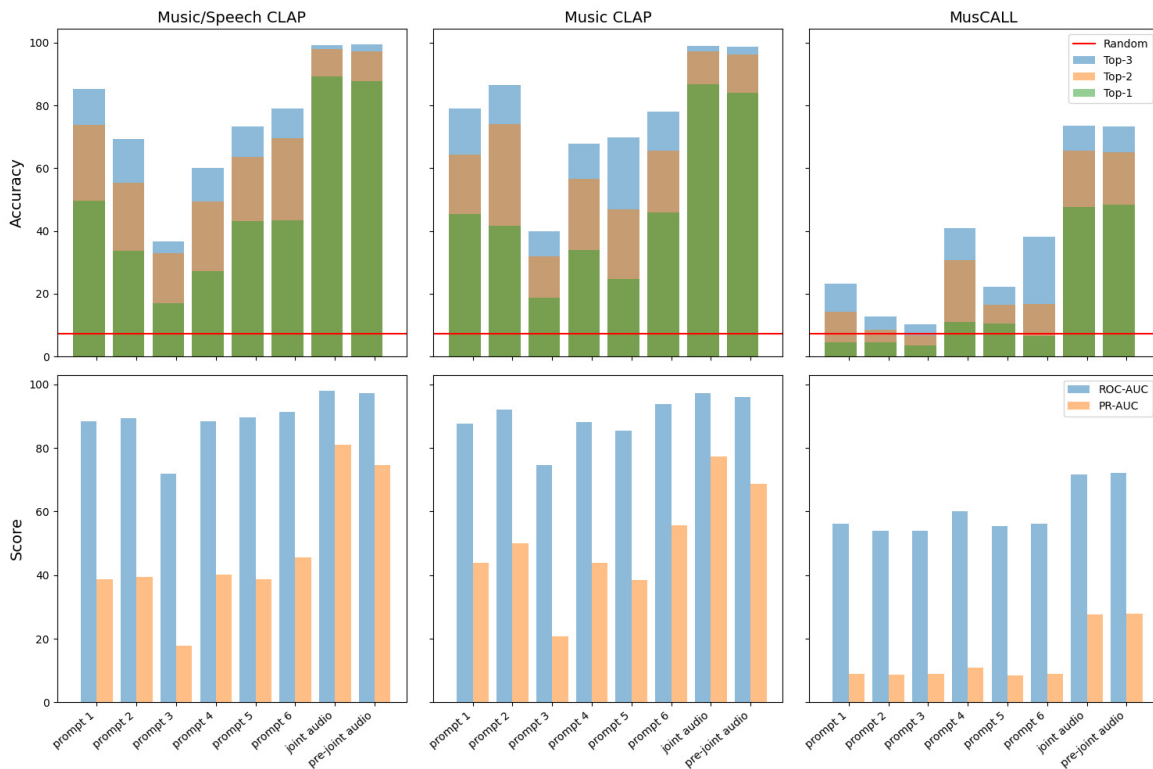


Figure 2: Metrics for 6 textual prompts (See Section 3.3), 2 audio based label embeddings (See Section 3.4) and the 3 two-tower multimodal systems. The top row contains top-1 through top-3 accuracy and the bottom ROC-AUC and PR-AUC. The red line represents random choice.

Top-1 accuracy is worse than random for 4 out of 6 textual prompts for MusCALL. This might be caused by the small size of training data used, the absence of instrument-specific captions or their underrepresentation in the captions used, as well as the absence of single-note recordings in LPMusicCaps-MTT. While the metrics are low for MusCALL in most of the cases, a relatively large performance is still evident for the audio-only scenario. This implies that the problem lies in the audio-text alignment or the text branch.

The performance of CLAP models seems to be heavily correlated with the instrument labels themselves. Removing the label from definitions provides evidence that relevant context cannot be leveraged properly. Also, using musically informed prompts doesn't always result in greater or even comparable results. Specifically, top-1 accuracy drops when using the second musically informed prompt for CLAP models, despite the prompt being a more precise description of what is occurring in the audio.

These results suggest that CLAP models do not leverage extra context in the input prompt effectively. Both models performed worse when using relevant context without the instrument word, suggesting that the textual encoders put a lot more emphasis on the presence of specific words rather than the meaning of the prompts themselves. In addition, using a generic prompt provided better results than a musically informed one in most cases. Furthermore, any kind of context added at the prompts seems to harm the performance in most of the cases and provide more evidence that

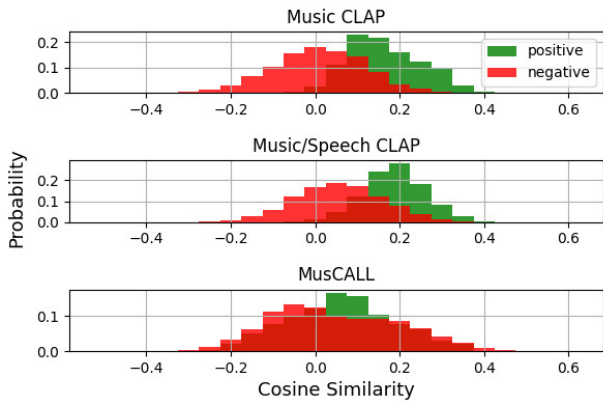
the model's text encoder cannot properly decompose the sentence to its constituents and use these semantically. Despite this observation, using definitions (prompt 2) seems promising for Music CLAP and for every metric apart from top-1 accuracy.

In the following experiments, we will consider only the "MusCALL prompt" as it leads to the highest top-1 accuracy, when model accuracy surpasses random choice.

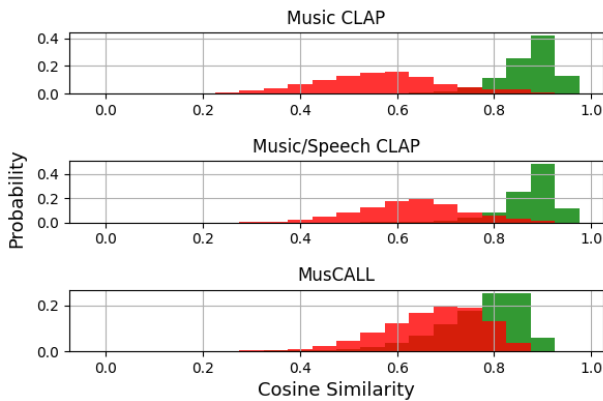
3.4 Experiment 2: Inspecting the cosine similarity distributions

As a next experiment, we calculate the cosine similarity between the joint space embeddings of each recording from TinySOL and the MusCALL prompt for each instrument, then compare the similarities of positive pairs vs negative pairs. We define a positive pair as an audio-label pair where the label corresponds to the instrument in the recording, and the negative pairs as all other pairs. Figure 3a presents histograms of similarities for positive and negative pairs when using text prompts. If the audio-text coherence is good, positive and negative histograms should be well separated.

Positive and negative similarity distributions overlap greatly, as can be seen in Figure 3a. As a result, retrieval is far from optimal. Fundamentally, a caption is a multi-faceted sentence. We suspect that treating a sentence as only one embedding point (mean of word embeddings) is fundamentally problematic and greatly hinders the semantic properties of the joint space. A hypothesis that needs



(a) Histogram of cosine similarity between TinySOL data and MusCALL prompts in joint audio-text space.



(b) Histogram of cosine similarity between TinySOL audio data and the mean of intra-class embeddings in joint audio-text space.

Figure 3: Histograms of audio and label embeddings for positive and negative pairs. When using textual prompts (a), the alignment is problematic, as can be seen from the overlap between positive and negative distributions.

testing is that by using composite sentences, a model cannot properly infer the relative embeddings of the sentence constituents.

To further evaluate if the audio encoder produces meaningful representations, we use the mean of intra-class song embeddings as the label embedding. This label embedding takes the role of the prompt embedding in the previous experiment. We generate the embeddings in joint audio-text space for each song. Then, we collect the songs that belong to k -th class c_k and estimate the mean of the embeddings. The latter serves as the optimal embedding that the text label would have to be mapped to in order to maximize performance and will be referred to as the “audio-only” label. Note that this embedding is only optimal in the case of TinySOL data.

The resulting histograms for positive and negative pairs are shown in Figure 3b. They are well separated, indicating that the audio-encoder itself produces meaningful, separable embeddings.

Audio embeddings seem to be of good quality before and after the projection to the joint audio-text space, as the metrics are almost equal before and after projection in Fig-

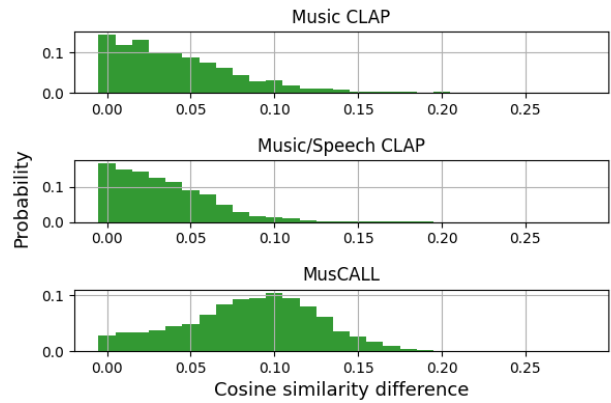


Figure 4: The histogram of top-2 class similarities for every song in TinySOL. The CLAP models tend to be not very confident while the metrics are greater than the over-confident MusCALL with the worst metrics.

ure 2. The metrics almost double when using any audio-only labels, which further provides evidence that the problem resides in the text branch, or joint-space projection and there remains a large performance gap to bridge.

3.5 Experiment 3: How confident are two-tower systems in their prediction?

We calculate the histogram of the difference between the top-2 candidate classes for each recording [44] to quantify the classification confidence. The similarity between each audio and instrument embeddings is estimated and they are sorted in descending order. The difference between top-2 similarities for each song is then calculated and a histogram of that difference is plotted in Figure 4.

MusCALL seems to be overly confident in its prediction, which is unwarranted given the metrics reported. The opposite can be stated for CLAP models, where despite their better performance, the difference has a median value of 0.05-0.08.

3.6 Experiment 4: Quantitative evaluation of the text branch

While there are datasets that can be used to quantify the semantic properties and/or quality of a LM, there isn’t one that focuses on music. To overcome this lack of text data for the case of instrument recognition, we can utilize instrument ontologies, which encompass semantic similarity of instruments at multiple levels. We propose to leverage them to quantify the semantic similarity between different instruments and instrument families. In this experiment we use the instrument ontology by Henry Doktorski⁶ (HDIO). As every instrument ontology has its limitations [45], repeating the same experiment with other ontologies is left for future work.

We extract the tree based on HDIO and form every possible triplet of TinySOL instrument labels in the tree for a total of $364 = \binom{14}{3}$ combinations of positive word

⁶ <https://free-reed.net/>

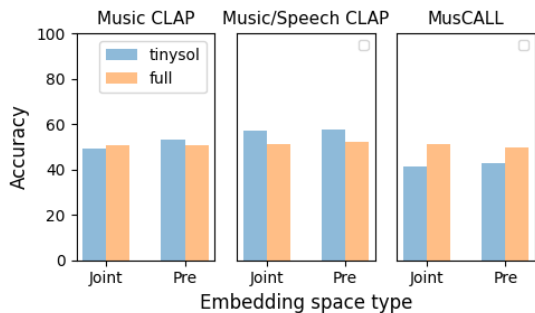


Figure 5: Semantic meaningfulness quantification leveraging Henry Doktorski’s instrument ontology. We evaluated the systems over valid triplets obtained through TinySOL labels, as well as every available triplet obtained from the ontology’s labels. Accuracy ranges from 49-59% which stresses that the models do not properly understand musical instruments in depth.

pairs without repetition. The triplets are of the form (*<anchor>*, *<positive>*, *<negative>*) where the *<anchor>* label has to be more semantically similar according to HDIO to the *<positive>* than the *<negative>*, e.g (“violin”, “violoncello”, “trumpet”). Subsequently, every (*<anchor>*, *<positive>*) pair that is linked through the root node of HDIO is excluded. The number of remaining triplets is 273. As a way to quantify semantic meaningfulness with respect to musical instruments, we calculate cosine similarity between the (*<anchor>*, *<positive>*) and (*<anchor>*, *<negative>*) pairs for each system. Triplets for which the similarity is higher for the first pair than the second are considered “correct”, and triplets where this is not the case are considered “incorrect”. We compute the accuracy score as the percentage of correct triples.

We repeat this procedure with every valid triplet from the full ontology, as opposed to just using the instrument labels appearing in TinySOL. This gives us $\approx 443k$ triplets.

The accuracy for triplets from TinySOL and the full HDIO ontology are both presented in Figure 5. We see that half of the triplets are “incorrect” and this means that abstract semantic relations between instruments are not effectively captured in the textual branch, indicating a need for fine-tuning textual encoders on music related data. Note that the accuracy is roughly the same as we would obtain by creating arbitrary triplets, though it is important to highlight that several instruments and instrument categories are words that are not frequently used in English. Closer examination of the validity and usefulness of specific triplet cases (e.g. “stringed”, “plucked”, “violin”) is left for future work.

3.7 Experiment 5: Does joint space mapping introduce noise?

To further examine the origins of the problematic embedding alignment, we repeat zero-shot evaluation with audio-only labels described in Sections 3.4 and semantic mean-

ingfulness evaluation described in 3.6 with the embeddings in the audio space and text space before the joint audio-text space mapping.

A minor performance increment can be seen when using the joint embedding instead of the pre-joint audio embedding, as can be seen in the last two columns of Figure 2, apart from MusCALL where the metrics remain almost the same. We believe that the reduction in dimensionality of the joint space compared to the separate spaces is the underlying cause of these increments.

On the other hand, the accuracy based on HDIO remains the same, except for a decrement observed for Music CLAP and TinySOL subset of HDIO triplets, as can be seen in Figure 5. This could be an indication that the MLP can effectively map knowledge to the joint space. This is a further hint that potentially the problem lies in the LM used and fine-tuning might be needed to enforce musical semantics to be better represented.

4. CONCLUSIONS AND FUTURE WORK

In this paper, we evaluated 3 two-tower multimodal systems for instrument classification. We provided a zero-shot classification analysis and an elaborate evaluation of the audio and text embeddings in the pre-joint and joint audio-text spaces. We also proposed a novel way to quantify the semantic meaningfulness of text embeddings based on triplets derived from an instrument ontology.

Generally, experiments showed that audio encoders are of good quality and hence, the alignment issue might be traced back to the text branch and/or the joint audio-text space mapping. Therefore, a solution could be to freeze the audio encoder and map the text information to audio space. Also, further attention to modality imbalance [46] can be placed with weighing in negative and positive examples [18, 47–50]. Additionally, to avoid sensitivity towards instrument labels and the inability to leverage context, we propose to use text augmentation over captions or masking/removing the words from them. It is important to state that the relation between sentence and word embeddings is not as straightforward as with bag-of-words Language Models [51] and as a result, the way to utilize captions or put additional emphasis on their constituents have to be further tested.

As a result, using two-tower systems might not be very useful for multi-class scenarios, given the large overlap between positive and negative histograms of cosine similarities shown in our experiments. We believe that it is essential for a music terminology similarity corpus to be established. The benefits will be two-fold: (1) it will provide a useful way of quantifying the semantic meaningfulness of the textual branch for two-tower model and (2) it can serve as a baseline to quantify the need for music-informed fine-tuning. Last but not least, genre and emotion ontologies can be used to further evaluate the semantic meaningfulness of language models.

5. REFERENCES

- [1] N. Ndou, R. Ajoodha, and A. Jadhav, "Music genre classification: A review of deep-learning and traditional machine-learning approaches," in *2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS)*, 2021, pp. 1–6.
- [2] D. Han, Y. Kong, J. Han, and G. Wang, "A survey of music emotion recognition," *Frontiers of Computer Science*, vol. 16, 2022.
- [3] M. Won, J. Spijkervet, and K. Choi, *Music Classification: Beyond Supervised Learning, Towards Real-world Applications*. <https://music-classification.github.io/tutorial>, November 2021.
- [4] Z. Fu, G. Lu, K. M. Ting, and D. Zhang, "A survey of audio-based music classification and annotation," *IEEE Transactions on Multimedia*, vol. 13, no. 2, pp. 303–319, 2011.
- [5] J. C. Lena and R. A. Peterson, "Classification as culture: Types and trajectories of music genres," *American Sociological Review*, vol. 73, no. 5, pp. 697–718, 2008.
- [6] A. van Venrooij and V. Schmutz, "Categorical ambiguity in cultural fields: The effects of genre fuzziness in popular music," *Poetics*, vol. 66, pp. 1–18, 2018.
- [7] R. Hennequin, J. Royo-Letelier, and M. Moussallam, "Audio based disambiguation of music genre tags," in *International Society for Music Information Retrieval Conference*, 2018.
- [8] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley, "PANNs: Large-scale pretrained audio neural networks for audio pattern recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2880–2894, 2019.
- [9] K. Chen, X. Du, B. Zhu, Z. Ma, T. Berg-Kirkpatrick, and S. Dubnov, "HTS-AT: A hierarchical token-semantic audio transformer for sound classification and detection," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, 2022.
- [10] L. Wang, P. Luc, Y. Wu, A. Recasens, L. Smaira, A. Brock, A. Jaegle, J.-B. Alayrac, S. Dieleman, J. Carreira, and A. van den Oord, "Towards learning universal audio representations," *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4593–4597, 2021.
- [11] H. Xie, O. J. Räsänen, and T. Virtanen, "Zero-shot audio classification with factored linear and nonlinear acoustic-semantic projections," *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 326–330, 2020.
- [12] J. Choi, J. Lee, J. Park, and J. Nam, "Zero-shot learning for audio-based music classification and tagging," in *International Society for Music Information Retrieval Conference*, 2019.
- [13] H. Xie and T. Virtanen, "Zero-shot audio classification via semantic embeddings," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 1233–1242, 2020.
- [14] H. Xie and V. Tuomas, "Zero-shot audio classification based on class label embeddings," *2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pp. 264–267, 2019.
- [15] S. Pratt, I. Covert, R. Liu, and A. Farhadi, "What does a platypus look like? generating customized prompts for zero-shot image classification," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2023, pp. 15 691–15 701.
- [16] Y. Ge, J. Ren, A. Gallagher, Y. Wang, M.-H. Yang, H. Adam, L. Itti, B. Lakshminarayanan, and J. Zhao, "Improving zero-shot generalization and robustness of multi-modal models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 11 093–11 101.
- [17] S. Goyal, A. Kumar, S. Garg, Z. Kolter, and A. Raghunathan, "Finetune like you pretrain: Improved finetuning of zero-shot vision models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 19 338–19 347.
- [18] I. Manco, E. Benetos, E. Quinton, and G. Fazekas, "Contrastive audio-language learning for music," in *Proceedings of the 23rd International Society for Music Information Retrieval Conference (ISMIR)*, 2022.
- [19] Q. Huang, A. Jansen, J. Lee, R. Ganti, J. Y. Li, and D. P. W. Ellis, "MuLan: A joint embedding of music audio and natural language," in *International Society for Music Information Retrieval Conference*, 2022.
- [20] B. Elizalde, S. Deshmukh, M. A. Ismail, and H. Wang, "CLAP: learning audio concepts from natural language supervision," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.
- [21] Y. Wu, K. Chen, T. Zhang, Y. Hui, T. Berg-Kirkpatrick, and S. Dubnov, "Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, 2023.
- [22] C. Emanuele, D. Ghisi, V. Lostanlen, F. Lévy, J. Fineberg, and Y. Maresz, "TinySOL: An audio dataset of isolated musical notes (5.0)," 2020.

- [23] X. Zhai, X. Wang, B. Mustafa, A. Steiner, D. Keysers, A. Kolesnikov, and L. Beyer, “LiT: Zero-shot transfer with locked-image text tuning,” *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18 102–18 112, 2021.
- [24] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *CoRR*, vol. abs/1409.1556, 2014.
- [25] Y. Gong, Y.-A. Chung, and J. Glass, “AST: Audio Spectrogram Transformer,” in *Proc. Interspeech 2021*, 2021, pp. 571–575.
- [26] J. Pennington, R. Socher, and C. Manning, “GloVe: Global vectors for word representation,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, A. Moschitti, B. Pang, and W. Daelemans, Eds. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1532–1543.
- [27] T. Mikolov, K. Chen, G. S. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” in *International Conference on Learning Representations*, 2013.
- [28] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in *North American Chapter of the Association for Computational Linguistics*, 2019.
- [29] D. Dogan, H. Xie, T. Heittola, and T. Virtanen, “Zero-shot audio classification using image embeddings,” *2022 30th European Signal Processing Conference (EUSIPCO)*, pp. 1–5, 2022.
- [30] X. Du, Z. Yu, J. Lin, B. Zhu, and Q. Kong, “Joint music and language attention models for zero-shot music tagging,” in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 1126–1130.
- [31] K. He, Y. Wang, and J. Hopcroft, “A powerful generative model using random weights for the deep image representation,” in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, ser. NIPS’16. Red Hook, NY, USA: Curran Associates Inc., 2016, p. 631–639.
- [32] A. Vaswani, N. M. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Neural Information Processing Systems*, 2017.
- [33] A. van den Oord, Y. Li, and O. Vinyals, “Representation learning with contrastive predictive coding,” *CoRR*, vol. abs/1807.03748, 2018. [Online]. Available: <http://arxiv.org/abs/1807.03748>
- [34] Y. Tian, D. Krishnan, and P. Isola, “Contrastive multiview coding,” in *European Conference on Computer Vision*, 2019.
- [35] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, “RoBERTa: A robustly optimized BERT pretraining approach,” *CoRR*, vol. abs/1907.11692, 2019.
- [36] S. Doh, K. Choi, J. Lee, and J. Nam, “LP-MusicCaps: LLM-based pseudo music captioning,” vol. abs/2307.16372, 2023.
- [37] E. Law, K. West, M. I. Mandel, M. Bay, and J. S. Downie, “Evaluation of algorithms using games: The case of music tagging,” in *International Society for Music Information Retrieval Conference*, 2009.
- [38] S. Doh, M. Won, K. Choi, and J. Nam, “Toward universal text-to-music retrieval,” in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023.
- [39] L. Zhen, P. Hu, X. Wang, and D. Peng, “Deep supervised cross-modal retrieval,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 10 386–10 395.
- [40] X. Wang, B. Ke, X. Li, F. Liu, M. Zhang, X. Liang, and Q. Xiao, “Modality-balanced embedding for video retrieval,” in *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR ’22. New York, NY, USA: Association for Computing Machinery, 2022, p. 2578–2582.
- [41] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, “Language models are few-shot learners,” in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 1877–1901.
- [42] M. Won, A. Ferraro, D. Bogdanov, and X. Serra, “Evaluation of cnn-based automatic music tagging models,” in *Proc. of 17th Sound and Music Computing*, 2020.
- [43] M. J. Mirza, L. Karlinsky, W. Lin, H. Possegger, M. Kozinski, R. Feris, and H. Bischof, “Lafter: Label-free tuning of zero-shot classifier using language and unlabeled image collections,” in *Advances in Neural Information Processing Systems*, A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., vol. 36. Curran Associates, Inc., 2023, pp. 5765–5777.
- [44] C. C. Aggarwal, *Data Classification: Algorithms and Applications*, 1st ed. Chapman & Hall/CRC, 2014.

- [45] Şefki Kolozali, M. Barthez, G. Fazekas, and M. B. Sandler, “Knowledge representation issues in musical instrument ontology design,” in *International Society for Music Information Retrieval Conference*, 2011.
- [46] X. Wang, B. Ke, X. Li, F. Liu, M. Zhang, X. Liang, Q.-E. Xiao, and Y. Yu, “Modality-balanced embedding for video retrieval,” *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2022.
- [47] D. Oneață and H. Cucu, “Improving multimodal speech recognition by data augmentation and speech representations,” *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 4578–4587, 2022.
- [48] K. Margatina, G. Vernikos, L. Barrault, and N. Aletras, “Active learning by acquiring contrastive examples,” in *Conference on Empirical Methods in Natural Language Processing*, 2021.
- [49] J. Choi, S. Jang, H. Cho *et al.*, “Towards proper contrastive self-supervised learning strategies for music audio representation,” in *2022 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2022, pp. 1–6.
- [50] S. Ma, Z. Zeng, D. McDuff, and Y. Song, “Learning audio-visual representations with active contrastive coding,” *CoRR*, vol. abs/2009.09805, 2020. [Online]. Available: <https://arxiv.org/abs/2009.09805>
- [51] M. Alian and A. Awajan, “Factors affecting sentence similarity and paraphrasing identification,” *Int. J. Speech Technol.*, vol. 23, no. 4, p. 851–859, dec 2020.

STREAMING PIANO TRANSCRIPTION BASED ON CONSISTENT ONSET AND OFFSET DECODING WITH SUSTAIN PEDAL DETECTION

Weixing Wei¹ Jiahao Zhao¹ Yulun Wu² Kazuyoshi Yoshii³

¹Graduate School of Informatics, Kyoto University, Japan

²School of Computer Science and Technology, Fudan University, China

³Graduate School of Engineering, Kyoto University, Japan

{wei.weixing.23w, zhao.jiahao.56h}@st.kyoto-u.ac.jp, yoshii.kazuyoshi.3r@kyoto-u.ac.jp

ABSTRACT

This paper describes a streaming audio-to-MIDI transcription method that can sequentially translate a piano recording into a sequence of note-on and note-off events. The sequence-to-sequence learning nature of this task may call for using a Transformer model, which has been used for offline transcription and could be extended for streaming transcription with a causal restriction of the attention mechanism. We assume that the decoder of this model suffers from the performance limitation. Although time-frequency features useful for onset detection are considerably different from those for offset detection, the single decoder is trained to output a mixed sequence of onset and offset events without guarantee of the correspondence between the onset and offset events of the same note. To overcome this limitation, we propose a streaming encoder-decoder model that uses a convolutional encoder aggregating local acoustic features, followed by an autoregressive transformer decoder detecting a variable number of onset events and another decoder detecting the offset events of the active pitches with validation of the sustain pedal at each time frame. Experiments using the MAESTRO dataset showed that the proposed streaming method performed comparably with or even better than the state-of-the-art offline methods while significantly reducing the computational cost.

1. INTRODUCTION

Automatic music transcription (AMT) is a central topic in the field of music information retrieval (MIR), which refers to converting a music recording into a symbolic musical score (MusicXML format) or a piano-roll representation (MIDI format) [1]. It has remarkably been improved with the technical progress of deep learning techniques and the public availability of large-scale music datasets. In this paper, we focus on streaming audio-to-MIDI AMT because it remains relatively unexplored unlike streaming automatic

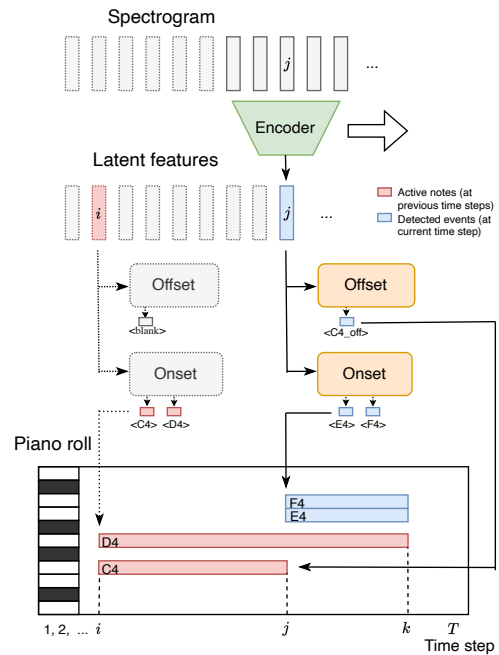


Figure 1. An overview of the proposed streaming audio-to-MIDI piano transcription method aware of onset-offset correspondence.

speech recognition (ASR) [2–4] and forms the basis of real-time music applications such as performance evaluation and interactive jam session. The previous research in [5] applied auto-regressive convolutional recurrent neural network (CRNN) frame-by-frame for piano transcription. The auto-regressive CRNN model can be easily adapted for the online scenario [6]. But the transcription performance for note offsets still has significant room for improvement.

Inspired by the sequence-to-sequence learning for ASR, many studies on AMT have recently attempted to use the Transformer [7] by serializing the polyphonic information of the estimation target [8, 9]. AMT is essentially different with ASR in a sense that the onsets, durations, and pitches of musical notes should be estimated, while the temporal information of output tokens (e.g., words and characters) is not considered in ASR. For audio-to-MIDI piano transcription, one may define the input and output of the Transformer as a sequence of raw audio features (e.g., mel and constant-Q spectrograms) and a sequence of note-on

and note-off events sorted in time and pitch, respectively. The performance of this naive approach, however, is potentially limited. Despite the significant differences in features needed for detecting onsets and offsets, the Transformer decoder estimates these events in a mixed manner. In addition, the correspondence between the onset and offset events of the same note is not guaranteed.

For streaming AMT, one can use the *causal* Transformer that restricts the self-attentive region to a certain number of past frames, which could reduce the computational cost of the basic self-attention mechanism that increases quadratically with the input length. Nonetheless, due to the strong coupling between note events, Transformer-based transcription methods often underperform the state-of-the-art frame-level methods [10, 11], especially in offset detection and velocity estimation.

To overcome these limitations, we propose a streaming audio-to-MIDI piano transcription method based on a novel encoder-decoder architecture (Fig. 1). The encoder is implemented with a convolutional neural network (CNN) that sequentially aggregates latent features from local regions of an input piano recording. The two Transformer decoders that operate framewise are then separately used for detecting a variable number of onset events and offset events for the active pitches with guarantee of onset-offset correspondence. For further improvement, the offset decoder is trained to judge the activation of the sustain pedal in a way of multitask learning.

The main contribution of this study is to develop an efficient streaming encoder-decoder model and pave a way for interactive and responsive applications based on real-time music transcription. We experimentally show that our method performs comparably with a state-of-the-art offline transcription method and outperforms existing sequence-to-sequence transcription methods.

2. RELATED WORK

This section reviews related work on automatic music transcription and sequence-to-sequence transcription.

2.1 Automatic Piano Transcription

Automatic piano transcription (APT) is the most popular form of AMT. Early methods rely on handcrafted features and rule-based algorithms [12–15], while modern methods use deep learning models such as CNNs [16–19], recurrent neural networks (RNNs) [20, 21], and transformers [22, 23]. In APT, the framewise transcription has still been the mainstream approach due to its superior performance and accuracy [10, 24]. In this approach, audio features such as short-time Fourier transform (STFT) spectrograms are mapped to a binary matrix of dimensions $T \times N$ indicating the presence of pitches over time frames, where T represents the number of frames and N the number of pitches. Early transcription methods, mostly based on CNNs, perform comparably at the frame level but underperform in term of note-level.

Onsets and Frames [19] is a major breakthrough in APT

that learns to sequentially predict note onsets and pitches in a multitask framework. To improve the performance, a music language model (MLM) based on a bidirectional long short-term memory (BiLSTM) network is used for modeling the temporal dependency of musical notes. This study has triggered many extensions. Kong et al. [25], for example, proposed a high-resolution piano transcription (HPT) model that simultaneously deals with onset, offset, velocity, and frame prediction tasks. The predicted velocities are used as conditional information to predict onsets, and the predicted onsets and offsets are used to predict frame-wise pitches, forming a hierarchical structure.

Our previous work [24] proposed HPPNet that uses harmonic dilated convolution for constant-Q transform (CQT) spectrograms and an enhanced frequency grouped LSTM (FG-LSTM) as a MLM. This model exhibits improved performance in both frame-level and note-level predictions. To capture long-term temporal and spectral dependencies, Toyama et al. [10] proposed a two-level hierarchical frequency-time transformer (hFT-Transformer) and achieved the state-of-the-art performance on the prediction of note with offset and velocity.

2.2 Sequence-to-Sequence Transcription

Sequence-to-sequence models are able to learn a mapping between input and output sequences of variable lengths and have actively been investigated in many fields such as natural language processing (NLP) and automatic speech recognition (ASR). Such models have recently been implemented with the Transformer or the self-attention mechanism due to its excellent performance. Awiszus et al. [26], for example, proposed a piano transcription model based on an LSTM and a Transformer for frame-level multi-pitch estimation. The performance of this method, however, is limited due to the lack of training data and using improper relative time shifts.

Inspired by this study, Hawthorne et al. [8] proposed a note-level piano transcription model that uses Transformer encoder and decoder in a way similar to the T5 model [27]. The encoder extracts latent features from an input spectrogram and the decoder refers to the input in an autoregressive manner, and the token with the highest probability is selected at each frame. This method achieved promising performance on the MAESTRO dataset and was later extended to multi-track music transcription [9]. However, this sequence-to-sequence transcription method still faces limitations. It encodes all types of note events and absolute time location of each event into a single sequence. This increases the complexity of sequence-to-sequence transformation and also constrains the length of the input sequence.

3. PROPOSED METHOD

This section explains the proposed method of streaming audio-to-MIDI piano transcription based on a single encoder and onset and offset decoders.

3.1 Streaming Transcription

As shown in Algorithm 1, the model takes a spectrogram $\mathbf{X} \in \mathbb{R}^{T \times F_i}$ as input, where T represents the number of frames and F_i represents the number of frequency bins. It outputs an onset sequence list \mathbf{Y} and an offset sequence list $\bar{\mathbf{Y}}$, where each element \mathbf{Y}_t in \mathbf{Y} represents the detected onsets sequence of frame t with sequence length k_t , and each element $\bar{\mathbf{Y}}_t$ in $\bar{\mathbf{Y}}$ represents the detected offsets sequence with sequence length n_t in frame t .

The model consists of one encoder and two decoders (Fig. 2). The encoder is implemented with a CNN that efficiently extracts and aggregates local features from the audio spectrogram \mathbf{X} . The two separate decoders are then used at each frame for detecting a variable number of onset times and judging the offset of the detected notes by focusing on different aspects of the latent features.

More specifically, at each frame t , the encoder takes as input the audio spectrogram around frame t with a receptive field of a fixed size M and outputs a hidden embedding sequence $\mathbf{H}_t \in \mathbb{R}^{F_h \times D}$ in the frequency domain with a sequence length of F_h and the hidden embedding size of D . In addition, positional encodings are incorporated into the encoder hidden states \mathbf{H}_t . Then the decoders receive \mathbf{H}_t with the cross attention (encoder-decoder attention).

For onset detection, the onset sequence \mathbf{Y}_t at frame t is initialized with the beginning-of-sequence token (BOS). The onset events are then detected using the onset decoder Decoder_{on} iteratively until the end-of-sequence token (EOS) is obtained, considering the current encoder hidden state \mathbf{H}_t , the onset sequences $\mathbf{Y}_{1:t-1}$ detected in previous times, the current onset sequence at frame t , and decoder positional encodings. The detected onset events are finally added to the active onsets set \mathbf{A} . The process is repeated throughout the input sequence \mathbf{X} .

The offset events are detected using the offset decoder Decoder_{off} , considering the current encoder hidden state \mathbf{H}_t , the active onsets set \mathbf{A} , and decoder positional encodings. Then active onsets corresponding to the detected offsets are removed from \mathbf{A} indicating the end of notes. It should be emphasized that the offset decoder does not perform sequence prediction. Instead, it predicts the offset for each onset that has been activated in the past time steps all at once.

3.2 Encoder

The encoder is based on the harmonic dilated convolution originally used for HPPNet [24] and uses the same configuration proposed for the acoustic model of HPPNet. It extracts local acoustic features with a fixed receptive field and feeds them to the decoders. There are three sets of convolutional layers with different kernel sizes: three layers with a kernel size of 7×7 , one harmonic dilated convolution layer with a kernel size of 1×3 , and five layers with a kernel size of 5×3 . The resulting receptive field in the time dimension is $M = 39$.

For streaming piano transcription, we use the shifting window approach for sequentially feeding an input spectrogram to the encoder. Instead of feeding the entire spec-

Algorithm 1 Streaming piano transcription. The length of output onset sequence equals to the number of the detected onsets, while the length of offset sequence has an additional output for pedal offset indexed as 0.

```

1: Input: Source sequence  $\mathbf{X} = (x_1, x_2, \dots, x_T)$ 
2: Output:
3: Onset sequence  $\mathbf{Y} = (Y_1^{1:k_1}, Y_2^{1:k_2}, \dots, Y_T^{1:k_T})$ 
4: Offset sequence  $\bar{\mathbf{Y}} = (\bar{Y}_1^{0:n_1}, \bar{Y}_2^{0:n_2}, \dots, \bar{Y}_T^{0:n_T})$ 
5: Parameters:
6: Receptive field of encoder:  $M$ 
7: Initialize positional encodings:  $\mathbf{PE}_{enc}$  and  $\mathbf{PE}_{dec}$ 
8: Initialize active onsets set:  $\mathbf{A} = \{\}$ 
9: for  $t = 1$  to  $T$  do
10:    $H_t \leftarrow \text{Encoder}(X_{t-\frac{M}{2}:t+\frac{M}{2}})$ 
11:    $H_t \leftarrow H_t + \mathbf{PE}_{enc}$ 
12:   // Offset decoder
13:    $n_t \leftarrow \mathbf{A}.size()$ 
14:    $\bar{Y}_t \leftarrow \text{Decoder}_{off}(H_t, \mathbf{A}, \mathbf{PE}_{dec})$ 
15:   Delete onsets in  $\mathbf{A}$  corresponding to offsets in  $\bar{Y}_t$ 
16:   // Onset decoder
17:    $k_t \leftarrow 0$ 
18:    $Y_t^{k_t} \leftarrow \text{BOS}$ 
19:    $y \leftarrow \text{BOS}$ 
20:   while  $y \neq \text{EOS}$  do
21:      $y \leftarrow \text{Decoder}_{on}(H_t, Y_{1:t-1}, Y_t^{0:k_t}, \mathbf{PE}_{dec})$ 
22:     if  $y == \text{EOS}$  then
23:       break
24:     end if
25:      $k_t \leftarrow k_t + 1$ 
26:      $Y_t^{k_t} \leftarrow y$ 
27:   end while
28:    $\mathbf{A}.add(Y_t)$ 
29: end for

```

rogram at once, we segment it into smaller chunks or windows to simulate real-time processing. These windows are shifted along the time axis, allowing the model to gradually analyze incoming audio data. We define the size of each window based on the desired temporal context for transcription. Typically, smaller window sizes facilitate faster processing but may sacrifice some contextual information, whereas larger window sizes provide more context but may introduce latency. To ensure continuity of transcription and avoid information loss at window boundaries, we apply overlap between consecutive windows.

3.3 Decoder

Both the onset and offset decoders are the same as the decoder of T5 [27] (Fig. 2). In the decoder architecture, the embedding size is set to $D_{dec} = 256$, and decoder layers to $L = 6$, attention head number to $N_{head} = 8$. The multi-layer perceptron (MLP) dimension is set to $D_{mlp} = 1024$. A maximum decoder sequence length $N_{seq} = 64$. The length of the decoder output varies with the number of activated onsets. During the training phase, we use padding and masking to fix the output tokens length of offset decoder to 16.

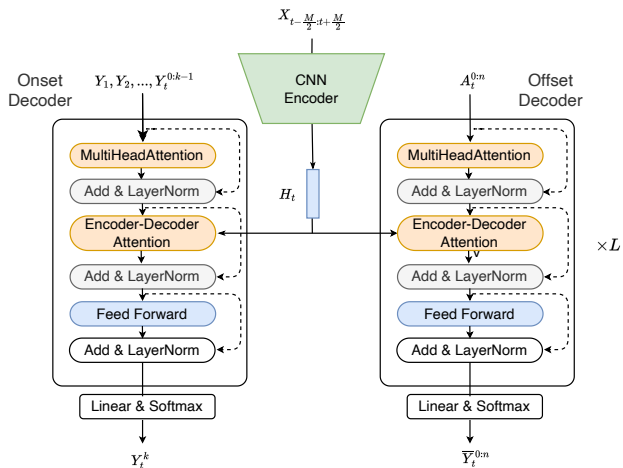


Figure 2. The implementation of the streaming transcription model that uses one encoder for latent feature extraction and two decoders for onset and offset detection.

3.4 Consistent Decoding

Existing piano transcription models that applied onset and offset detection [9, 25] often face issues with mismatched detected onsets and offsets. This is due to the little constraints in the detection processes for onsets and offsets. Although this issue can be addressed with post-processing methods, we prefer to solve it end-to-end within the model. Our proposed architecture makes a constriction to the offset decoder to detect offsets for detected onsets only, and also detects sustain pedal release events to improve performance of note offsets detection.

The onset decoder sequentially outputs onset events in an autoregressive manner while the offset decoder detects all the offset events at once for the active notes detected by the onset decoder with judgement of the sustain pedal. If the offset event for an active note is not detected at the current frame, a special token *BLANK* is obtained as described in Section 4.1.3. The onset decoder considers only notes detected in the past and current frames. The sustain pedal plays a crucial role in expressive piano performance and considerably affects offset detection. The lifting time of the sustain pedal is highly relevant to the absolute offset times and thus determines the duration and decay characteristics of musical notes.

The input of the onset decoder in each step at frame t consists of the onset tokens detected in the previous step and the onset tokens detected at previous frames. This enables to capture long-term dependency between musical notes. By incorporating information from previous frames, the decoder can better understand the context of the current onset detection and facilitate the recognition of typical patterns and structures in the music sequence over time.

4. EVALUATION

This section reports a comparative experiment conducted for evaluating the performance of the proposed and conventional piano transcription methods.

Time	Target Tokens
1	<EOS> <blank>
2	<EOS> <blank>
...	
i	<C4><D4><EOS> <blank>
$i+1$	<C4><D4><EOS> <blank><blank><blank>
$i+2$	<EOS> <blank><blank><blank>
...	
j	<E4><F4><EOS> <blank><C4_off><blank>
...	
k	<EOS> <pedal_off><D4_off><E4_off><F4_off>

Table 1. Target tokens for onset decoder(red) and offset decoder(blue).

4.1 Experimental Conditions

We explain the dataset used for evaluation and the input and output data of the proposed method.

4.1.1 Dataset

We used the MAESTRO dataset V3.0.0 [28] composed of about 200 hours of virtuosic piano performances captured with fine alignment between note events and audio recordings. The split of the dataset into training, validation, and test sets was defined officially. The validation set was used for selecting the best-performing trained model based on its performance on unseen data. The dataset also provides information about the states (on or off) of the sustain pedal. The pedal information is crucial for accurately transcribing piano performances as it affects the *actual* durations and offset times of sustained notes.

4.1.2 Input

The original audio recordings were resampled with a sampling rate of 16 kHz. To increase the variation of the training data and reduce the memory footprint, 10-sec segments were randomly clipped from the recordings and the CQT spectrograms were computed on the fly with the nnAudio library [30]. We used the CQT for its capability of capturing both higher and lower-frequency components in the logarithmic frequency domain suitable for analyzing music signals. The lowest frequency was set to 27.5 Hz corresponding to the lowest key of the standard 88-key piano. One octave was divided into 48 frequency bins and the total number of frequency bins was 352. This ensures a fine frequency resolution over the entire audible frequency range. The hop length was set to 320 samples (20 ms), taking the balance between the time resolution and the computational efficiency. After obtaining the CQT spectrogram, the amplitude values were converted to decibels (dB) using transforms available in the torchaudio library.

4.1.3 Output

The vocabulary of output tokens used in our study was the same as that used for the music transformer 3 (MT3) [8, 9] except that time location tokens were not used. This contributes to reducing the length of the output sequence and stabilizing the training. The output vocabulary consists of the following tokens:

Model	Params	Frame-level			Note-level (onset only)			Note-level (onset + duration)		
		P (%)	R (%)	F1(%)	P (%)	R (%)	F1(%)	P (%)	R (%)	F1(%)
Onsets & Frames [28]	26M	92.11	88.41	90.15	98.27	92.61	95.32	82.95	78.24	80.50
Semi-CRFs [29]	9M	93.79	88.36	90.75	98.69	93.96	96.11	90.79	86.46	88.42
HPPNet-sp [24]	1.2M	92.79	93.59	<u>93.15</u>	98.45	95.95	<u>97.18</u>	84.88	82.76	83.80
hFT-Transformer [10]	5.5M	92.82	93.66	93.24	99.64	95.44	97.44	92.52	88.69	90.53
Streaming Seq2Seq (ours)	16M	91.91	91.73	91.75	98.30	94.83	96.52	91.08	87.89	<u>89.44</u>

Table 2. The transcription performances of the existing and proposed methods on MAESTRO V3.0.0 test set.

Model	Segment	Encoder Input Seq-Length	Decoder Output Seq-Length	Latency	Note F1	Note w/ Offset F1
Seq2Seq [8]	4088 ms	511	1024	4088 ms	96.01	83.94
Streaming Seq2Seq (ours)	-	39	64	380 ms	96.52	89.44

Table 3. The transcription performances of sequence-to-sequence transcription models on MAESTRO V3.0.0 test set.

Onsets and offsets (128+128 tokens) Each token represents the presence of an onset or offset of the corresponding pitch given as a MIDI note number.

Pedal states (2 tokens) Two tokens representing the presence and absence of the sustain pedal.

BLANK (1 token) A special token representing silence or absence of any musical event.

BOS and EOS (2 tokens) Special tokens representing the beginning and end of the output sequence.

The onset decoder and offset decoder both need only part of the vocabulary. But we kept the full vocabulary for all decoders to maintain consistency in the model architecture, regardless of whether there is only one decoder or multiple decoders. We set the length of each onset and offset events into 2 frames. During the transcription process, if consecutive onsets or offsets were detected, we only kept the first one and discard the duplicates. To estimate note events from the output of the decoders we used a simple greedy regression algorithm. We then selected the nearest corresponding offsets after the onsets to determine the duration of the notes. If offsets were not detected, we selected the nearest pedal offset as the offsets for the notes or a maximum duration of 4 seconds.

4.1.4 Training

We used the cross entropy loss for training the proposed model. It represents the negative log-posterior probability over output tokens for the ground-truth annotation. For optimization, we utilized the AdamW optimizer [31], which is a variant of the Adam optimizer with weight decay regularization. The mini-batch size was set to 16 and the learning rate was set to 6e-4. A dropout rate of 0.1 was applied to the decoder layers to prevent overfitting. Training was iterated for 200,000 steps with early stopping.

4.1.5 Metrics

The performance of piano transcription was evaluated with the mir_eval library [32] in terms of the precision and re-

call rates and F1 score at the frame and note levels. In the note-level evaluation, an estimated note was judged as correct if its onset time was detected correctly or if both the onset time and duration were estimated correctly. The error tolerance in onset estimation was set to 50 msec as in many studies. The error tolerance in duration estimation was set to the larger of 50 msec or 20% of the ground-truth duration. These metrics were averaged over the test set.

4.2 Experimental Results

We report the experimental results obtained through comparative and ablation studies.

4.2.1 Comparison with Existing Methods

We conducted a comprehensive experiment that compared our method with state-of-the-art methods such as frame-level and event-level transcription methods (Table 2). We found that our method achieved competitive performance and surpassed an event-level transcription method named Semi-CRFs in terms of both the note-level F1-scores with and without duration evaluation. This superiority indicates the robustness of our method in capturing the musical onset events and their corresponding offsets.

4.2.2 Sequence to Sequence Transcription

For comparison, we tested the generic transformer-based sequence-to-sequence transcription model [8] (Table 3). Audio recordings were split into segments of 4088 msec to be fed to the encoder. Since different segments were transcribed independently, the long-term correlation between note events is hard to learn from the data. Moreover, increasing the segment length would exponentially increase the computational complexity of the self-attention layers. It would increase the number of absolute-time-location tokens and further complicates the estimation of time locations for note events.

Thanks to the streaming encoder-decoder architecture, the proposed model kept the actual input length constant

Decoder	Onset	Offset	Pedal	Note-level (onset only)			Note-level (onset+duration)		
				P (%)	R (%)	F1(%)	P (%)	R (%)	F1(%)
1	✓	✓	✓	98.32	93.36	95.73	89.91	85.41	87.56
2	✓	✓		98.23	94.75	96.44	88.11	85.00	86.51
2	✓	✓	✓	98.30	94.83	96.52	91.08	87.89	89.44

Table 4. Ablation study on MAESTRO V3.0.0 test set.

and significantly reduced the computational complexity of the self-attention layers. The length of the encoder input was set to 39 and the maximum length of the decoder output was set to 64. This enables the processing of variable-length audio recordings without the need for segmentation and offers the potential for real-time transcription. Compared with the generic model, our streaming model showed better performance in terms of the note-level F1-scores with and without duration evaluation. This indicates the potential application to streaming and sequence-to-sequence music transcription scenarios.

4.2.3 Latency

The latency of a streaming model refers to the gap between the actual time of an onset or offset event and the time of the event output. Putting the actual computational speed aside, the latency of a non-streaming model is equal to the length of the input sequence because the whole sequence needs to be processed for generating outputs. In contrast, for streaming models, the latency is equal to the length of future frames in the input data stream.

In Table 3, our streaming model had a latency of 380 msec. The CNN-based encoder takes 19 future frames and 19 past frames as input. Even with a short input context, the streaming model still achieved competitive performance on piano transcription. This indicates that onset and offset events could be detected without heavily relying on long-term dependency of acoustic features.

4.2.4 Ablation Study

To verify the effectiveness of sustain pedal detection and that of the separated decoders for onset and offset detection, we conducted an ablation study. Besides the proposed model, we trained a model without pedal detection and another model that uses a single decoder for onset, offset, and pedal detection. The training and evaluation were performed in the same way.

Table 4 shows the performances of the compared methods. We found that removing the pedal detection slightly decreased the note-level F1-score without duration estimation, but significantly degraded the note-level F1-score with duration estimation. This suggests that pedal detection plays a crucial role in estimating note durations. Similarly, using a single decoder for both onset and offset detection degraded both the note-level F1-scores with and without duration estimation, compared with the proposed model. This demonstrated the effectiveness of incorporating pedal detection and a separated decoder for onset and offset prediction for better piano transcription.

5. CONCLUSION

In this paper, we have presented a novel streaming audio-to-MIDI piano transcription method. We tackled an open problem of detecting note onset and offset events from a piano recording in an online manner. Our method is based on a streaming encoder-decoder architecture that combines a convolutional encoder for aggregating local acoustic features with separate transformer decoders for detecting onset and offset events at each time step while validating the use of the sustain pedal.

In extensive experiments with the MAESTRO dataset, our method attained competitive performance, compared with the state-of-the-art offline methods. Our model also outperformed the generic transformer-based sequence-to-sequence model in terms of both accuracy and latency. The ablation study showed the effectiveness of incorporating pedal detection and that of using the separated decoders for onset and offset detection. Our method uses a limited number of incoming frames for detecting the onset and offset events and paved a way for latency-critical practical applications. We achieved a system latency of 380 msec and plan to thoroughly investigate the trade-off between the latency and the transcription performance. Additionally, decoding every frame may not be necessary. Some scenarios might not require such high temporal precision. The setting of the time step also requires further exploration for real-time scenarios.

6. ACKNOWLEDGMENTS

This work was partially supported by JST SPRING No. JPMJSP21110, JST FOREST No. JPMJFR2270, and JSPS KAKENHI Nos. 24H00742 and 24H00748.

7. REFERENCES

- [1] E. Benetos, S. Dixon, Z. Duan, and S. Ewert, “Automatic music transcription: An overview,” *IEEE Signal Processing Magazine*, vol. 36, no. 1, pp. 20–30, 2018.
- [2] A. Graves, A. Mohamed, and G. E. Hinton, “Speech recognition with deep recurrent neural networks,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 6645–6649.
- [3] C. Chiu and C. Raffel, “Monotonic chunkwise attention,” in *Proceedings of International Conference on Learning Representations (ICLR)*, 2018.

- [4] X. Chen, Y. Wu, Z. Wang, S. Liu, and J. Li, “Developing real-time streaming transformer transducer for speech recognition on large-scale dataset,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 5904–5908.
- [5] T. Kwon, D. Jeong, and J. Nam, “Polyphonic piano transcription using autoregressive multi-state note model,” in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, J. Cumming, J. H. Lee, B. McFee, M. Schedl, J. Devaney, C. McKay, E. Zangerle, and T. de Reuse, Eds., 2020, pp. 454–461.
- [6] D. Jeong and S. Telecom, “Real-time automatic piano music transcription system,” *Late Breaking/Demo of the International Society for Music Information Retrieval Conference (ISMIR)*, pp. 4–6, 2020.
- [7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Annual Conference on Neural Information Processing Systems (NIPS)*, 2017, pp. 5998–6008.
- [8] C. Hawthorne, I. Simon, R. Swavely, E. Manilow, and J. H. Engel, “Sequence-to-sequence piano transcription with transformers,” in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2021, pp. 246–253.
- [9] J. Gardner, I. Simon, E. Manilow, C. Hawthorne, and J. H. Engel, “MT3: multi-task multitrack music transcription,” in *Proceedings of International Conference on Learning Representations (ICLR)*, 2022.
- [10] K. Toyama, T. Akama, Y. Ikemiya, Y. Takida, W. Liao, and Y. Mitsufuji, “Automatic piano transcription with hierarchical frequency-time transformer,” in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2023, pp. 215–222.
- [11] W. T. Lu, J. Wang, and Y. Hung, “Multitrack music transcription with a time-frequency perceiver,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.
- [12] A. Khlif and V. Sethu, “An iterative multi range non-negative matrix factorization algorithm for polyphonic music transcription,” in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2015, pp. 330–335.
- [13] S. A. Abdallah and M. D. Plumbley, “Unsupervised analysis of polyphonic music by sparse coding,” *IEEE Transactions on Neural Networks*, vol. 17, no. 1, pp. 179–196, 2006.
- [14] E. Vincent, N. Bertin, and R. Badeau, “Adaptive harmonic spectral decomposition for multiple pitch estimation,” *IEEE Transactions on Speech Audio Processing*, vol. 18, no. 3, pp. 528–537, 2010.
- [15] J. Nam, J. Ngiam, H. Lee, and M. Slaney, “A classification-based polyphonic piano transcription approach using learned feature representations,” in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2011, pp. 175–180.
- [16] R. Kelz, M. Dorfer, F. Korzeniowski, S. Böck, A. Arzt, and G. Widmer, “On the potential of simple frame-wise approaches to piano transcription,” in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2016, pp. 475–481.
- [17] R. Kelz, S. Böck, and G. Widmer, “Deep polyphonic ADSR piano note transcription,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 246–250.
- [18] S. Sigtia, E. Benetos, and S. Dixon, “An end-to-end neural network for polyphonic piano music transcription,” *IEEE ACM Transactions on Audio Speech and Language Processing (TASLP)*, vol. 24, no. 5, pp. 927–939, 2016.
- [19] C. Hawthorne, E. Elsen, J. Song, A. Roberts, I. Simon, C. Raffel, J. H. Engel, S. Oore, and D. Eck, “Onsets and frames: Dual-objective piano transcription,” in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2018, pp. 50–57.
- [20] S. Böck and M. Schedl, “Polyphonic piano note transcription with recurrent neural networks,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 121–124.
- [21] N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent, “Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription,” in *Proceedings of the International Conference on Machine Learning (ICML)*, 2012.
- [22] L. Ou, Z. Guo, E. Benetos, J. Han, and Y. Wang, “Exploring transformer’s potential on automatic piano transcription,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 776–780.
- [23] R. Wu, X. Wang, Y. Li, W. Xu, and W. Cheng, “Piano transcription with harmonic attention,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 1256–1260.
- [24] W. Wei, P. Li, Y. Yu, and W. Li, “HPPNet: Modeling the harmonic structure and pitch invariance in piano transcription,” in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2022, pp. 709–716.
- [25] Q. Kong, B. Li, X. Song, Y. Wan, and Y. Wang, “High-resolution piano transcription with pedals by regressing onset and offset times,” *IEEE ACM Transactions*

on *Audio Speech and Language Processing (TASLP)*, vol. 29, pp. 3707–3717, 2021.

- [26] B. S. M. Awiszus, “Automatic music transcription using sequence to sequence learning,” Ph.D. dissertation, Master’s thesis, Karlsruhe Institute of Technology, 2019.
- [27] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, “Exploring the limits of transfer learning with a unified text-to-text transformer,” *Journal of machine learning research*, vol. 21, pp. 140:1–140:67, 2020.
- [28] C. Hawthorne, A. Stasyuk, A. Roberts, I. Simon, C.-Z. A. Huang, S. Dieleman, E. Elsen, J. Engel, and D. Eck, “Enabling factorized piano music modeling and generation with the MAESTRO dataset,” in *Proceedings of International Conference on Learning Representations (ICLR)*, 2019, pp. 1–6.
- [29] Y. Yan, F. Cwitkowitz, and Z. Duan, “Skipping the frame-level: Event-based piano transcription with neural Semi-CRFs,” in *Advances in Neural Information Processing Systems (NIPS)*, vol. 34, 2021, pp. 20 583–20 595.
- [30] K. W. Cheuk, H. Anderson, K. Agres, and D. Herremans, “nnAudio: An on-the-fly GPU audio to spectrogram conversion toolbox using 1d convolutional neural networks,” *IEEE Access*, vol. 8, pp. 161 981–162 003, 2020.
- [31] I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” in *Proceedings of International Conference on Learning Representations (ICLR)*, 2019.
- [32] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, D. P. Ellis, and C. C. Raffel, “Mir_eval: A transparent implementation of common mir metrics,” in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2014, pp. 1–6.

TOWARDS UNIVERSAL OPTICAL MUSIC RECOGNITION: A CASE STUDY ON NOTATION TYPES

Juan C. Martinez-Sevilla¹

David Rizo^{1,2}

Jorge Calvo-Zaragoza¹

¹ Pattern Recognition and Artificial Intelligence Group, University of Alicante, Spain

² Instituto Superior de Enseñanzas Artísticas de la Comunidad Valenciana, Spain

{jcmartinez.sevilla, drizo, jorge.calvo}@ua.es

ABSTRACT

Recent advances in Deep Learning have propelled the development of fields such as Optical Music Recognition (OMR), which is responsible for extracting the content from music score images. Despite progress in the field, existing literature scarcely addresses core issues like performance in real-world scenarios, user experience, maintainability of multiple pipelines, reusability of architectures and data, among others. These factors result in high costs for both users and developers of such systems. Furthermore, research has often been conducted under certain constraints, such as using a single musical texture or type of notation, which may not align with the end-user requirements of OMR systems. For the first time, our study involves a comprehensive and extensive experimental setup to explore new ideas towards the development of a *universal* OMR system—capable of transcribing all textures and notation types. Our investigation provides valuable insights into several aspects, such as the ability of a model to leverage knowledge from different domains despite significant differences in music notation types.

1. INTRODUCTION

Optical Music Recognition (OMR) is a field of research focused on converting written music documents into machine-readable formats, such as Humdrum `**kern`, MEI, or MusicXML [1–3]. This technology holds significant promise for digital musicology, libraries, and academia, facilitating the digitization of scores for further musical analysis, large-scale information retrieval, and making vast musical archives more accessible [4].

Historically, the development of OMR has evolved from relying on basic heuristic methods to a more dynamic application of Deep Learning (DL) techniques [5]. This shift brought new advances to the field, leading to substantial improvements in the accuracy of music score transcrip-

tion [6–9]. However, despite these advances, OMR models still face significant challenges in generalization. The DL methodologies, while robust in specific contexts, often struggle to perform consistently across diverse data distributions [10]. This is particularly evident when dealing with a variety of music notations and textures, from ancient Neumatic chants to modern polyphonic compositions. Most existing OMR works focus on a narrow range of music types (often just one), which limits their usability for more comprehensive archival tasks [11].

In response to these limitations, this paper proposes the conceptualization of a *universal* OMR system capable of processing *all* types of musical notations and textures.¹ The long-term objective is to develop a versatile technology that can adapt to any musical document, regardless of its historical period or stylistic characteristics.

This paper takes the first steps towards such a system by exploring a few alternatives to achieve this goal. In particular, we carry out a specific case study focused on diverse notation types, involving medieval square notation, Mensural notation, and Common Western Modern Notation (CWMN) corpora. We consider whether it is more feasible to develop separate OMR models for each notation or to create a single, all-encompassing model. This dichotomy has not been thoroughly studied before. Separate OMR models for each notation maximize accuracy by addressing specific characteristics, but require extensive resources and individual updates. Conversely, a single, all-encompassing model enhances scalability and maintenance efficiency, benefiting from shared knowledge across notations—a potential advantage in deep learning—although it may struggle with variability. Additionally, we include an intermediate case in which a part of the model is common and only one specialized module is created for each notation, thereby representing a trade-off between the previous pros and cons.

This paper is organized as follows: Section 2 offers background information on OMR. In Section 3, we outline our methodology for analyzing the question at hand and the different training scenarios to leverage the system’s performance. Section 4 details the experimental setup, while

¹ In this work, we will focus on Western notations that share some fundamental characteristics, such as indicating duration with the shape of the music-notation symbols and pitch with their position over a set of staff lines. These also follow a left-to-right reading order.



Section 5 presents the work results and analysis. Finally, we conclude the paper in Section 6, along with potential avenues for future research.

2. RELATED WORK

Modern research in OMR using DL methodologies has led to several successful approaches [4,5,12]. Notably, one approach that stands out is the so-called “end-to-end” formulation. This approach provides a holistic method where images of music notation are directly inputted into the model, which then predicts their content. The end-to-end formulation represents the state of the art in related areas such as text or speech recognition and is now considered by several works in OMR [11, 13–15].

Some works have successfully addressed end-to-end OMR for monophonic staff images, likely because most ancient notations depict monophonic staves. Specific efforts are underway to address other textures such as homophonic scores [6], polyphonic music [7, 16], and vocal pieces [8, 17]. However, despite recent advances in the field, there is still no approach for building a *universal* OMR system capable of handling all this variability of music notation types and textures simultaneously.

The fundamental challenge lies in an unsolved problem in DL models: they perform well when there are regular statistics and abundant data to train on, allowing them to learn the regularities in the distribution properly [10]. This is not the case in the OMR problem, where rich labeled data is scarce and the graphical feature variability is extensive, making it a complex task.

Due to the inherent characteristics of DL methodologies, the existing literature work with analogous or highly similar train-test distributions [11]. Consequently, since there is a lack of research focusing on the development of *universal* OMR systems capable of processing any input score regardless of its content, we propose the first study aimed at developing, understanding, and evaluating a *universal* OMR system for dealing with different notation types simultaneously.

3. METHODOLOGY

Our objective is to explore an initial approach towards developing a *universal* OMR system. Specifically, we consider the case of accounting for different notation types. To achieve this, we consider three different scenarios: (i) a single model per dataset; (ii) a model leveraging all available data; and (iii) a hybrid model, for which some parts are common across all cases, but there are also specific layers tailored to each notation type. We opt for a deep end-to-end model as representative of the state-of-the-art in OMR. Below, we provide a detailed explanation of how this model works and then explain the different approaches selected to address the task.

3.1 Learning framework

The end-to-end OMR model seeks to directly retrieve the music notation from a single staff image. As in recent lit-

erature [11, 13, 14], we assume that a certain preprocessing stage has already separated the staves of the score [18].

Based on other works addressing the OMR challenge [13], a Convolutional Recurrent Neural Network (CRNN) scheme is considered for the end-to-end pipeline. The CRNN architecture incorporates an *encoder*: a block of convolutional layers that learns a set of features from the input image. Then, it includes a *decoder*: group of recurrent stages that model the temporal dependencies of the feature-learning block. Finally, a fully connected network with a softmax activation is used to retrieve a posterio-gram, which is decoded to obtain the predicted *musical symbols*². The Connectionist Temporal Classification (CTC) training procedure [19] is used to achieve an end-to-end scheme, as it allows training the network using unsegmented sequential data.

For training, let $\mathcal{T} \subset \mathcal{X} \times \Sigma^*$ be a set of data where sample $x_i \in \mathcal{X}$ of single staff image is related to symbol sequence $\mathbf{z}_i = (z_{i1}, z_{i2}, \dots, z_{i|\mathbf{z}_i|}) \in \Sigma^*$, where Σ represents the symbol vocabulary used for encoding the music score. Note that the use of CTC to model the transcription task requires the inclusion of an additional “*blank*” symbol in the Σ vocabulary, i.e., $\Sigma' = \Sigma \cup \{\textit{blank}\}$.

At prediction, for a given music staff image input $x_i \in \mathcal{X}$, the model outputs a posterio-gram $p_i \in \mathbb{R}^{|\Sigma'| \times K}$, where K represents the number of frames provided by the recurrent stage. Finally, the predicted sequence $\hat{\mathbf{z}}_i$ is obtained resorting to a *greedy* policy that retrieves the most probable symbol per frame in p_i , later a subsequent mapping function merges consecutive repeated symbols and removes *blank* labels.

3.2 Approaches to OMR for different notation types

In order to explore diverse learning frameworks to assess the transcription performance, we pose three different scenarios that differ in how data is fed to the model and the training strategies for the model layers. An overview of these scenarios is described as follows (illustrated in Figure 1):

Only: In this scenario, one model is trained for each single dataset. This is the baseline of our experiments and it will allow us to compare properly the different approaches selected. It should be emphasized that this training scenario will employ a set of resources and time associated with each corpora. This methodology stands as the current state of the art, as recent research resorted to training individual models as described in Sec.2.

All: For this scenario, all available notation types in this work are merged to train a single model. As commented in the introduction, our long-term objective is to create a *universal* OMR capable of retrieving all types of notation and textures. This option allows us to explore the capabilities and drawbacks of integrating all possibilities in the same model.

² In this work, a *musical symbol* is represented as the conjunction of the glyph or shape and the position within the staff.

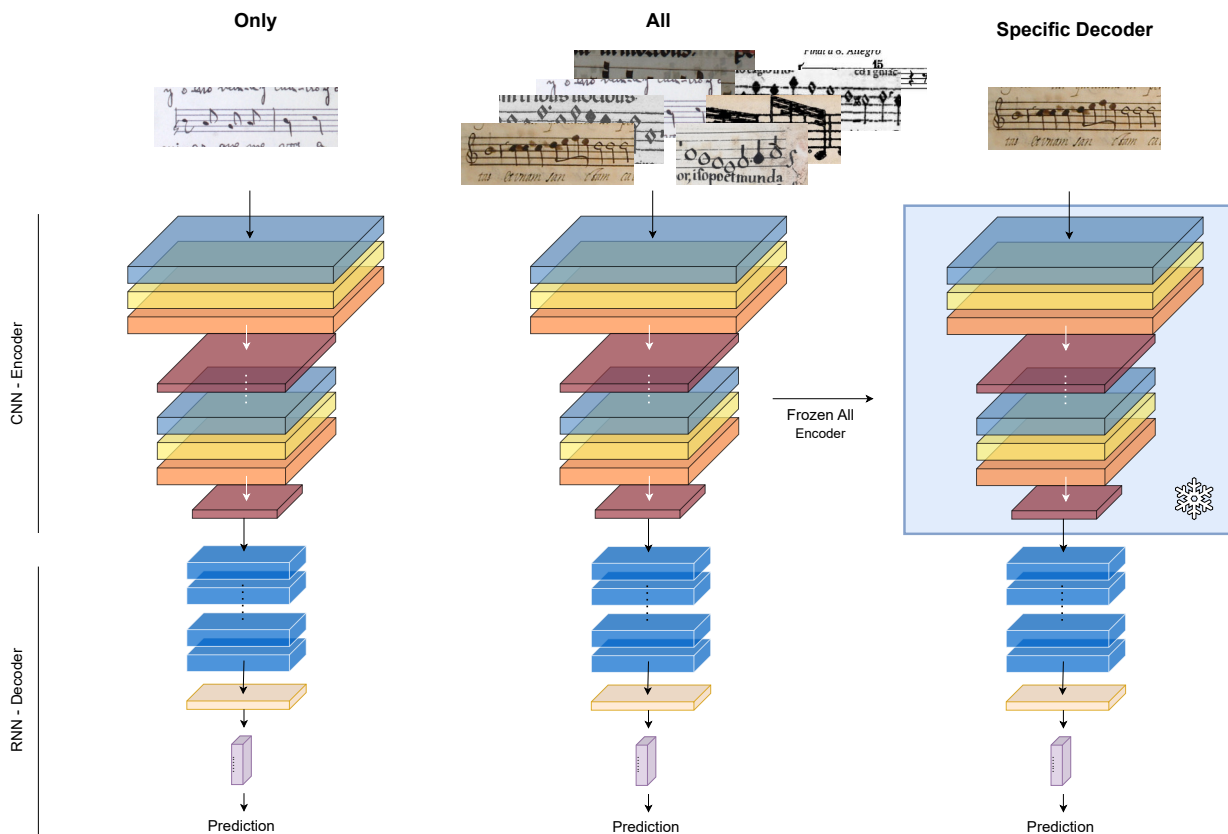


Figure 1: Graphical scheme of the three different approaches considered for this work using the CRNN architecture as the backbone. *Only*: a model trained per notation type individually. *All*: a model trained with all the notation types available for this work. *Specific Decoder*: once the *All* scenario is finished, the encoder (already trained) is frozen to train specific decoders for each notation type.

Specific Decoder: Recent DL approaches pictured the adequacy of learning via a general feature extractor (encoder) [20]. Similarly, we leverage the encoder block of the *All* approach weights as our starting point. By doing so, we establish an already-evaluated feature extractor shared across all corpora. Having the features extracted, we then fine-tune a notation-specific decoder block based on the unique underlying musical context.

The selection of these scenarios helps to study performance but also other important aspects such as maintainability, reusability, or resource leveraging, which are valuable for real-case systems and have been barely analyzed in OMR literature.

4. EXPERIMENTAL SETUP

According to the choices made for the experimental road map, we first introduce the studied evaluation metrics. Later we give further details about the learning model hyperparameters selected and the training techniques used. Eventually, we describe the data collections used for train and evaluation.

4.1 Evaluation

Current OMR systems are designed to serve as a tool. Bearing this in mind, it should be more than interesting

to compute the amount of effort it would take a user to correct the errors made by the system. However, there is not a clear way of properly measuring this case. This is why when evaluating an OMR system we resort to the *Symbol Error Rate* (SER). Given a prediction \hat{z}_i and the ground truth musical symbol sequence z_i , SER is calculated as the average number of elementary editing operations (insertions, deletions, or substitutions) required to convert prediction \hat{z}_i into reference z_i , normalized by the length of the latter. Formally, this is expressed as:

$$SER (\%) = \frac{\sum_{i=1}^{|\mathcal{S}|} ED(\hat{z}_i, z_i)}{\sum_{i=1}^{|\mathcal{S}|} |z_i|} \tag{1}$$

where \mathcal{S} is a set of test data, $ED : \mathcal{Z} \times \mathcal{Z} \rightarrow \mathbb{N}_0$ denotes the string edit distance, and \hat{z}_i and z_i respectively represent the estimated and target sequences.

4.2 Neural model configuration

The CRNN hyperparameters used in this study are based on the ones used in previous works [11, 13]. Authors adopt a 4 Convolutional layer block with batch normalization 2D, Leaky ReLu activation, and max-pooling 2D down-sampling. Feature maps extracted from the encoder, i.e. the Convolutional Neural Network (CNN) block, are introduced into 2 Bidirectional Long Short-Time Memory

(BLSTM) layers with 256 hidden units each and a dropout value of $d = 50\%$ followed by a fully connected network with $|\Sigma|$ units. The architecture described results in a model with 5.3M parameters.

All the models were trained with a batch size of 16 samples—it is important to mention that given the different sizes of the datasets, all the generated batches had the same proportion of samples from each dataset so the network did not adjust to the bigger dataset, i.e., dataset interleaving. The ADAM [21] optimizer was considered, a fixed learning rate of 10^{-3} , and weight decay of 10^{-6} . We iterate over 200 epochs using image augmentation techniques (blur, rotation, contrast, erosion, brightness, etc.), ensuring the robustness of the model, keeping the weights of the model that minimize the SER evaluation metric in the validation partition. The early stopping technique is used with a patience of 20 epochs. Lastly, all experiments were run using the Python language (v3.10.13) with the PyTorch and PyTorch Lightning frameworks on a single NVIDIA GeForce RTX 4090 card with 24GB of GPU memory.

4.3 Datasets

As introduced in Sec.1, music manuscripts depict a great challenge for transcription methods. Their variety in content and appearance poses a still unsolved question. In order to study the adequacy of a *universal* OMR system, we gathered data sources taking into account their variability in terms of notation, graphical appearance, and musical context, aiming to reflect Western musical diversity. A set of 40 different works has been collected that have been grouped to simplify the experimentation and insights reported. Among them, we find square notation, white Mensural notation, and CWMN. A brief description of some dataset features and staves can be found in Table 1 and Fig. 2.

Table 1: Dataset descriptions in terms of notation type, pages, music fragments (staves), and vocabulary sizes.

Notation type	Dataset	Number of pages	Music fragments	Vocabulary size
Square	AUSTRIA	685	4 850	270
	BNE	4 125	27 746	709
	SEILS	151	1 136	206
	GUATEMALA	385	3 263	316
	CAPITAN	97	828	373
Mensural	FMT	348	1 305	425
	CATEDRALES	52	308	245
CWMN	CATEDRALES	52	308	245
	CAMERA-PRIMUS	–	15 000	1 443

Diverse cases have been considered looking for different printers, copyists, authors, and periods considering the more variability the better. The list of datasets used is classified by notation type and ordered temporally below.

4.3.1 Square Notation

Square notation is written on a staff with four lines and three spaces. In this notation, ascending notes are shown as stacked squares, while descending notes are written with

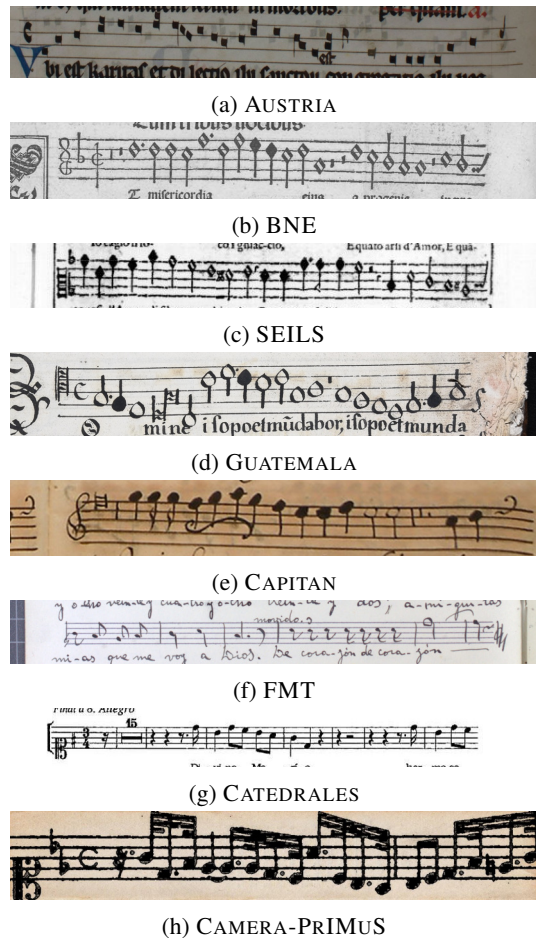


Figure 2: Samples of staves of the different datasets employed in the experimentation.

diamonds. This system of notation appears in liturgical chant books.

AUSTRIA. The Austria dataset contains 685 printed pages of 15th-century manuscripts in German Gothic square notation. Provided by the Austrian Centre for Digital Humanities and Cultural Heritage.³

4.3.2 White Mensural Notation

Notation system used in polyphonic European vocal music. Mensural notation can use different note shapes to denote rhythmic durations. It is written on a staff with five lines and four spaces.

BNE. The “Biblioteca Nacional de España (BNE)” dataset corresponds to the pages from the corpus obtained from the collection of mensural books of the Biblioteca Digital Hispánica.⁴ It comprises multiple authors and printers, e.g., F. Guerrero, H. of G. Scoto or Antonio Gardano, with a size of 4 125 pages. License: public.

SEILS. The “Second Edition of the Il Lauro Secco (SEILS)” dataset consists of 151 printed pages of the “Il

³ <https://www.oeaw.ac.at/> (accessed April 8th, 2024)

⁴ <https://www.bne.es/es/catalogos/biblioteca-digital-hispanica> (accessed April 8th, 2024)

Lauro Secco” collection corresponding to an anthology of 16th-century Italian madrigals in white Mensural notation [22]. License: public.

GUATEMALA. The Guatemala dataset incorporates 383 handwritten pages from a polyphonic choir book, part of a larger collection held at the “Archivo Histórico Arquidiocesano de Guatemala” [23]. License: private.

CAPITAN. The Capitan dataset contains 100 handwritten pages of 17th-century manuscripts in late white Mensural notation extracted from collections found in the “Catedral del Pilar” in Zaragoza [24]. License: private.

4.3.3 Common Western Modern Notation

Current notation system, written in five lines and four spaces. It is capable of indicating to the musician all the parameters to properly interpret the piece, such as dynamics or tempo changes.⁵

FMT. This collection consists of four groups of handwritten score sheets of popular Spanish songs transcribed by musicologists between 1944 and 1960. taken from the “Fondo de Música Tradicional IMF-CSIC”⁶, with a total of 348 images. License: public.

CATEDRALES. The Catedrales dataset contains 52 pages of printed liturgical examples from Málaga, Granada, and Sevilla cathedral archives [25]. License: public.

CAMERA-PRIMUS. The Printed Images of Music Staves (PRIMuS) dataset is a hybrid corpus, i.e., the musical content comprehends the RISM Database⁷ but the images have been obtained using the digital engraver tool Verovio [26]. To the generated images multiple distortions and textures are applied to simulate the look and conditions of the real sources. Although the original dataset consists of almost 100 000 samples, we have randomly selected 15 000 to make it more suitable for our experimentation [27]. License: public.

All the datasets presented use an agnostic output encoding which represents a musical symbol as `glyph:position_in_staff`. This encoding helps transcribe the tokens given their graphical appearance rather than their musical meaning, which can be ambiguous in many situations for the model to learn, making it unsuitable for OMR. Additionally, the agnostic encoding facilitates a straightforward conversion to standard formats such as MusicXML, MEI, or Humdrum `**kern` [28].

5. RESULTS

Table 2 presents the test results obtained with the proposed experimental scheme in terms of the SER (%) metric.

The *Only* scenario acts as our baseline. Here training, validation, and testing splits comprise exclusively samples

⁵ For evaluation, pitch, rhythm and articulation are considered.

⁶ <https://musicatradicional.eu/es/home> (accessed April 8th, 2024)

⁷ <https://rism.info/> (accessed April 8th, 2024).

Table 2: Results in terms of the SER(%) metric for the training scenarios *Only*, *All* and *Specific Decoder*.

Dataset	Only (baseline)	All	Specific Decoder
AUSTRIA	3.77	3.87	3.78
BNE	3.25	3.67	3.31
SEILS	2.71	1.88	1.94
GUATEMALA	2.22	1.87	1.88
CAPITAN	8.60	6.80	7.91
FMT	8.98	5.72	7.11
CATEDRALES	17.34	8.49	17.94
CAMERA-PRIMUS	1.54	3.07	1.60

from each individual dataset. We observe varying performances across different datasets. Notably, the SER metric ranges from 1.54% for the CAMERA-PRIMUS dataset to 17.34% for the CATEDRALES dataset. This indicates significant variability in model performance depending on the dataset size, notation, and graphical features, being the higher values the ones associated with CWMN, where we find more complex musical symbols and context.

When training on the *All* scenario, the model demonstrates performance improvements compared to the *Only* scenario for most datasets. This proves the validity of unifying training pipelines for different notations as the model learns to extract more robust features from the images, which helps in datasets with fewer samples while sacrificing very little accuracy in other datasets, e.g., BNE or CAMERA-PRIMUS. It is worth highlighting the great improvement in the CATEDRALES dataset reducing the SER from 17.34% to 8.49%. On the other hand, we lose accuracy in datasets such as AUSTRIA (from 3.77% to 3.87%), BNE (from 3.25% to 3.67%), and CAMERA-PRIMUS (from 1.54% to 3.07%). This situation reports valuable insights given that on bigger datasets like BNE or CAMERA-PRIMUS with enough data to be trained individually we lose performance, but if we are willing to sacrifice that performance we improve in several datasets. AUSTRIA poses a different situation, due to being the only square notation dataset, the labeling is slightly different to the other corpora increasing the SER metric when merging it with the other datasets.⁸

After training on the *All* scenario, experiment outcomes show the adequacy of merging different datasets to better learn the data features. Thus, in the *Specific Decoder* scenario, the *All* encoder, or CNN, is frozen, and specific decoders were trained for each dataset individually. This approach is aimed at capturing dataset-specific features and learning the underlying musical language of each dataset.

While some datasets exhibited improved performance

⁸ For the *All* scenario we have checked that the tokens predicted are present in the target vocabulary. Without performance variation, such a fact evinces the adequacy of using all data available to train a unique model, to better learn the image features and the inherent difficulty of music when applying OMR without focusing on a specific given dataset.

(e.g., SEILS with a 1.94% SER in the *Specific Decoder* compared to 2.71% SER in the *Only* setup), others experienced only marginal improvements or even a slight degradation in performance (BNE, GUATEMALA, CAPITAN, FMT, CAMERA-PRIMUS). Since this approach could be discarded at first glance for not being the best performing, we make an in-depth explanation of the results obtained in the latter scenario in Sec. 5.1.

5.1 Time-efficient model training

Another important factor to take into account when looking at the experiment results is the time consumption, which is a key factor to better understand the outcomes of this research. Given the datasets presented in this work, we employ a total of 54 436 monophonic staff images with different notation types and graphical features. In Fig. 3, we report the runtime of the experiments presented. When using the training scenario specified as *All* and the configuration explained, the time that took to train the model was 1D 20H 36M 49S. If we evaluate the performance obtained in the *All* scenario we could think that these are the best approaches, as the SER metric poses improvements even in datasets with few samples. However, in real scenarios, this approach would have to be retrained from scratch in case we want to integrate a new dataset⁹. That is why the *Specific Decoder* scenario—where a common CNN is trained and specific decoders, i.e., BLSTMs, are created for each dataset—emerges, given that once the encoder block (CNN) is trained the average time to integrate a new dataset, i.e., train its decoder block, is 1H 6M 11s. This time-efficient model training approach attends more accurately to the end-user requirements in conjunction with better resource management.

This analysis strengthens our proposal of building a *universal* OMR system, that leverages all the existent musical data and is capable of transcribing multiple notation types. In these experiments, we explore the end-to-end architecture for every notation type, which clearly helps as explained in Sec.5. This will allow creating a robust, maintainable, reusable system as a first step never done before towards *universal* OMR.

6. CONCLUSIONS

This work stands out as the first to introduce the *universal* OMR goal, which involves the design, construction, and evaluation of a system capable of retrieving musical content from a document, taking into account different notation types and textures, such as monophonic, homophonic, vocal, polyphonic, etc., and the end-user requirements in real-case scenarios. To achieve this, we studied and compared different settings of real and heterogeneous data corpora to provide invaluable insights into these first steps towards *universal* OMR.

The obtained results validate the capabilities of current OMR state-of-the-art model architectures to transcribe real

⁹ Except if we use Continual Learning techniques [29], yet to be explored in OMR.

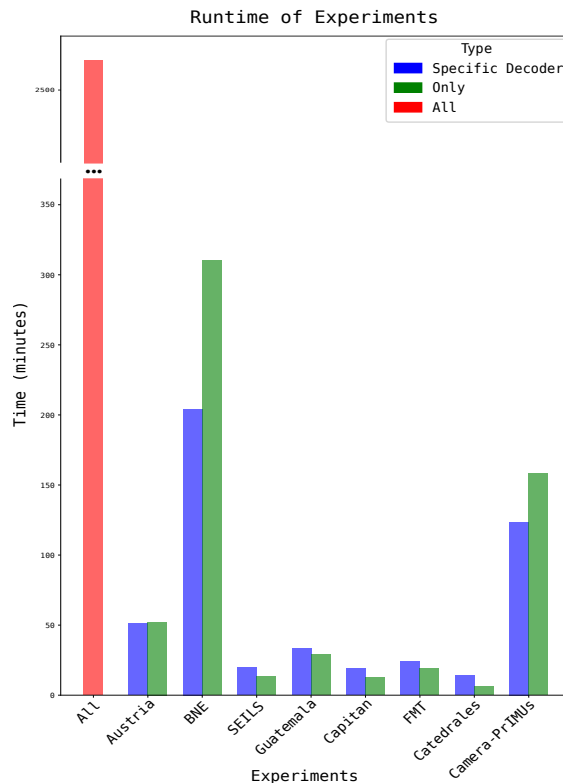


Figure 3: Runtime of experiments presented in this work in minutes for the *All*, *Only* (baseline) and *Specific Decoder* scenarios.

documents with different notation types, as the SER(%) rates match those observed in works that exclusively address one notation (either square, Mensural, or CWMN). Moreover, the use of a frozen trained encoder block as a common feature extractor proves to be useful for saving resources, maintaining the system, and reducing training time, since in some cases it considerably improves the overall transcription performance when there are not enough samples.

Future work seeks to expand the presented assortment by considering other textures such as homophony, vocal, or polyphony, to provide further insights and analysis towards *universal* transcription pipelines. Fine-tuning all or certain layers of the encoder would also be relevant, given that differences among datasets manifest in their visual representation rather than in their output. Furthermore, given the results obtained, another promising avenue is to investigate adequate encoding formats to properly represent music from different centuries and textures.

7. ACKNOWLEDGEMENTS

This paper is supported by grant CISEJI/2023/9 from “Programa para el apoyo a personas investigadoras con talento (Plan GenT) de la Generalitat Valenciana”.

8. REFERENCES

- [1] D. Huron, “Humdrum and Kern: Selective Feature Encoding BT - Beyond MIDI: The handbook of musi-

- cal codes,” in *Beyond MIDI: The handbook of musical codes*. Cambridge, MA, USA: MIT Press, Jan 1997, pp. 375–401.
- [2] A. Hankinson, P. Roland, and I. Fujinaga, “The music encoding initiative as a document-encoding framework,” in *Proceedings of the 12th International Society for Music Information Retrieval Conference, ISMIR 2011, Miami, Florida, USA, October 24-28, 2011*. University of Miami, 2011, pp. 293–298.
- [3] M. Good *et al.*, “Musicxml: An internet-friendly format for sheet music,” in *Xml conference and expo*. Citeseer, 2001, pp. 03–04.
- [4] M. Alfaro-Contreras, D. Rizo, J. M. Iñesta, and J. Calvo-Zaragoza, “OMR-assisted transcription: a case study with early prints,” in *Proceedings of the 22nd International Society for Music Information Retrieval Conference*. Online: ISMIR, Nov. 2021, pp. 35–41.
- [5] J. Calvo-Zaragoza, J. Hajic Jr, and A. Pacha, “Understanding optical music recognition,” *ACM Computing Surveys (CSUR)*, vol. 53, no. 4, pp. 1–35, 2020.
- [6] M. Alfaro-Contreras, J. M. Iñesta, and J. Calvo-Zaragoza, “Optical music recognition for homophonic scores with neural networks and synthetic music generation,” *Int. J. Multim. Inf. Retr.*, vol. 12, no. 1, p. 12, 2023.
- [7] J. Mayer, M. Straka, J. H. Jr., and P. Pecina, “Practical end-to-end optical music recognition for pianoform music,” *CoRR*, vol. abs/2403.13763, 2024.
- [8] M. Villarreal and J. A. Sánchez, “Synchronous recognition of music images using coupled n-gram models,” in *Proceedings of the ACM Symposium on Document Engineering 2023*, 2023, pp. 1–9.
- [9] A. Ríos-Vila, J. Calvo-Zaragoza, D. Rizo, and T. Paquet, “Sheet music transformer ++: End-to-end full-page optical music recognition for pianoform sheet music,” *CoRR*, vol. abs/2405.12105, 2024.
- [10] Y. Bengio, Y. Lecun, and G. Hinton, “Deep learning for AI,” *Communications of the ACM*, vol. 64, no. 7, pp. 58–65, 2021.
- [11] J. C. Martinez-Sevilla, A. Rosello, D. Rizo, and J. Calvo-Zaragoza, “On the performance of optical music recognition in the absence of specific training data,” in *Proceedings of the 24th International Society for Music Information Retrieval Conference, ISMIR 2023, Milan, Italy, November 5-9, 2023*, 2023, pp. 319–326.
- [12] L. Tuggener, R. Emberger, A. Ghosh, P. Sager, Y. P. Satyawan, J. Montoya, S. Goldschagg, F. Seibold, U. Gut, P. Ackermann *et al.*, “Real world music object recognition,” *Transactions of the International Society for Music Information Retrieval*, vol. 7, no. 1, pp. 1–14, 2024.
- [13] J. Calvo-Zaragoza, A. H. Toselli, and E. Vidal, “Handwritten music recognition for mensural notation with convolutional recurrent neural networks,” *Pattern Recognition Letters*, vol. 128, pp. 115–121, 2019.
- [14] P. Torras, A. Baró, L. Kang, and A. Fornés, “On the integration of language models into sequence to sequence architectures for handwritten music recognition,” in *Proceedings of the 22nd International Society for Music Information Retrieval Conference*. Online: ISMIR, Nov. 2021.
- [15] Y. Li, H. Liu, Q. Jin, M. Cai, and P. Li, “Tromr: Transformer-based polyphonic optical music recognition,” in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.
- [16] A. Ríos-Vila, D. Rizo, J. M. Iñesta, and J. Calvo-Zaragoza, “End-to-end optical music recognition for pianoform sheet music,” *International Journal on Document Analysis and Recognition (IJ DAR)*, vol. 26, no. 3, p. 347–362, 2023.
- [17] J. C. Martinez-Sevilla, A. Rios-Vila, F. J. Castellanos, and J. Calvo-Zaragoza, “A holistic approach for aligned music and lyrics transcription,” in *International Conference on Document Analysis and Recognition*. Springer, 2023, pp. 185–201.
- [18] A. Pacha, “Incremental supervised staff detection,” in *Proceedings of the 2nd international workshop on reading music systems*, 2019, pp. 16–20.
- [19] A. Graves, S. Fernández, F. J. Gomez, and J. Schmidhuber, “Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks,” in *Proceedings of the Twenty-Third International Conference on Machine Learning, (ICML 2006), Pittsburgh, Pennsylvania, USA, June 25-29, 2006*, 2006, pp. 369–376.
- [20] R. Girdhar, A. El-Nouby, Z. Liu, M. Singh, K. V. Alwala, A. Joulin, and I. Misra, “Imagebind one embedding space to bind them all,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*. IEEE, 2023, pp. 15 180–15 190.
- [21] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” in *3rd Int. Conf. on Learning Representations*, Y. Bengio and Y. LeCun, Eds., San Diego, USA, 2015.
- [22] E. Parada-Cabaleiro, A. Batliner, and B. W. Schuller, “A diplomatic edition of il lauro secco: Ground truth for OMR of white mensural notation,” in *Proceedings of the 20th International Society for Music Information Retrieval Conference, ISMIR 2019, Delft, The Netherlands, November 4-8, 2019*, 2019, pp. 557–564.

- [23] M. E. Thomae, J. E. Cumming, and I. Fujinaga, “Digitization of choirbooks in guatemala,” in *Proceedings of the 9th International Conference on Digital Libraries for Musicology*, ser. DLfM ’22. New York, NY, USA: Association for Computing Machinery, 2022, p. 19–26.
- [24] J. Calvo-Zaragoza, D. Rizo, and J. M. I. Quereda, “Two (note) heads are better than one: Pen-based multimodal interaction with music scores,” in *Proceedings of the 17th International Society for Music Information Retrieval Conference, ISMIR 2016, New York City, United States, August 7-11, 2016*, 2016, pp. 509–514.
- [25] A. Madueño, A. Rios-Vila, and D. Rizo, “Automatized incipit encoding at the andalusian music documentation center,” in *Proceedings of the 9th International Conference on Digital Libraries for Musicology*, ser. DLfM ’21, 2021.
- [26] L. Pugin, R. Zitellini, and P. Roland, “Verovio: A library for engraving MEI music notation into SVG,” in *Proceedings of the 15th International Society for Music Information Retrieval Conference, ISMIR 2014, Taipei, Taiwan, October 27-31, 2014*, 2014, pp. 107–112.
- [27] J. Calvo-Zaragoza and D. Rizo, “Camera-primus: Neural end-to-end optical music recognition on realistic monophonic scores,” in *Proceedings of the 19th International Society for Music Information Retrieval Conference, ISMIR 2018, Paris, France, September 23-27, 2018*, 2018, pp. 248–255.
- [28] A. Ríos-Vila, M. Esplà-Gomis, D. Rizo, P. J. Ponce de León, and J. M. Iñesta, “Applying automatic translation for optical music recognition’s encoding step,” *Applied Sciences*, vol. 11, no. 9, 2021.
- [29] A. Awasthi and S. Sarawagi, “Continual learning with neural networks: A review,” in *Proceedings of the ACM India Joint International Conference on Data Science and Management of Data*, 2019, pp. 362–365.

CONTROLLING SURPRISAL IN MUSIC GENERATION VIA INFORMATION CONTENT CURVE MATCHING

Mathias Rose Bjare¹

Stefan Lattner²

Gerhard Widmer^{1,3}

¹ Institute of Computational Perception, Johannes Kepler University Linz, Austria

² Sony Computer Science Laboratories (CSL), Paris, France

³ LIT AI Lab, Linz Institute of Technology, Austria

mathias.bjare@jku.at

ABSTRACT

In recent years, the quality and public interest in music generation systems have grown, encouraging research into various ways to control these systems. We propose a novel method for controlling surprisal in music generation using sequence models. To achieve this goal, we define a metric called Instantaneous Information Content (IIC). The IIC serves as a proxy function for the perceived musical surprisal (as estimated from a probabilistic model) and can be calculated at any point within a music piece. This enables the comparison of surprisal across different musical content even if the musical events occur in irregular time intervals. We use beam search to generate musical material whose IIC curve closely approximates a given target IIC. We experimentally show that the IIC correlates with harmonic and rhythmic complexity and note density. The correlation decreases with the length of the musical context used for estimating the IIC. Finally, we conduct a qualitative user study to test if human listeners can identify the IIC curves that have been used as targets when generating the respective musical material. We provide code for creating IIC interpolations and IIC visualizations on <https://github.com/muthissar/iic>.

1. INTRODUCTION

In music generation, controlling the generation process with user inputs is essential for creating flexible systems that support a creative human/machine co-creation process [1]. Typically, controls are based on low-level features with a direct musical interpretation, for instance, the pitch of a generative synthesizer [2], or meter, harmony, and instrumentation for symbolic generation [3]. A high-level musical feature that has received little attention in generative composition systems is musical surprisal — how surprising a musical event is to a listener, given the past musical context. The surprisal tends to be high when the

music is complex, when a pitch deviates from the prevailing tonality, or when there is a variation in rhythm [4, 5]. As such, musical surprisal shares similarities with musical complexity, however, it is importantly also affected by *learning*: Repeating complex musical content can lead to decreased surprisal on the repetitions as a result of learning [6]. In contrast, the musical content and the complexity remain unchanged across repetitions.

Studies suggest that the amount of musical surprisal needs to be balanced for music to be deemed preferable [7, 8], which is typically achieved by balancing regularity and novelty [9]. Being able to control surprisal in generated music might help users create compositions that balance regularity and novelty and thus suit listeners' preferences. In addition, if this can be controlled, rather low surprisal could be used indirectly to induce repetitions in machine-generated music and high surprisal to produce novel parts, possibly with high perceived complexity.

In [10], it was proposed to quantify the surprisal of a musical event by its Information Content (IC) conditioned on past musical events. For that, a sequence of musical events is modeled as a stochastic process, where the conditional distribution and, hence, the conditional IC can be estimated. As such, a surprising event is an event that is unlikely to occur under the estimated distribution given the past musical context. In the works of [11], the authors find correlations between the IC of a variable-order Markov model (called IDyOM) [12] and perceived surprise in a controlled pitch anticipation experiment. A correlation between high IC and tonal and rhythmic complexity was shown in [4, 5].

This indicates that the IC of trained sequence models can be used as a proxy for human perception of musical surprisal and that its measurement can identify musical complexity and regularities. This paper proposes a novel framework for generating music with user control over the IC. Specifically, we define an *Instantaneous Information Content* (IIC) measure, which can be calculated at any time point based on the IC of musical events in the recent past and approximates a causal information density. We use the IIC as a fitness score to direct a beam search toward generating samples following a given IIC target curve. Our sampling strategy can be used with any pretrained autoregressive generative music model. We demonstrate our approach in symbolic classical music generation using a



pretrained PIA model [13] and show quantitatively that our approach can generate samples that follow IIC curves extracted from real data. We conducted a qualitative study to test if humans can identify simple IIC curves used for generation. Finally, we analyze relationships between IIC and harmonic, rhythmic, and note density complexity.

2. METHODS

In the following, we describe a method for IC-controlled token sequence generation. Let $\text{IC}^*(t)$ be a target curve with support in the time interval $[0, T]$, representing the desired information content over time of a generated sequence of tokens $\mathbf{x} = x_1, x_2, \dots, x_n \in \mathcal{X}$, with a duration of T seconds. ‘Tokens’ are not necessarily individual notes or note onsets but can be any token type commonly used in Transformer-based music generation systems (e.g., [13–15]). Also, note that we operate on the physical time dimension, not symbolic (score) time measured, e.g., in beats or number of tokens.

Furthermore, let q be a generative sequence model and p an autoregressive *critic model*, used for estimating the the i ’th token’s conditional token information content

$$\text{IC}(x_i|x_{<i}) = -\log p(x_i|x_{<i}), \quad (1)$$

where $x_{<i} = x_1, x_2, \dots, x_{i-1}$. In our context, p will be a Transformer model. The proposed method creates new samples using q with an information content that matches the target curve as measured by p . Our method works as follows: Firstly, we define the *Instantaneous Information Content* (IIC) – a mapping from a (temporally irregular) token sequence and its information content values to a function representing the musical surprisal in the continuous time domain. Secondly, we define an *IC deviation* – a metric for comparing the similarity between a sequence’s IIC curve and the target curve. Finally, we devise a method for generating token sequences with q that minimize the *IC deviation*.

2.1 Instantaneous Information Content

2.1.1 Temporal Localization of IC Estimates

To align the information content of musical events, measured on sequence tokens, with the time-domain target IC (IC^*), we face a challenge: IC is calculated on sequence elements, while IC^* pertains to the time domain. Our solution involves assigning each token a temporal position using a mapping function f , effectively “temporally localizing” or aligning tokens within the musical timeline. Note that f can be constructed by analyzing the specific detokenization method associated with \mathbf{x} ’s tokenization that involves turning a sequence of tokens into a time-based music representation like MIDI¹. In section 3.1, we present an example of such f using the tokenization of [13].

Temporal Localization allows us to map IC tokenizations to their respective time points in the music. This is crucial, especially for analyzing tokenizations of symbolic

music commonly used with Transformers [13–15], where the decoded musical events do not uniformly align in time. Through this approach, IC measured on tokens can be directly compared with the time-domain IC^* , facilitating a coherent analysis across different domains of musical representation.

2.1.2 Interpolation

Let $f : \mathbb{N} \times \mathcal{X} \rightarrow \mathbb{R}$ be a localization function, mapping the i ’th token of sequence $\mathbf{x} \in \mathcal{X}$ to the time domain. The IIC at time t in a piece (represented by token sequence \mathbf{x}), is a real number computed by a time interpolation of \mathbf{x} ’s token ICs:

$$\text{IIC}(t, \mathbf{x}) = \sum_{f(i, \mathbf{x}) < t} \lambda(t - f(i, \mathbf{x}), i) \cdot \text{IC}(x_i|x_{<i}). \quad (2)$$

$\lambda(t, i)$ defines a weighting of the information of the i ’th token and the constraint $f(i, \mathbf{x}) < t$ ensures causality. As a result, the IIC at any time step t is a weighted sum of IC values of past events, using a weighting kernel λ .

The choice of the critic model p in combination with the weight function λ defines different perceptual models of the instantaneous information content. We propose to choose λ so that the recent past is weighted higher than the remote past. More specifically, we define λ as a window function centered around t and equal to zero at time steps greater than t . In this initial work, we chose a Hann window for the following reasons: As it is (half) bell-shaped, it is insensitive to inaccuracies in the temporal localization of recent events. It is smooth at the boundaries, preventing sudden drops as events “leave” the window.

Using the IIC, we quantify the *segment surprisal* of segment $[t_1, t_2]$ by the L^1 norm of the IIC with support restricted to $[t_1, t_2]$ by calculating:

$$\|\text{IIC}|_{t_1}^{t_2}\|_1 = \int_{t_1}^{t_2} |\text{IIC}(t, \mathbf{x})| dt. \quad (3)$$

In section 3.5, we compare segment surprisal with segment-based complexity metrics.

2.2 IC Deviation

Given a sample \mathbf{x} , the *IC deviation* of $\text{IIC}(\cdot, \mathbf{x})$ from the target IC^* is defined as the L^1 norm of their function difference:

$$\|\text{IC}^* - \text{IIC}\|_1 = \int_0^T |\text{IC}^*(t) - \text{IIC}(t, \mathbf{x})| dt. \quad (4)$$

Which is equal to zero if $\text{IC}^* = \text{IIC}(\cdot, \mathbf{x})$ almost everywhere, implying that minimizing eq. (4), aligns the target curve IC^* with the IIC curve. In practice, we compute eq. (4) by the Riemann sum:

$$\|\text{IC}^* - \text{IIC}\|_1 \approx \sum_{i=1}^m |\text{IC}^*(t_i) - \text{IIC}(t_i, \mathbf{x})| \Delta t, \quad (5)$$

¹<https://midi.org/midi-1-0-detailed-specification> where $m\Delta t = T$.

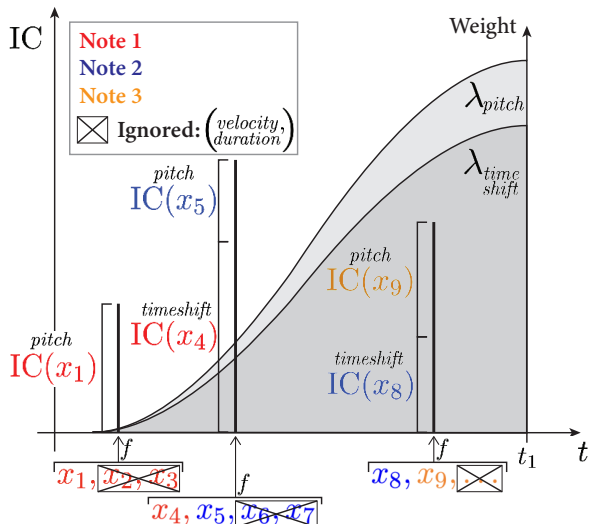


Figure 1. The temporal localization function f and the weight function λ , involved in computing the IIC of x_1, x_2, \dots , a sequence of three notes, at time t_1 .

2.3 Information Content Conditioned Sampling

We can now rank sequences of different lengths according to their proximity to the target IC^* using eq. (5). We use this to guide a beam search to follow the target curve. The beam search is done in iterations. At each iteration, we generate k continuations of the best-performing sample from the last iteration (initially the empty sequence) in parallel. We stop expanding the continuation when the duration of the newly generated content exceeds a predefined step size t' . We then evaluate eq. (5) and keep only the continuation with the lowest IC deviation for the next iteration². We stop when the generation’s duration is T .

3. EXPERIMENTS

3.1 Model and Data

All experiments are performed with a PIA Transformer model [13], a symbolic music generation system pretrained on expressive classical piano performances. The model was trained on data consisting of 1,184 MIDI files of expressive music recorded with high precision on a Yamaha Disklavier [16], as well as a larger dataset of 10,855 MIDI files containing automatically transcribed piano performances [17]. For evaluation, we use the dataset of [18], consisting of performances of 36 Mozart piano sonata movements. The midi files are tokenized using a *structured MIDI encoding* [13], where midi notes, sorted by their onset times, are serialized successively using four tokens *Pitch*, *Velocity*, *Duration*, *Timeshift* in that order. Therefore, every fourth token represents the same token type. *Pitch* is an integer describing the 88-note pitch values on the piano. *Velocity* is an integer describing the 128 possible midi velocity values. *Duration* is an integer representing quantized note duration in seconds:

² Practically, in beam search iteration i , we evaluate the integral of eq. (4) from 0 to it' .

$\{0.02, 0.04, \dots, 1.0, 1.1, \dots, 5.0, 6.0, \dots, 19.0\}$. *Timeshift* is an integer encoding the inter-onset intervals (IOI, i.e., the time durations between subsequent note onsets). *Timeshift* is quantized similarly to the duration token, with the addition of an extra symbol representing a time shift of zero, allowing the model to understand that notes less than 0.02 seconds apart are to be played concurrently. In contrast to the PIA model described in [13], which does non-causal inpainting, we use a causal Transformer based on the Perceiver IO architecture [19] and do continuation generation³. We make these modifications such that the IC calculations ignore future observations. We use the same pretrained model both as the *generator* model q and the *critic* model p and leave the exploration of other critic models for future work.

3.2 IIC

The elements involved in computing the IIC are given in fig. 1. For IIC calculations, we choose to consider only the surprisal of *Pitch* and *Timeshift* tokens, such that the token’s IC represents the surprisal of pitches and IOI. We ignore *Velocity* and *Duration* tokens because they contribute less to the perception of surprise, being mostly related to the performance dimensions dynamics and articulation. This is achieved in the IIC calculation by setting $\lambda(t, i) = 0$ for $i = 2, 6, 10, \dots$ and $i = 3, 7, 11, \dots$ in eq. (2). We choose f such that the pitch token contributes to the surprisal function at its note onset time⁴, and the timeshift token contributes to its surprisal at the onset of the following note (as an IOI is perceived at the onset of the next note). The remaining weights are then defined by the scaled half Hann window

$$\lambda(t, i) = \begin{cases} c_i \frac{1}{L} \cos^2\left(\frac{\pi t}{L}\right) & \text{for } 0 < t < \frac{L}{2}, \\ 0 & \text{otherwise} \end{cases}, \quad (6)$$

where c_i is a weight that takes on two different values for the pitch and timeshift tokens, respectively. c_i is used to weigh the IC of pitches and timeshifts, respectively. For both token types to have equal importance, we estimate a normalization constant empirically by calculating a mean IC over all tokens of the evaluation dataset. The window length is chosen to be $L = 4$ so that the weight is zero after 2 seconds.

3.3 Beam Search Parametrization Study

Using the beam search strategy described in section 2.3, we run initial experiments to determine the effect of parameters associated with the beam search on the similarity between generated samples and target curves IC^* extracted from real music. Specifically, we randomly select 400 snippets of the MIDI files (10 seconds long) and create the IIC curve associated with those snippets. Then, we generate four new samples using our beam search and evaluate the IC deviation between the IIC curve induced by the

³ The model generates sequences using an initial context of real music.

⁴ The note onset times are found by accumulating the time values associated with previous *Timeshift* tokens.

real data and the generated data. We discretize the integral in the IC deviation (see eq. (5)) with $\Delta t = 0.1s$.

To investigate the effect of the step size t' , we fix the number of continuations generated in parallel to $k = 16$ to reduce computation. To investigate the importance of the number of parallel generated samples k , we use a fixed step size of $t' = 0.3s$.

We find that in cases where the generation model q and the critic model p are the same ($p = q$), it is challenging to sample a single continuation $x_i, x_{i+1}, \dots, x_{i+m}$ (using q) that has a high segment surprisal $\|\text{IIC} \left| \frac{f(x_{i+m})}{f(x_i)} \right\|_1$ (measured by p), precisely because the probability of sampling such a continuation is low.

To sample low-probability tokens more efficiently, we propose a heuristic that alters the entropy⁵ of the generating distribution $H(q)$ using a temperature parameter dynamically set using the IIC. Specifically, in iteration $i-1$ of the beam search, we measure $\text{IC}^*(it')$, the target IC at the time where the generation of the continuations halts next time, and calculate a target entropy:

$$H_{target} = \min \left(\frac{\text{IC}^*(it')}{C_H}, H_{max} \right), \quad (7)$$

where C_H is a constant parameter to be estimated and H_{max} is the entropy of the uniform distribution. We then fix q 's entropy to the target entropy H_{target} by searching for a temperature r such that $H_{target} = H(q) = H(\text{softmax}(l/r))$ with binary search, where l are the logits of the neural network. Note that temperature is only used for the generator q and not for the critic model p .

3.4 Qualitative Evaluation

We conducted an online user study to investigate if the IIC curves computed on generated and real music correspond to users' experience of being musically surprised.

Firstly, we present the participant with a musical section generated by our method using one of five target curves. The participant is then tasked to select the IIC curve that best describes their perceived surprise when listening to the section. Secondly, we present the user with a segment of real music and IIC curves extracted from real music, one of which corresponds to the music segment. The user's task is to identify the corresponding curve.

The experiment is conducted on a website that, after an initial experiment description, asks the user for their years of musical training (more or less than five years). Then, it shows an example of a generated piano music section and the surprisal curve used as a target for the generation (together with a textual explanation).

The participant is then presented with five pages, like the one in fig. 2. Each presents a musical section generated using one of five simple target curves. The participant is asked to identify which of the five curves they think has been used to generate the section. The final page contains a 10-second segment of real piano music from the evaluation set and two IIC curves, one corresponding to the piano

music and the other to a randomly selected 10-second segment from the evaluation dataset.

The samples for the first five pages are generated as follows: As contexts for the model, we select the first 13 measures of Mozart K.331, 1st mvt. and the first 16 measures of K.332, 2nd mvt. from the evaluation dataset and generate 200 samples for every combination of the two musical contexts and the five IIC curves shown on the page, with $C_H = 50$, $t' = 0.3s$ and $k = 128$ (i.e., the optimal beam search parameters, as shown in table 1). For each combination, we then select the 25 samples with the lowest IC deviation for the user study. For the final page of the user study related to real performances, we select 300 different 10-second segments from the evaluation dataset and compute the IIC curves. The results of the user study will be presented and discussed in section 4.2

3.5 Analysis of IIC

As discussed in the introduction, IC and surprisal might be related to aspects of musical complexity, but learning effects may lead to a decrease in surprisal in passages with repeated musical content. To investigate these relationships, we designed an experiment to determine if the IIC correlates with harmonic complexity, as quantified by *tonal tension (cloud diameter)* [20], where the IIC is calculated using progressively larger segments of musical context. Tonal tension is calculated for a segment of music by considering its most dissonant pitch class interval, where an interval dissonance is measured as the distance between the interval pitches embedded in a specific Euclidian space where the position is based on the circle of fifths [21].

We extracted one-second segments centered on the onsets of notes in the evaluation dataset. For $i = 1, \dots, 1000$, we then compute the Pearson correlation coefficient between the tonal tension and the segment surprisal (see eq. (3)) of the first i segments within every performance.

In addition, we investigate complexity in terms of note density, i.e., the number of notes per segment. To do so, we use the same setup as for tonal tension but count the number of notes within one-second segments.

Finally, we investigate rhythmical complexity using the *IOI histogram entropy* of measures [22]. We choose this measure over other structural rhythmical complexity measures [23–28] since it does not assume the rhythm to be cyclic. We follow the same procedure as mentioned above, but instead of selecting fixed-sized segments centered around note-onsets, we select segments of one measure based on the measure annotations [18]. More specifically, we match the notes of the performance with its score notes and extract for each measure: 1) the normalized entropy of the score notes IOI histogram and 2) the segment IIC of the measure normalized with the length of the measure. The segment boundaries are estimated by the mean onset time of the first and last note in subsequent measures.

⁵ Entropy is the expectation of IC.

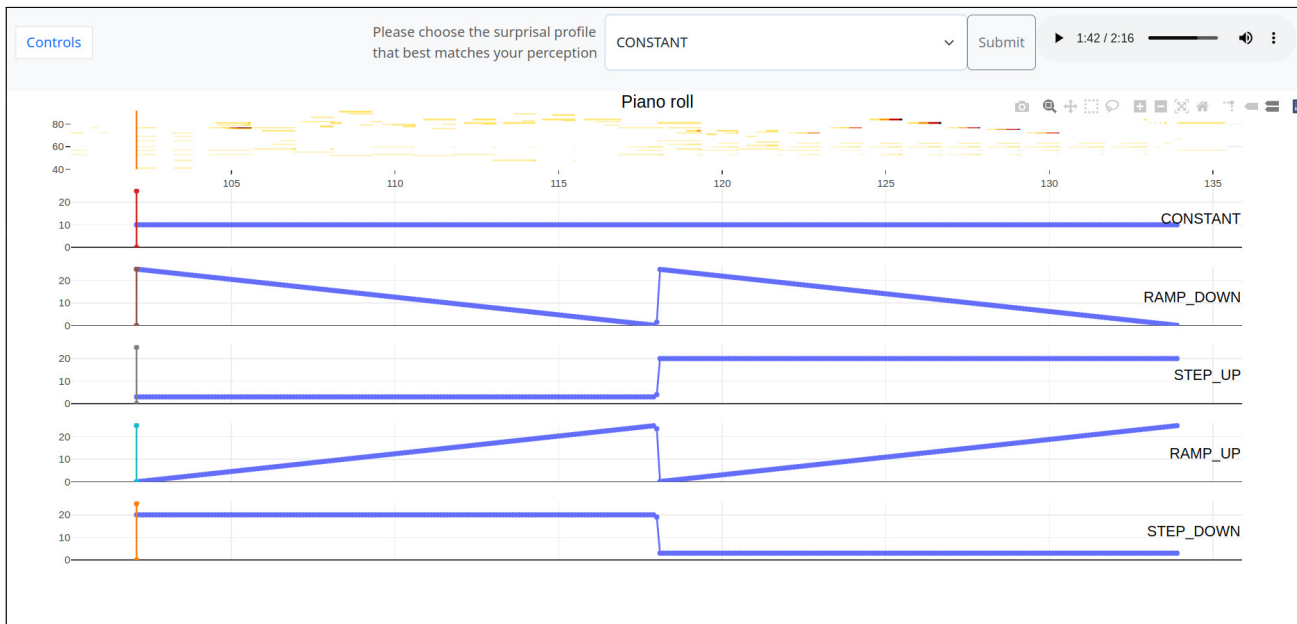


Figure 2. Example page of the user study with a generated musical section and five target curves to choose from.

t'	0.1s	0.2s	0.3s	0.4s	0.5s	0.6s	0.7s	0.8s	1.0s	2.0s
IC dev.	3.63	2.71	2.61	2.72	2.69	2.93	3.03	3.11	3.31	3.90
k	1	2	4	16	32	64	96	128		
IC dev.	8.41	5.90	4.36	2.61	2.14	1.89	1.76	1.69		
C_H	10	20	30	40	50	60	70	80	120	No
IC dev.	8.33	4.11	2.67	2.21	2.15	2.15	2.25	2.45	3.12	2.61

Table 1. IC deviation between target curves IC* extracted from real music, and IIC curves from continuations generated with different beam search parameters.

4. RESULTS

4.1 Beam Search Parameter Study

In table 1, we report the mean IC deviation of samples generated with different beam search step sizes t' , numbers of continuations generated in parallel k and C_H , constants used for setting the softmax temperature dynamically. Bigger step sizes create longer continuations with high IC deviation variance, resulting in worse performance. The lowest values ($t' = 0.1s, 0.2s$) also worsen IC deviation, likely because sampled notes exceed the timestep, causing inaccuracies in the next beam search iteration. For the number of continuations generated in parallel k , we find that the IC deviation always decreases with higher k . This is not surprising as the model has more candidate continuations to choose from. The decrease flattens out as seen by the small IC deviation differences when $k \geq 64$. For the dynamic temperature, we find that $C_H = 50, 60$ reduces the IC deviation compared to using no temperature scaling (marked with "No" in table 1).

4.2 Qualitative Results

The user study results reported as a binary classification of finding the correct curve, among the curves described

in section 3.4, are presented in Table 2. 29 users participated, 23 participants had more than 5 years of musical training, and 6 participants had less than 5 years of experience. 152 generated samples and 21 samples of real music were classified in total. Due to the imbalance in the number of untrained and trained participants and since we found little difference in the classification performance between the groups, we combined their results in the table.

The overall F_1 -score was reported as 0.52 for generated data and 0.71 for real data, which is reasonably above the proportions 0.2 and 0.5, being the F_1 scores of random classifiers, with 5 and 2 classes respectively. The results for the individual curves show difficulty differences in classifying the different curve types, with RAMP_DOWN having the lowest F_1 -score of 0.41 and STEP_UP having the highest F_1 -score of 0.71. We therefore investigate the confusion of curves in Figure 3. We find that the confusion of CONSTANT is evenly distributed on all curves, except for STEP_UP, which is reasonable since CONSTANT does not share any characteristics with the other curves. We furthermore find that generations that start with the same IIC value, either high or low, are confused. This is seen by the confusion of RAMP_DOWN with STEP_DOWN and the confusion of STEP_UP and RAMP_UP.

	IIC Curves					Gen. all curves	Real
	CONSTANT	RAMP_DOWN	STEP_UP	RAMP_UP	STEP_DOWN		
F_1	0.53	0.41	0.71	0.48	0.49	0.52	0.71
#True	36	34	29	29	24	152	21
#Pred	36	33	31	30	22	152	21

Table 2. Results from the user study reported as F_1 -score of identifying the: IIC curve used for generation, the IIC curve of real music.

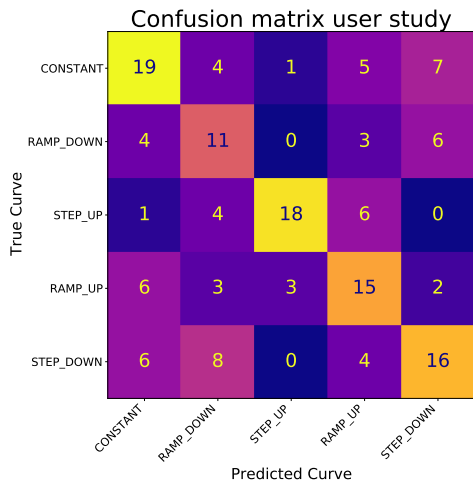


Figure 3. The confusion matrix for users identifying the IIC* curves used to generate the music examples.

4.3 Analysis of IIC

The correlations between IIC and the tonal tension tt , note density d , and the IOI histogram entropy he were calculated on the first n segments of the 36 evaluation data performances as described in section 3.5 and reported in fig. 4. We report the results for IIC calculated using *Pitch* only, *Timeshift* only, or both token types. For tt , d , and $he_{Timeshift}$, the correlations reported were found significant using a significance level of 0.05, whereas for he_{Pitch} and he_{Both} , the correlations are not significant.

The results show a moderate to high correlation of IIC with all metrics at the beginning of the performances (when n is small). However, these correlations decrease in later parts of the performances (when n is high), likely due to “learning” (simulated by longer context) over time.

The highest correlations are found for note density d . This may be explained by the definition of IIC (see eq. (2)) as a weighted sum of token ICs since more tokens per segment simply lead to higher sums.

Considering the different token type combinations, we find that tt is most correlated with IIC calculated using only *Pitch* tokens and he using only *Timeshift* tokens. This is reasonable, considering that very dissonant segments and very complex rhythms tend to be associated with *Pitch* and *Timeshift* tokens, respectively, which are infrequent in the training dataset, resulting in a high token IC. Interestingly, tt is also correlated with IIC calculated using only *Timeshift* tokens (encoding IOIs), which

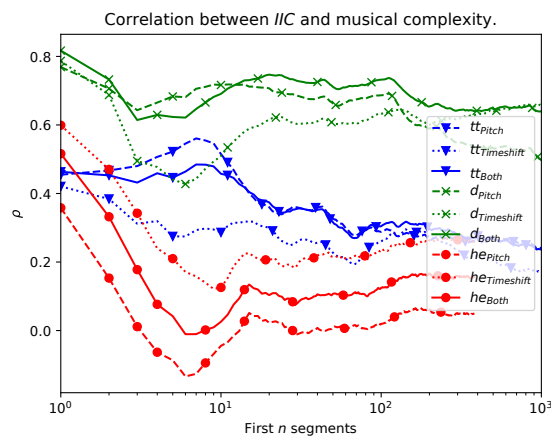


Figure 4. Correlation between IIC and tonal tension tt , note density d , and IOI histogram entropy (he).

might stem from the critic model facing greater uncertainty in predicting any token type when confronted with a highly harmonic complex context that is infrequent in the dataset.

The curves of *Both* and *Pitch* follow each other closely for total tension and note density, indicating that using both *Timeshift* and *Pitch* tokens does not significantly reduce the complexity correlations. For rhythmical complexity, using *Both* tokens instead of *Timeshift* tokens alone decreases the correlation more.

5. CONCLUSION

In this study, we introduced a novel framework for controlling musical surprisal through Instantaneous Information Content (IIC), which maps token-based surprisal to a continuous time-domain function. Using a beam search algorithm, we demonstrated that our approach can generate music that closely follows predefined IIC curves, effectively aligning generated and target surprisal curves.

Our user study confirmed that participants could reasonably identify target IIC curves from generated music, indicating that our method captures perceptible aspects of musical surprise. Furthermore, our analysis showed that IIC correlates with measures of musical complexity such as tonal tension and note density.

Future work will explore alternative critic models, like personalized models, trained on music that is familiar to the user or models with smaller context windows to more directly control local musical complexity.

6. ACKNOWLEDGMENTS

The work leading to these results was conducted in a collaboration between JKU and Sony Computer Science Laboratories Paris under a research agreement. GW's work is supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme, grant agreement 101019375 ("Whither Music?").

7. REFERENCES

- [1] C. A. Huang, H. V. Koops, E. Newton-Rex, M. Dinulescu, and C. J. Cai, "Human-ai co-creation in song-writing," in *ISMIR*, 2020, pp. 708–716.
- [2] J. Engel, C. Resnick, A. Roberts, S. Dieleman, M. Norouzi, D. Eck, and K. Simonyan, "Neural audio synthesis of musical notes with wavenet autoencoders," in *International Conference on Machine Learning*. PMLR, 2017, pp. 1068–1077.
- [3] D. von Rütte, L. Biggio, Y. Kilcher, and T. Hofmann, "FIGARO: Controllable music generation using learned and expert features," in *ICLR*. OpenReview.net, 2023.
- [4] S. A. Sauv e and M. T. Pearce, "Information-theoretic modeling of perceived musical complexity," *Music Perception: An Interdisciplinary Journal*, vol. 37, no. 2, pp. 165–178, 2019.
- [5] M. R. Bjare, S. Lattner, and G. Widmer, "Exploring sampling techniques for generating melodies with a transformer language model," in *ISMIR*, 2023, pp. 810–816.
- [6] —, "Differentiable short-term models for efficient online learning and prediction in monophonic music," *Trans. Int. Soc. Music. Inf. Retr.*, vol. 5, no. 1, p. 190, 2022.
- [7] M. T. Pearce and G. A. Wiggins, "Auditory expectation: The information dynamics of music perception and cognition," *Top. Cogn. Sci.*, vol. 4, no. 4, pp. 625–652, 2012.
- [8] T. E. Matthews, M. A. Witek, O. A. Heggli, V. B. Penhune, and P. Vuust, "The sensation of groove is affected by the interaction of rhythmic and harmonic complexity," *PLoS One*, vol. 14, no. 1, p. e0204539, 2019.
- [9] I. Zioga, P. M. C. Harrison, M. T. Pearce, J. Bhattacharya, and C. D. B. Luft, "From learning to creativity: Identifying the behavioural and neural correlates of learning to predict human judgements of musical creativity," *NeuroImage*, vol. 206, 2020.
- [10] L. B. Meyer, "Meaning in music and information theory," *The Journal of Aesthetics and Art Criticism*, vol. 15, no. 4, pp. 412–424, 1957.
- [11] M. T. Pearce, M. H. Ruiz, S. Kapasi, G. A. Wiggins, and J. Bhattacharya, "Unsupervised statistical learning underpins computational, behavioural, and neural manifestations of musical expectation," *NeuroImage*, vol. 50, no. 1, pp. 302–313, 2010.
- [12] D. Conklin and I. H. Witten, "Multiple viewpoint systems for music prediction," *Journal of New Music Research*, vol. 24, no. 1, pp. 51–73, 1995.
- [13] G. Hadjeres and L. Crestel, "The piano inpainting application," *CoRR*, vol. abs/2107.05944, 2021.
- [14] C. A. Huang, A. Vaswani, J. Uszkoreit, I. Simon, C. Hawthorne, N. Shazeer, A. M. Dai, M. D. Hoffman, M. Dinulescu, and D. Eck, "Music transformer: Generating music with long-term structure," in *ICLR (Poster)*. OpenReview.net, 2019.
- [15] Y. Huang and Y. Yang, "Pop music transformer: Beat-based modeling and generation of expressive pop piano compositions," in *ACM Multimedia*. ACM, 2020, pp. 1180–1188.
- [16] C. Hawthorne, A. Stasyuk, A. Roberts, I. Simon, C. A. Huang, S. Dieleman, E. Elsen, J. H. Engel, and D. Eck, "Enabling factorized piano music modeling and generation with the MAESTRO dataset," in *ICLR*. OpenReview.net, 2019.
- [17] Q. Kong, B. Li, J. Chen, and Y. Wang, "Giantmidi-piano: A large-scale MIDI dataset for classical piano music," *Trans. Int. Soc. Music. Inf. Retr.*, vol. 5, no. 1, pp. 87–98, 2022. [Online]. Available: <https://doi.org/10.5334/tismir.80>
- [18] P. Hu and G. Widmer, "The batik-plays-mozart corpus: Linking performance to score to musicological annotations," in *ISMIR*, 2023, pp. 297–303.
- [19] A. Jaegle, S. Borgeaud, J. Alayrac, C. Doersch, C. Ionescu, D. Ding, S. Koppula, D. Zoran, A. Brock, E. Shelhamer, O. J. H enaff, M. M. Botvinick, A. Zisserman, O. Vinyals, and J. Carreira, "Perceiver IO: A general architecture for structured inputs & outputs," in *ICLR*. OpenReview.net, 2022.
- [20] D. Herremans and E. Chew, "Tension ribbons: Quantifying and visualising tonal tension," in *Proceedings of the International Conference on Technologies for Music Notation and Representation - TENOR2016*, R. Hoadley, C. Nash, and D. Fober, Eds. Cambridge, UK: Anglia Ruskin University, 2016, pp. 8–18.
- [21] E. Chew, "The spiral array: An algorithm for determining key boundaries," in *ICMAI*, ser. Lecture Notes in Computer Science, vol. 2445. Springer, 2002, pp. 18–31.
- [22] A. A. Moles, *Information theory and esthetic perception*. The University of Illinois Press, Urbana and London, 1966.

- [23] H. C. Longuet-Higgins and C. S. Lee, “The rhythmic interpretation of monophonic music,” *Music Perception*, vol. 1, no. 4, pp. 424–441, 1984.
- [24] L. M. Smith, H. Honing *et al.*, “Evaluating and extending computational models of rhythmic syncopation in music,” in *ICMC*, 2006.
- [25] M. Keith, *From polychords to poly: adventures in musical combinatorics*. Vinculum Press, 1991.
- [26] J. Pressing, “Cognitive complexity and the structure of musical patterns,” in *Proceedings of the 4th Conference of the Australasian Cognitive Science Society*, vol. 4, 1999, pp. 1–8.
- [27] F. Gómez, A. Melvin, D. Rappaport, and G. T. Toussaint, “Mathematical measures of syncopation,” in *Renaissance banff: Mathematics, music, art, culture*, 2005, pp. 73–84.
- [28] A. I. Mezza, M. Zaroni, and A. Sarti, “A latent rhythm complexity model for attribute-controlled drum pattern generation,” *EURASIP J. Audio Speech Music. Process.*, vol. 2023, no. 1, p. 11, 2023.

TOWARD A MORE COMPLETE OMR SOLUTION

Guang Yang¹ Muru Zhang¹ Lin Qiu¹ Yanming Wan¹ Noah A. Smith^{1,2}

¹Paul G. Allen School of Computer Science & Engineering, University of Washington, United States

²Allen Institute for Artificial Intelligence, United States

{gyang1, nasmith}@cs.washington.edu

ABSTRACT

Optical music recognition (OMR) aims to convert music notation into digital formats. One approach to tackle OMR is through a multi-stage pipeline, where the system first detects visual music notation elements in the image (object detection) and then assembles them into a music notation (notation assembly). Most previous work on notation assembly unrealistically assumes perfect object detection. In this study, we focus on the MUSCIMA++ v2.0 dataset, which represents musical notation as a graph with pairwise relationships among detected music objects, and we consider both stages together. First, we introduce a music object detector based on YOLOv8, which improves detection performance. Second, we introduce a supervised training pipeline that completes the notation assembly stage based on detection output. We find that this model is able to outperform existing models trained on perfect detection output, showing the benefit of considering the detection and assembly stages in a more holistic way. These findings, together with our novel evaluation metric, are important steps toward a more complete OMR solution.

1. INTRODUCTION

Optical music recognition (OMR) focuses on converting music notation into digital formats amenable to playback and editing. OMR systems are generally divided into two categories: end-to-end systems (which directly convert the image into music notation) and multi-stage systems. Proposed and refined by [1–3], a standard multi-stage system consists of four stages: preprocessing, music object detection, notation assembly, and encoding. In this study, we focus on the object detection and notation assembly stages.

MUSCIMA++ [4] suggests representing music notation as a graph where each pair of musical symbols is linked by a binary relationship, allowing for clear notation reconstruction. The authors created a dataset of handwritten scores with a bounding box for each music object and a human-annotated graph of object relationships in each image. Notation assembly on MUSCIMA++ can be framed

as a set of binary classification decisions to predict the pairwise relationships between music symbols. Most prior research has explored notation assembly with the assumption of perfect detection output [5], but such assumptions can introduce unwanted biases that deteriorate the performance of the notation assembly system when applied as part of a pipeline. Pacha et al. [6] evaluate a notation assembler on realistic detector output, finding some degradation relative to gold-standard objects, but they do not seek to mitigate the problem.

To improve notation assembly robustness, we propose a training method to complete notation assembly on top of (imperfect) object detection output directly. To have a strong detector to start with, we train YOLOv8 [7] and perform a set of preprocessing steps to adapt the model to the MUSCIMA++ v2.0 dataset. Our detector outperforms previous detectors on MUSCIMA++ v2.0 [8] by 2.4%, establishing a solid foundation for notation assembly.

Traditional evaluation methods, which perform notation assembly over all pairs of ground-truth objects and report an F1 score or a precision-recall curve, become inadequate when the input objects come from imperfect detection. We propose an end-to-end evaluation metric, called Match+AUC, that accounts for both detection errors and assembly errors by first matching detected objects with their ground-truth counterparts before assessing notation assembly accuracy. It complements metrics that evaluate pipeline components individually.

Our code for reproducing all of the experiments is publicly available at <https://github.com/guang-yng/completeOMR>.

2. MULTI-STAGE OMR

We focus on the MUSCIMA++ v2.0 dataset [4] and follow its multi-stage pipeline for the OMR system. This dataset includes 140 high-resolution annotated images out of 1000 images from the CVC-MUSCIMA dataset [9]. It contains 91,254 symbol-level annotations and 82,247 relationship annotations between symbol pairs by human annotators. These annotations span 163 distinct classes of music symbols. Figure 2 shows an example from this dataset.

As the MUSCIMA++ dataset provides symbol-level pairwise relationships, it allows study of two stages of the pipeline: (i) detection and (ii) assembly. In (i), given an image as input, an object detector is used to extract all music symbols in the image, denoted as the set $V = \{v_i\}_i$,



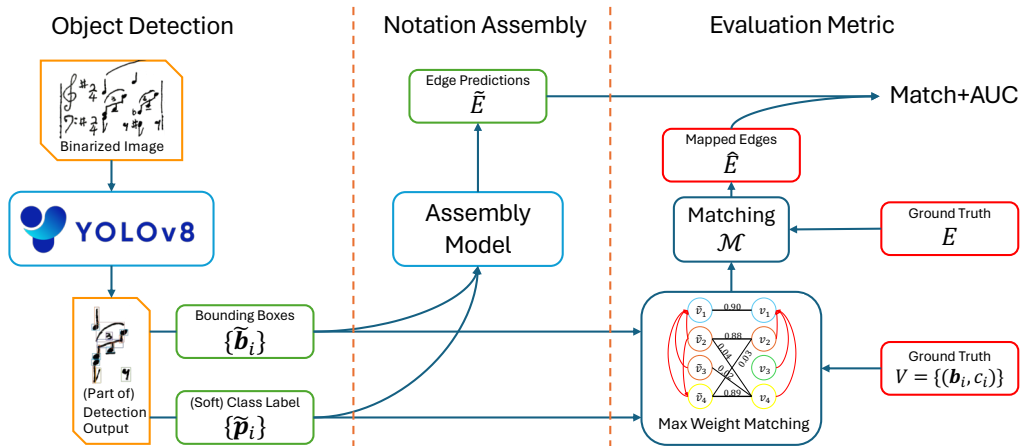


Figure 1. An overview of our OMR pipeline, highlighting key components: object detection, notation assembly, and evaluation metric. Detailed explanations of each component can be found in Subsections 3.1, 3.2, and 3.3 respectively.



Figure 2. Example of a music image (binarized) extracted from the MUSCIMA++ dataset.

where $v_i = (b_i, c_i)$ is a tuple of a bounding box and a class label. Each pair of music symbols (v_i, v_j) is then fed into (ii) the notation assembly model to predict whether or not there exists a relationship between them. The notation assembly stage can be framed as an edge prediction problem where the model needs to output a set of edges E to get a directed graph $G = (V, E)$. MUSCIMA++ defines a grammar over all possible music symbol classes so that the direction of an edge is uniquely determined by the class labels (c_i, c_j) of the vertices (v_i, v_j) . Consequently, the edge prediction problem can be reduced to predicting an undirected graph. The authors of [4] argue that such a graph G enables straightforward reconstruction of the full symbolic music notation, so we do not consider the decoding process after (i) and (ii) in this work.

In previous works, the two stages are considered separately, either focusing on object detection, without fully analyzing its effect on downstream notation assembly [8, 10, 11]; or focusing on notation assembly and assuming perfect detection input during training [5, 6]. This raises the question of whether the best object detector is a good fit for the best notation assembly model. To investigate, we developed an end-to-end metric that evaluates the performance of the entire pipeline, as explained in Section 3.3. We found that, compared with our approach where both stages are considered together—specifically, where the notation assembly model is trained using the output of the object detector—treating the two stages separately leads to poorer results.

3. METHODOLOGY

We describe our method for each stage, and how we connect the two stages together and evaluate the entire pipeline. Figure 1 shows an overview of our methods.

3.1 Music Symbol Detection

A music object detection system analyzes an image to identify each music object it contains, providing both the bounding box and class label for every detected object [10]. Traditionally, this process would begin with an initial stage of image preprocessing, typically aimed at removing staff lines, followed by a second stage focusing on the segmentation and classification of symbols. Thanks to recent advances in computer vision, there are mature solutions for image preprocessing and staff line removal, allowing us to treat it as a largely solved problem [12–14]. In our case, MUSCIMA++ provides us with staff line removed images as input, so we directly build our detectors on top of these images.

Following the work of Zhang et al. [8], we adopted a convolutional neural network-based approach for page-level object detection of handwritten music notes, opting for this approach over segmentation-based methods, because segmentation-based methods often struggle with overlapping symbols. We choose YOLOv8 [7], which is the latest version of YOLO [15], due to its superior performance on traditional computer vision tasks. Compared to YOLOv4 [16], which is used by [8], YOLOv8 has a new loss function and a new anchor-free detection head, achieving higher performance on various detection tasks. YOLOv8 has not yet, to our knowledge, been applied to OMR. Furthermore, since the images of handwritten music notation in MUSCIMA++ have high resolution and music objects are drastically different from the objects considered in computer vision research, directly applying the training strategy of YOLOv8 doesn’t work well. We follow [2, 8, 17] to crop images into small snippets during the training stage to alleviate this issue. Specifically, we randomly crop the images during training and compactly segment the image during inference. More details are pre-

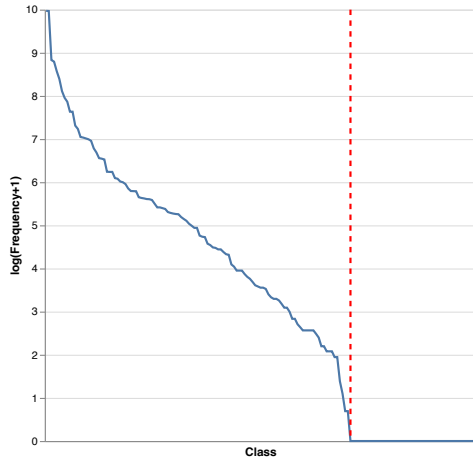


Figure 3. Frequencies of different classes in the dataset, from most- to least-frequent. A long-tailed distribution with 48 classes on the right of the red line that never appear. The y -axis shows the value of $\ln(\text{frequency} + 1)$. The top-5 classes are `stem`, `nodeheadFull`, `ledgerLine`, `beam`, and `staffSpace`.

sented in Section 4.1.2.

The MUSCIMA++ v2.0 dataset includes 163 object classes in total, covering a large variety of notation. However, most of the classes scarcely appear and barely affect the replayability of the OMR output (e.g., the construction of a MIDI file encoding the score). The distribution of classes is shown in Figure 3; 48 classes never appear in the entire dataset. Given this, we manually remove these 48 classes along with some other rare classes, leading to a subset of 73 attested “essential” classes that are observed in the dataset. To get a direct comparison with previous methods, while also keeping a focus on essential classes, we report results using both the full class set and essential classes only. Meanwhile, we also report results on the 20 “primitive” classes selected for evaluation by [8].

3.2 Notation Assembly

The notation assembly model takes a pair of nodes as input, and gives a binary output indicating whether there is a relationship between them. An intuitive method is to first concatenate the features of two nodes, and then pass the pair as a single feature vector through a series of layers of a multi-layer perceptron (MLP). A sigmoid function σ is applied at the end to output the probability that there exists a relationship.

$$\hat{e}_{ij} = \sigma(\phi_{\text{MLP}}([v_i, v_j])) \quad (1)$$

As notation assembly is essentially binary classification, we use binary cross-entropy as our loss function:

$$\mathcal{L}_{\text{BCE}}(\hat{e}_{ij}) = -e_{ij} \log(\hat{e}_{ij}) - (1 - e_{ij}) \log(1 - \hat{e}_{ij}).$$

We adopt the input feature design in [5], where each v_i is represented by its 4-dimensional bounding box and the class label. The class label is passed to an embedding layer with x dimensions. Therefore, the input to MLP will be a $(4 + x) \times 2$ dimensional vector.

Existing work assumes perfect detection output; therefore, the input bounding box and class label are the ground-truth information. While previous work has attempted to manually perturb the bounding box as a test of robustness, such perturbations don’t reflect the kind of errors that might arise in a practical object detector.

To ensure our notation assembly system can adapt to errors introduced in the detection stage, we propose a supervised training pipeline that directly trains the assembly model on detection output \tilde{V} . Since most of the time $\tilde{V} \neq V$, we can’t directly use the ground truth E as the supervision signal.

To deal with this issue, we construct a maximum weight matching M in the bipartite graph $G_M = (\tilde{V}, V)$ and build \hat{E} for supervising our notation assembly model. We describe the detail of our matching procedure in Section 3.3, where it is also employed in evaluation. We adopt the edges from the ground truth according to our matching. Given a pair $(\tilde{v}_i, v_k) \in M$ and an edge $(v_k, v_h) \in E$, we add $(\tilde{v}_i, \tilde{v}_j)$ to \hat{E} if $(\tilde{v}_j, v_h) \in M$. Our method essentially builds a training set for the detection output that is in the same format as the ground-truth, allowing seamless training and evaluation.

3.3 End-to-End Evaluation

The main challenge of OMR evaluation is finding the edit distance between two music scores under some particular representation (e.g., XML format [18]). Hajič [19] argued that intrinsic evaluation is needed to decouple research of OMR methods from individual downstream use-cases, since specific notation formats change much faster than music notation itself. Some works have taken steps to analyze the complexity of standard music notation [20] and propose common music representation formats [21].

As a general system consisting several modules, we seek to also evaluate our OMR pipeline holistically, without a specific focus on what the downstream processing will be. We therefore propose a novel matching-based evaluation metric to assess predictions that include errors from the detection stage. For the same reason we had to adapt ground-truth edges to create training data for the notation assembly model ($\tilde{V} \neq V$), we cannot straightforwardly use the ground-truth graph to evaluate notation assembly. Our metric finds a matching between a test instance’s predicted objects and those in the ground-truth object detection, and then uses this as a bridge to evaluate the edges returned by the notation assembly module.

The results reported by Pacha et al. [6] are the sole benchmark for assessing a notation assembly model using detected symbols. To address the matching issue between \tilde{V} and V , Pacha et al. employ a rule-based method, considering two objects identical if they belong to the same class and their intersection over union is at least 50%. However, this greedy matching approach is inadequate, as inaccuracies in symbol detection cannot be compensated for by the notation assembly model. Furthermore, Pacha et al. use conventional precision/recall metrics with a hard decision boundary, which fails to capture the overall performance

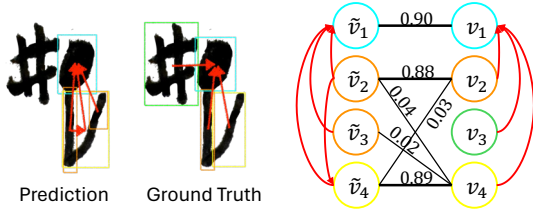


Figure 4. An example of detected objects and predicted graph, alongside ground truth. At the right is the constructed bipartite graph (zero-weight edges not shown). Thick edges represent the matching function \mathcal{M} induced by the matching algorithm. In our notation, $E = \{(v_2, v_1), (v_3, v_1), (v_4, v_1)\}$ and the matching function maps v_1 to \tilde{v}_1 , v_2 to \tilde{v}_2 and v_4 to \tilde{v}_4 . Therefore, $\hat{E} = \{(\tilde{v}_2, \tilde{v}_1), (\tilde{v}_4, \tilde{v}_1)\}$. Because $\tilde{E} = \{(\tilde{v}_2, \tilde{v}_1), (\tilde{v}_2, \tilde{v}_4), (\tilde{v}_3, \tilde{v}_1), (\tilde{v}_4, \tilde{v}_1)\}$, we get a precision of 0.5 and recall of 1.0.

of the model comprehensively. To resolve these issues, we propose a complementary metric based on a global optimal matching and area under the precision-recall curve.

Formally, we denote $\tilde{V} = \{\tilde{v}_1, \tilde{v}_2, \dots, \tilde{v}_n\}$ as the set of symbols obtained from an object detection model, where $\tilde{v}_i = (\tilde{\mathbf{b}}_i, \mathbf{p}_i)$ is a tuple of a bounding box $\tilde{\mathbf{b}}_i \in \mathbb{R}^4$ and a probability distribution vector $\mathbf{p}_i \in \mathbb{R}^C$ over all symbol classes. A notation assembly prediction on \tilde{V} would be an edge set $\tilde{E} = \{\tilde{e}_1, \tilde{e}_2, \dots, \tilde{e}_m\}$ where each edge \tilde{e}_i is a tuple of two vertices. Similarly, we denote the ground truth notation graph as $G = (V, E)$ with $V = \{v_1, v_2, \dots, v_n\}$, $v_i = (\mathbf{b}_i, c_i)$, $E = \{e_1, e_2, \dots, e_m\}$, where $\mathbf{b}_i \in \mathbb{R}^4$ is a bounding box and $c_i \in \{1, 2, \dots, C\}$ is a symbol class label.

We first construct a complete weighted bipartite (\tilde{V}, V) where the weight for edge (\tilde{v}_i, v_j) is $w_{ij} = \text{IoU}(\tilde{\mathbf{b}}_i, \mathbf{b}_j) \cdot \mathbf{p}_{i,c_j}$. Here, IoU is the intersection-over-union between the area occupied by the two boxes, defined as:

$$\text{IoU}(\mathbf{b}_i, \mathbf{b}_j) = \frac{\text{Area}(\mathbf{b}_i \cap \mathbf{b}_j)}{\text{Area}(\mathbf{b}_i \cup \mathbf{b}_j)}.$$

Based on this bipartite graph, we find the maximum weighted matching M using the implementation described in [22] and filter the “weak” matching edges with weight w_{ij} less than a threshold T_{match} to get the matching function $\mathcal{M} : V \rightarrow \tilde{V} \cup \{\emptyset\}$:

$$\mathcal{M}(v_j) = \begin{cases} \tilde{v}_i, & \text{if } (\tilde{v}_i, v_j) \in M \text{ and } w_{ij} > T_{\text{match}}, \\ \emptyset, & \text{otherwise.} \end{cases}$$

Here, T_{match} is a filtering threshold for matching and we set it to 0.05 without tuning.

After getting the matching function, the ground truth assembly edges are naturally mapped back to edges between predicted vertices. The mapped edge set $\hat{E} = \{(\mathcal{M}(v_i), \mathcal{M}(v_j)) \mid (v_i, v_j) \in E, \mathcal{M}(v_i) \neq \emptyset, \mathcal{M}(v_j) \neq \emptyset\}$ represents a ground truth edge set on detected vertices, which can be used to evaluate predictions \tilde{E} to get a precision and recall. An example is shown in Figure 4.

Most notation assembly models predict a probability of the existence of an edge (v_i, v_j) , and the probability is

further compared with a threshold T_{predict} to determine whether (v_i, v_j) belongs to the prediction set \tilde{E} . By adjusting the model prediction threshold T_{predict} , we can get a series of predictions $\{\tilde{E}_1, \tilde{E}_2, \dots\}$ and therefore derive a series of precision-recall pairs, which are used to estimate the area-under-the-curve (AUC) score. We refer to the full evaluation metric as “Match+AUC.”

“Match+AUC” is an end-to-end evaluation metric for the OMR pipeline with following advantages:

- “Match+AUC” accounts for model performance in both the object detection and notation assembly stages. To be specific, given an object detector’s output, a notation assembly model will achieve a higher score if it predicts no edges among redundant objects, since connecting redundant nodes into the assembly graph would greatly affect the final output music score. Also, for the same assembly model, a worse object detector would generate a large amount of redundant and inaccurate objects, making it very hard for the assembly model to distinguish them.
- Instead of a hard rule-based matching used in past methods, “Match+AUC” creates a comprehensive matching among detected symbols and ground truth symbols, making the final score more accurate and sensitive.
- “Match+AUC” evaluates the model using the area under the precision-recall curve, which summarizes performance across a range of threshold choices that could be made by a downstream module or a system user.

We believe that our novel “Match+AUC” is a compelling tool for analyzing OMR pipelines that is complementary to existing approaches.

4. IMPLEMENTATION DETAILS

4.1 Music Symbol Detection

4.1.1 Model Details

We finetune the “large” version of YOLOv8 (YOLOv8l), an object detection model pre-trained on the COCO dataset [23], on MUSCIMA++ v2.0 for music object detection. The model consists of 43.7M parameters and is capable of detecting object bounding boxes and generating corresponding class distributions. The input image size of our model is set to 640.

4.1.2 Training

We used the MUSCIMA++ v2.0 dataset to train and evaluate the music symbol detection model [4]. The images are binarized (pixels are 0/1-valued) and in a size of approximately 3500×2000 pixels. For simplicity, we use images with staff lines removed. Additionally, following the exact method described in [6], we split the dataset into 60% training data, 20% validation data, and 20% test data. To effectively train YOLOv8 on these dense images involving many small annotations, which include augmentation dots and piano pedal markings, we have to reduce the image size. Therefore, following the methods used by [8],

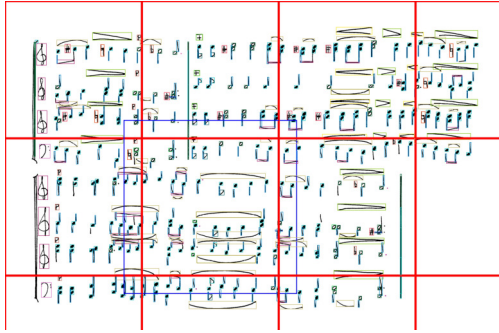


Figure 5. Example of music symbol detection segments for inference. The thick red line indicates the primary cropped area, while the thick blue line represents an extended cropped section designed to include partial symbols that may extend beyond the main cropped area. For better visualization, we only show the extended area of one image crop. Image crops on the right and bottom border of the page are padded to fit into YOLOv8.

given a large music score image, we randomly sample $14\ 1216 \times 1216$ crops and further resize them in to 640×640 to fit the YOLOv8 input requirement.

We fine-tune the YOLOv8 model for 500 epochs with a batch size of 8. We use the AdamW optimizer with a learning rate of 5.5×10^{-5} and a momentum of 0.9, which are automatically set by the YOLOv8 codebase [7]. During training, we use the early stopping strategy with a patience of 100 epochs. We keep the checkpoint with the highest validation performance as our final model.

4.1.3 Inference

Since our detector is trained on cropped data, during the inference stage, we also need to segment the large images into smaller segments. However, partial objects at the edges of these crops would be hard to detect since the model can't see the full object. To resolve this issue, we extend every crop with a margin, which serves as a context for each image. The cropping is visualized in an example in Figure 5. We then perform symbol detection on each extended crop and consolidate the detection results. To make sure the objects on the edges are only detected once, overlapping bounding boxes are filtered based on their Intersection over Union (IoU) overlap rate.

4.2 Notation Assembly

4.2.1 Model Details

We use a 4-layer MLP for ϕ_{MLP} , where the two hidden layers both have hidden dimension 32. The embedding dimension for the symbol class is also set to be 32. We use ReLU [24] as the activation function.

4.2.2 Training

Again we used the MUSCIMA++ v2.0 dataset to train and evaluate the notation assembly model [4]. Following previous work [5, 6], we balance the positive and negative pairs in the training set by filtering out the pairs of nodes that are too distant from each other since they are unlikely to

be connected. Before feeding the bounding box coordinates to the model, we normalize them by the image width while keeping the aspect ratio fixed, so that all of the x -coordinate values fit in the range of $[-1, 1]$.

We train our models for 200 epochs with batch size 256, and use Adam optimizer with a learning rate of 0.0001. We evaluate our model every 20 epochs and pick the checkpoint with highest validation Match+AUC as our final model. All of the experiments are conducted with three different random seeds.

In our experiments, we consider three methods for training the notation assembly model:

- A **baseline**, which uses the ground-truth object lists provided in the MUSCIMA++ dataset to train the notation assembly model. This is the setup used in [5].
- A **pipeline**, which runs the music object detection model on the images to construct the training set for the notation assembly model, as discussed in Section 3.2.
- A “soft” variant of the pipeline, where we replace the embedding layer for the symbol class with a linear layer that maps the symbol class probabilities outputted from the music object detection model to a 32-dimensional vector. Note that this linear layer will have the same parameter count (number of classes multiplied by the hidden dimension) as the replaced embedding layer.

4.2.3 Inference

Since we consider both stages together, the input to the notation assembly stage should correspond to the output of the object detection stage. As described in Section 3.2, the detection output is converted into (V', E') . We then pass each pair of nodes to the notation assembly model, and feed the result into our evaluation function. We hypothesize that this realistic setup introduces a distribution shift to the model that was trained on the ground-truth objects and we will make the comparison in Section 5.

5. EXPERIMENTS

In this section, we first report the performance of our music symbol detection model. Then, we compare the performance of different notation assembly training pipelines using the evaluation metric described in Section 3.3.

5.1 Music Symbol Detection

Following the evaluation protocols of the Pascal VOC challenge [25], which is used by previous methods [8, 10, 11], we present both the mean average precision (mAP) and the weighted mean average precision, as detailed in Table 1. To elaborate, a predicted bounding box $\tilde{\mathbf{b}}_i$ is thought to be a true positive only if $\text{IoU}(\tilde{\mathbf{b}}_i, \mathbf{b}_j) > 0.5$ for some ground truth box \mathbf{b}_j . Then, average precision (AP) computes the area under the precision-recall curve, providing a single value that encapsulates the model's precision and recall performance. The weighted/unweighted mean Average Precision (mAP) extends the concept of AP by calculating the average AP values across multiple object classes,

Models	# Classes		mAP (%)	Weighted mAP (%)
YOLOv8 + cropping (ours)	163	(all)	84.79	92.67
YOLOv8 + cropping (ours)	73	(essential)	85.67	89.96
YOLOv8 + cropping (ours)	20		94.22	95.72
YOLOv4 + CBAM [8]	20		91.8	94.56 [†]
PP-YOLO-V2 [8]	20		91.1	–
YOLO-X [8]	20		90.4	–
YOLOv4 [8]	20		89.1	–
Faster R-CNN [8]	20		86.2	–

Table 1. Object detection results on test set. “mAP” is mean average precision. We compared it with results reported by [8]. The lower block is included for comparability with the 20-class setting from past work. †: Value computed from average precision per class reported in [8].

Models	# Classes	Match+AUC	
		Average	S.D.
MLP baseline (train on ground truth objects)	73	92.44 ± 0.24	
+ pipelined training (ours)	73	93.09 ± 0.16	
+ pipelined training + soft label (ours)	73	95.00 ± 0.18	
MLP baseline (train on ground truth objects)	163	83.97 ± 3.04	
+ pipelined training (ours)	163	85.76 ± 0.42	
+ pipelined training + soft label (ours)	163	87.10 ± 1.19	

Table 2. Multi-stage system results (test set) using our Match+AUC metric.

taking into account the number of occurrences of each class in a weighted or unweighted manner. Our experiments are conducted with the MUSCIMA++ v2.0 dataset, while the authors of most previous methods [10, 11] have only tested their models on MUSCIMA++ v1.0. This introduces a misalignment between our results. Thanks to Zhang et al. [8], who provided reproduced results of most previous methods on MUSCIMA++ v2.0, we directly report their reproduced results in the table.

Our model outperforms Zhang et al.’s method on their selected 20 classes by 2.4% (mAP, absolute), likely due to the improvements in YOLOv8 compared to v4.

5.2 Notation Assembly

In this section, we complete the multi-stage OMR system by chaining different notation assembly models to the best music object detection model we trained in Section 5.1. We use the metric we designed in Section 3.3 to report the end-to-end performance of the OMR system.

In Table 2, we compare the notation assembly systems trained with baseline training, pipelined training, and soft pipelined training as described in Section 4.2. We found that pipelined training improves the Match+AUC score by 0.65% (essential) and 1.79% (all), absolute, and incorporating the soft class label further increases the performance by 1.91% (essential) and 1.34% (all), absolute. Training the notation assembly model on the detection model output and using the soft label probability to represent the class information, we are able to improve the Match+AUC of the OMR system by 3.13%. We hypothesize that pipelined training helps the assembly model adapt to any inaccuracies our object detector has, and incorporating the soft class labels enables the assembly model to consider alternative class labels, not just those chosen by the object de-

tector.

6. CONCLUSION AND FUTURE WORK

In our study, we reconsider a multi-stage OMR pipeline built and evaluated using the MUSCIMA++ dataset. We first propose a state-of-the-art music symbol detector, serving as a strong preprocessor for the notation assembly stage. We then propose a training pipeline in which notation assembly is learned from imperfect object detection outputs (rather than ground-truth objects), which leads to higher performance. Finally, we introduce an evaluation score, Match+AUC, which can jointly consider the error in both detection and assembly stages, allowing evaluation of the two stages together.

Match+AUC is not restricted to being an evaluation metric. Future research could explore the application of Match+AUC within a joint training objective function for both the object detection and notation assembly stages. This approach would enable the entire model to be optimized for retrieving a globally optimal music notation graph.

In this study, we focused on the object detection and notation assembly stages in the OMR pipeline. Progress on the encoding stage is also required for a complete OMR solution; while the music notation graph arguably contains the essential information for recovering a score [4], conversion of such graphs into standard formats remains unsolved.

7. ACKNOWLEDGMENTS

The authors wish to express our deepest gratitude to all creators of the public OMR datasets for their dedication

and generosity in collecting and sharing these invaluable resources. We extend our sincere thanks to Carlos Peñarubia for his assistance in clarifying questions regarding the reproduction of their method. We are also grateful to Tim Althoff for his insightful comments on our evaluation metric. Special thanks go to Victoria Ebert and Teerapat Jenrungrot for providing us with essential materials in the OMR field. Finally, we sincerely appreciate the constructive reviews, which have significantly enhanced the rigor and completeness of this paper.

8. REFERENCES

- [1] D. Bainbridge and T. Bell, “The challenge of optical music recognition,” *Computers and the Humanities*, vol. 35, pp. 95–121, 05 2001. [Online]. Available: <https://doi.org/10.1023/A:1002485918032>
- [2] A. Rebelo, I. Fujinaga, F. Paszkiewicz, A. R. S. Marcal, C. Guedes, and J. S. Cardoso, “Optical music recognition: state-of-the-art and open issues,” *International Journal of Multimedia Information Retrieval*, vol. 1, no. 3, pp. 173–190, Oct. 2012. [Online]. Available: <https://doi.org/10.1007/s13735-012-0004-6>
- [3] J. Calvo-Zaragoza, J. Hajič, Jr., and A. Pacha, “Understanding optical music recognition,” *ACM Comput. Surv.*, vol. 53, no. 4, jul 2020. [Online]. Available: <https://doi.org/10.1145/3397499>
- [4] J. Hajič and P. Pecina, “The MUSCIMA++ dataset for handwritten optical music recognition,” in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 01, 2017, pp. 39–46. [Online]. Available: <https://doi.org/10.1109/ICDAR.2017.16>
- [5] C. Peñarubia, C. Garrido-Munoz, J. J. Valero-Mas, and J. Calvo-Zaragoza, “Efficient notation assembly in optical music recognition,” in *Proceedings of the 24th International Society for Music Information Retrieval Conference. ISMIR*, Dec. 2023, pp. 182–189. [Online]. Available: <https://doi.org/10.5281/zenodo.10265253>
- [6] A. Pacha, J. Calvo-Zaragoza, and J. Hajič, Jr., “Learning notation graph construction for full- pipeline optical music recognition,” in *Proceedings of the 20th International Society for Music Information Retrieval Conference. ISMIR*, Nov. 2019, pp. 75–82. [Online]. Available: <https://doi.org/10.5281/zenodo.3527744>
- [7] G. Jocher, A. Chaurasia, and J. Qiu, “Ultralytics YOLOv8,” 2023. [Online]. Available: <https://github.com/ultralytics/ultralytics>
- [8] Y. Zhang, Z. Huang, Y. Zhang, and K. Ren, “A detector for page-level handwritten music object recognition based on deep learning,” *Neural Computing and Applications*, vol. 35, no. 13, pp. 9773–9787, May 2023. [Online]. Available: <https://doi.org/10.1007/s00521-023-08216-6>
- [9] A. Fornés, A. Dutta, A. Gordo, and J. Lladós, “CVC-MUSCIMA: a ground truth of handwritten music score images for writer identification and staff removal,” *International Journal on Document Analysis and Recognition (IJ DAR)*, vol. 15, no. 3, pp. 243–251, Sep. 2012. [Online]. Available: <https://doi.org/10.1007/s10032-011-0168-2>
- [10] A. Pacha, K.-Y. Choi, B. Coüasnon, Y. Ricquebourg, R. Zanibbi, and H. Eidenberger, “Handwritten music object detection: Open issues and baseline results,” in *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*, 2018, pp. 163–168. [Online]. Available: <https://doi.org/10.1109/DAS.2018.51>
- [11] A. Pacha, J. Hajič, Jr., and J. Calvo-Zaragoza, “A baseline for general music object detection with deep learning,” *Applied Sciences*, vol. 8, 2018. [Online]. Available: <https://doi.org/10.3390/app8091488>
- [12] A. Konwer, A. K. Bhunia, A. Bhowmick, A. K. Bhunia, P. Banerjee, P. P. Roy, and U. Pal, “Staff line removal using generative adversarial networks,” in *2018 24th International Conference on Pattern Recognition (ICPR)*, 2018, pp. 1103–1108. [Online]. Available: <https://doi.org/10.1109/ICPR.2018.8546105>
- [13] J. Calvo-Zaragoza, A. Pertusa, and J. Oncina, “Staff-line detection and removal using a convolutional neural network,” *Machine Vision and Applications*, vol. 28, no. 5, pp. 665–674, Aug. 2017. [Online]. Available: <https://doi.org/10.1007/s00138-017-0844-4>
- [14] A.-J. Gallego and J. Calvo-Zaragoza, “Staff-line removal with selectional auto-encoders,” *Expert Systems with Applications*, vol. 89, pp. 138–148, 2017. [Online]. Available: <https://doi.org/10.1016/j.eswa.2017.07.002>
- [15] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788. [Online]. Available: <https://doi.org/10.1109/CVPR.2016.91>
- [16] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “YOLOv4: Optimal speed and accuracy of object detection,” 2020. [Online]. Available: <https://doi.org/10.48550/arXiv.2004.10934>
- [17] A. Baró, P. Riba, J. Calvo-Zaragoza, and A. Fornés, “From optical music recognition to handwritten music recognition: A baseline,” *Pattern Recognition Letters*, vol. 123, pp. 1–8, 2019. [Online]. Available: <https://doi.org/10.1016/j.patrec.2019.02.029>
- [18] F. Foscarin, F. Jacquemard, and R. Fournier-S’niehotta, “A diff procedure for music score files,” in *Proceedings of the 6th International Conference on Digital Libraries for Musicology*, ser. DLFM ’19, 2019, p.

- 58–64. [Online]. Available: <https://doi.org/10.1145/3358664.3358671>
- [19] J. Hajič, Jr., “A case for intrinsic evaluation of optical music recognition,” *International Workshop on Reading Music Systems*, 2018.
- [20] D. Byrd and J. Simonsen, “Towards a standard testbed for optical music recognition: Definitions, metrics, and page images,” *Journal of New Music Research*, vol. 44, 07 2015. [Online]. Available: <https://doi.org/10.1080/09298215.2015.1045424>
- [21] P. Torras, S. Biswas, and A. Fornés, “The common optical music recognition evaluation framework,” *arXiv preprint arXiv:2312.12908*, 2023.
- [22] D. F. Crouse, “On implementing 2D rectangular assignment algorithms,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 52, no. 4, pp. 1679–1696, 2016. [Online]. Available: <https://doi.org/10.1109/TAES.2016.140952>
- [23] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft COCO: Common objects in context,” in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 740–755. [Online]. Available: https://doi.org/10.1007/978-3-319-10602-1_48
- [24] A. F. Agarap, “Deep learning using rectified linear units (ReLU),” 2019. [Online]. Available: <https://doi.org/10.48550/arXiv.1803.08375>
- [25] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes challenge: A retrospective,” *International Journal of Computer Vision*, vol. 111, no. 1, pp. 98–136, Jan. 2015. [Online]. Available: <https://doi.org/10.1007/s11263-014-0733-5>

AUGMENT, DROP & SWAP: IMPROVING DIVERSITY IN LLM CAPTIONS FOR EFFICIENT MUSIC-TEXT REPRESENTATION LEARNING

Ilaria Manco

Queen Mary University of London

i.manco@qmul.ac.uk

Justin Salamon

Adobe Research

salamon@adobe.com

Oriol Nieto

Adobe Research

onieto@adobe.com

ABSTRACT

Audio-text contrastive models have become a powerful approach in music representation learning. Despite their empirical success, however, little is known about the influence of key design choices on the quality of music-text representations learnt through this framework. In this work, we expose these design choices within the constraints of limited data and computation budgets, and establish a more solid understanding of their impact grounded in empirical observations along three axes: the choice of base encoders, the level of curation in training data, and the use of text augmentation. We find that data curation is the single most important factor for music-text contrastive training in resource-constrained scenarios. Motivated by this insight, we introduce two novel techniques, Augmented View Dropout and TextSwap, which increase the diversity and descriptiveness of text inputs seen in training. Through our experiments we demonstrate that these are effective at boosting performance across different pre-training regimes, model architectures, and downstream data distributions, without incurring higher computational costs or requiring additional training data.

1. INTRODUCTION

Music-text embedding models have become a cornerstone of music information retrieval (MIR), facilitating core tasks that underpin music organisation and search, such as music tagging and cross-modal retrieval [8, 11, 14, 26, 27]. At a high level, these are multimodal models that produce aligned audio-text representations by learning to project high-dimensional data from the audio and text modalities onto a lower-dimensional joint representation space whose structure encodes semantic similarity. The canonical learning framework to obtain such embeddings is dual-encoder multimodal contrastive learning, first popularised by CLIP [20] in the image domain, and soon after adopted in most areas of machine perception, including audio [9, 29] and music processing [8, 11, 14].

Driven by the empirical success of this framework, a recent line of research has attempted to analyse its inner workings from a theoretical perspective [18, 36] or elucidate which aspects are most responsible for its effectiveness in visual models [34, 35]. However, within the audio domain, our understanding of multimodal contrastive learning remains limited [8], with sparse effort into ablating design choices, or training data- and compute-efficient models. Among prior work that takes a step in this direction, the focus is mostly on comparing backbone models [8, 11, 29], but without considering other important factors such as model initialisation or training data. Additionally, audio-text learning poses specific challenges in the context of music, as the amount of data with aligned audio and text is typically orders of magnitude smaller than in other domains, where large-scale web-crawled data is commonplace. This makes transferring insights from other areas of representation learning particularly challenging.

In this paper we present a deep dive into music-text contrastive learning and its use in text-based music retrieval, adopting a practical perspective and thoroughly investigating the impact of major design choices. In particular, we study the problem of how to train this family of models under different resource-constrained scenarios (with respect to data and compute), and how to meaningfully evaluate them for real-world use. In brief, our contributions are as follows: (i) we systematically compare backbone encoders in parameter-efficient settings, and demonstrate that we can leverage this to enable multilingual support for the first time and without additional training data (Section 3); (ii) we study the trade-off between training dataset size and quality, showing that the impact of data curation outweighs that of scale (Section 4); (iii) building upon these findings, we propose a training recipe, *Augment, Drop & Swap* to construct more effective contrastive views (via *Augmented View Dropout*) and improve model robustness (via *TextSwap*) with no extra computational overhead (Section 5). Incorporating the proposed pipeline within variants of the music-text contrastive framework under different computational constraints, we show that this consistently improves over prior work, establishing a new state-of-the-art on three benchmark datasets. Finally, we conduct the first listening study to evaluate text-based music retrieval, further corroborating our automatic evaluations and underscoring the importance of accounting for distribution gaps when measuring performance.



© I. Manco, O. Nieto, and J. Salamon. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

Attribution: I. Manco, O. Nieto, and J. Salamon, "Augment, Drop & Swap: Improving Diversity in LLM Captions for Efficient Music-Text Representation Learning", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.*, Milan, Italy, 2023.

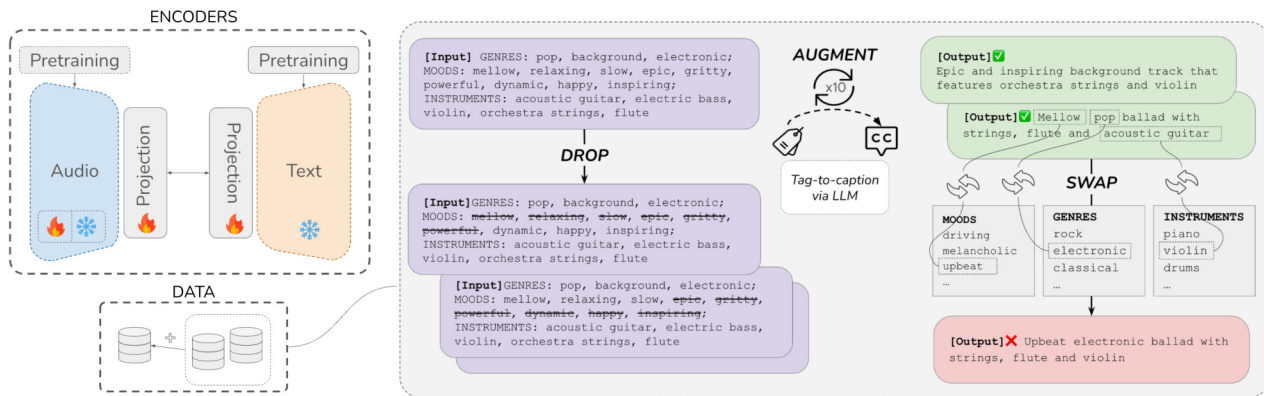


Figure 1: Overview of our approach. We study the role of encoders and data in music-text learning and propose a text augmentation pipeline, *Augment, Drop & Swap*, to increase data diversity and introduce hard negatives during training.

2. STUDYING THE DESIGN SPACE OF MUSIC-TEXT EMBEDDING MODELS

We explore two major factors in the design of music-text embedding models: architecture and data. While we acknowledge that there are others, such as training procedure, and alternative designs, we choose to restrict our focus exclusively to these two axes and to dual-encoder models, due to their predominance in the field. In the rest of the paper, we always refer to this family of models when discussing *music-text embeddings* or *music-text models*, and interchangeably use the terms *text* and *language*.

The typical design of a music-text embedding model consists of the following components: two modality-specific base encoders which separately process inputs of the text and audio modality to an intermediate representation space; a fusion or projection module responsible for mapping the intermediate representations to the shared embedding space; and a contrastive loss, through which the model parameters are optimised to encode semantically related audio and text inputs within the same neighbourhood of the embedding space, while pushing apart unrelated items. We provide an overview of this design in Figure 1. While prior works have converged towards standard choices for the last two components, it remains unclear how to reliably choose unimodal encoders among several existing options. We look at this in Section 3, before discussing the role of training data in Section 4.

2.1 Our experimental approach

Before delineating our areas of focus, we outline here the standard experimental setup used in our experiments.

Projection module We design our experiments to compare variations of the dual-encoder contrastive architecture described above, varying several components, but keeping two fixed throughout: the projection module and the loss. Similarly to [16, 24], we adopt a two-head, two-layer Transformer as our projection module. From a sequence of 256-dimensional embeddings produced by each projection head, we employ the [CLS] token embedding as the global representation for each branch. For ease of

reference, we denote this model architecture by DuET-MC (**D**ual-**E**ncoder **T**ext-**M**usic **C**ontrastive).

Training We optimise our network via the multimodal formulation of the InfoNCE loss [19], using cosine similarity between the l_2 -normalised projection embeddings from the audio and text branch as our scoring function, and a temperature parameter of 0.03. As part of our training procedure, we use the Adam optimizer with decoupled weight decay of 0.05, varying our learning rate through a cosine decay schedule from its peak value of $1e-3$, after a linear warm-up of 5 epochs. We train on 8 A100 NVIDIA GPUs, with an effective batch size of 1024 or 2048 based on memory requirements, for a maximum of 100 epochs, with early stopping based on the validation loss. Unless otherwise specified, our default training data is a corpus of licensed instrumental music with high-quality, manually curated genre, mood, and instrument tags, which we refer to as MusicTextHQ. For training, we select a subset totalling a duration of 100 hours, and augment tags into captions following our data augmentation strategy described in Section 4.

2.2 Evaluation

We evaluate all our models on text-based music retrieval, as this represents the most prominent task for music-text embedding models and has been shown to correlate to performance on other tasks [11, 14]. Retrieval is performed by ranking all audio clips in the dataset by decreasing cosine similarity of their embedding with the embedding of a text query. From this, we compute $\text{Recall}@k$ ($R@k$), the average number of times the target appears within the top- k retrieved items, and Median Rank (MR). To normalise performance scores by the different dataset sizes, we repeat this procedure on random subsets of 500 items, and report the average value for each metric. When reporting a single metric, we always refer to $R@10$.

Datasets In order to robustly measure performance across our experiments, we adopt a multi-dataset evaluation suite comprising three public datasets containing audio tracks paired with human-written captions: YT8M-MusicTextClips (MTC) [16], MusicCaps [2] and Song De-

Dataset	Hours*	Tags	Captions
<i>Training</i>			
LP-MusicCaps [7] (A)	50	Human	Synthetic
MusicTextHQ (B)	100	Human	Synthetic
YT8M-MV [1] (C)	270	Synthetic	Synthetic
<i>Evaluation</i>			
YT8M-MTC [16]	8	-	Human
MusicCaps [2]	8	-	Human
Song Describer [15]	2	-	Human

Table 1: Overview of the datasets used in our experiments. *Hours denotes the audio duration used in training.

Encoder	# Params	Model version
<i>Audio</i>		
HTS-AT [5]	30M	AudioSet ¹
MERT [33]	330M	MERT-v1-330M
<i>Text</i>		
RoBERTa [13]	125M	roberta-base
CLIP-T [20]	151M	clip-vit-base-patch32
T5 [21]	11.3B	flan-t5-xx
mT5 [30]	13B	mt5-xx

Table 2: Audio and text encoders we compare in our experiments on the impact of encoder backbones (Section 3).

scriber (SDD) [15]. These all represent out-of-distribution data (see Table 1), with different degrees and types of distribution shifts in both the audio and text modality. For example, MTC and MC both contain 10-second audio clips from YouTube videos, but they differ significantly in their captions, with respect to content, descriptiveness and even text length [15]. Audio in the SDD consists instead of music recordings from the music platform Jamendo [3], while captions describe much longer audio segments.

3. THE ROLE OF ENCODER BACKBONES

We experiment with two audio encoders, HTS-AT [5] and MERT [33], and three text encoders, RoBERTa [13], the text encoder from CLIP [20] (CLIP-T), T5 [21] and mT5 [30]. We choose these either because they represent the state of the art in their respective tasks, or because they have been previously used in contrastive audio-text learning, thus allowing for direct comparison with prior work.

3.1 Encoders: initialization and freezing

In this set of experiments, our goal is to study parameter-efficient configurations of existing audio and text encoders, training only a subset of the model weights. The motivation for exploring this setting is threefold: freezing part of the model lowers the memory budget and training time, it avoids catastrophic forgetting [17], and it reduces the risk of overfitting in data-constrained scenarios. To fulfil these requirements, we do not consider end-to-end finetuning, and instead focus on leveraging pre-training, locking the audio and text encoders based on their parameter size.

¹ We use the HTSAT_AudioSet_Saved_6 checkpoint of HTS-AT trained on AudioSet from the official repository.

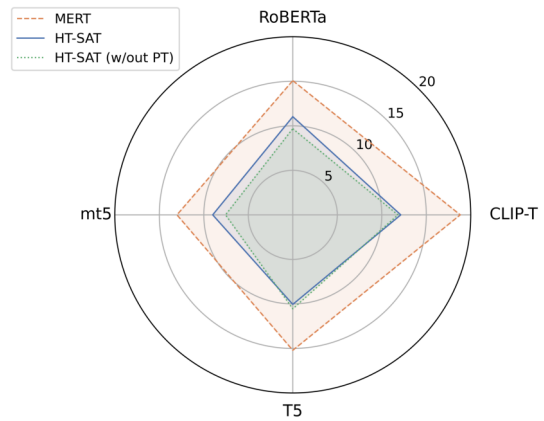


Figure 2: Retrieval performance (R@10) of different combinations of audio and text encoders compared through the lens of our DuET-MC framework.

Specifically, we keep all text encoders frozen, as these all count over 100M parameters, as shown in Table 2, and only train the full audio encoder, both with and without general-audio pre-training when using HTS-AS, due to its smaller size. When using MERT, we keep the encoder frozen but train a learnable aggregator over the hidden states of each layer, implemented as a 1D convolutional layer, to obtain audio representations that capture the different levels of abstraction encoded at different depths of the network [33].

Results From Figure 2 we first observe that, under the constraints described above, the overall best configuration is given by MERT and CLIP-T. We attribute this to two main reasons: with regards to the audio branch, the superior performance exhibited by MERT suggests that a larger model capacity and stronger music prior may be beneficial to music-text alignment; with regards to the text branch, while all encoders are characterised by large-scale pre-training, CLIP-T stands out as the only model with multimodal capabilities. Although this is somewhat surprising, as CLIP is pre-trained on image-text pairs, we note that prior work has also shown that it can be successfully transferred to the audio and music domains [6, 16, 28, 32]. Secondly, when using MERT with any of the text encoders considered, we find that we can train less than 1% of the total amount of weights (~ 3M making up the projection and aggregation layers) without loss of performance compared to current state-of-the-art models (shown later in Table 4). This demonstrates that we can successfully align locked text representations to the audio modality through lightweight music-text contrastive learning, confirming that our encoder locking strategy is effective when leveraging powerful music-specific pre-training, in line with similar findings in the visual domain [35]. With regards to the audio branch initialization, comparing the two variants of HTS-AT, we find that general-purpose audio pre-training can give a slight advantage over training from scratch, but this benefit is not consistent across the different text encoders HTS-AT is paired with. In the rest of the paper we fix the encoder configuration to locked MERT + locked CLIP-T in all experiments, unless otherwise specified.

Language	R@10		
	YT8M-MTC	MusicCaps	Song Describer
English	10.43	16.00	19.00
German	9.90	13.28	18.00
French	11.71	12.32	15.40
Italian	10.43	13.68	15.80
Spanish	11.60	13.48	18.40

Table 3: Multilingual retrieval performance.

3.2 Supporting retrieval in multiple languages

Due to a lack of data in different languages, music-text modelling has so far exclusively focussed on English. Real-world applications for music-text embeddings, however, can greatly benefit from the support of multiple languages. To address this limitation, we explore the use of pre-trained locked encoders, similarly to Section 3.1, this time adopting mT5 [30], a multilingual text-to-text Transformer model, as our text encoder. To evaluate multilingual performance, we choose a subset of four languages, German, French, Italian and English, and translate our evaluation datasets via GPT3.5-turbo [4]. In Table 3, we show that this approach provides a viable solution to text-based retrieval in multiple languages while using only English text paired with music in training and with only a minor drop in performance compared to English.

4. THE ROLE OF TRAINING DATA

Having established best practices with respect to choosing audio and text backbones, we now shift our attention to the training data. As widely acknowledged in the literature [7, 8, 14], a major limitation in training music-language models is the lack of large public datasets with paired audio-text data. To circumvent this issue, a number of works have proposed to employ large language models to augment text data more commonly found in music datasets, such as categorical labels, metadata and tags, into full natural language sentences, corresponding to *pseudo-captions* [7, 10, 16]. In the next section we present our investigation of the impact of tag-to-caption augmentation.

4.1 Tag-to-caption augmentation via LLMs

Following [16], we leverage the in-context learning ability of LLMs via few-shot prompting, and adopt a similar approach to augment tags into captions for our training dataset MusicTextHQ. For this, we use BLOOM-176B [23], a competitive open-access LLM trained on responsibly sourced data. Differently from [16], we do not employ synthetic tags, but use tags provided by expert annotators. We compare this to training on LP-MusicCaps-MTT [7] (LP-MusicCaps for short), a dataset obtained via a similar approach, where tags from the MagnaTagATune [12] dataset are augmented into captions via GPT3.5-turbo. To measure the impact of tag-to-caption augmentation, we train three variants of our model on each dataset, varying p_{cap} , the probability of selecting captions over tags as the text input for each training pair.

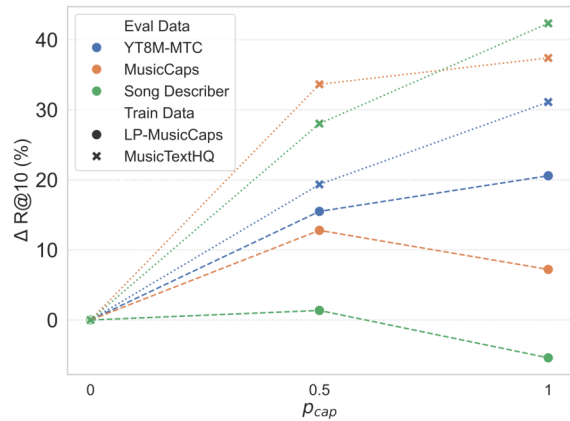


Figure 3: The effect of varying p_{cap} , the probability of swapping tags with captions. On the y-axis, we show the relative change in performance compared to $p_{cap} = 0$.

Results Results are shown in Figure 3, where we compare the effect of gradually shifting from tags to captions in the two training datasets considered. We first note that introducing tag-to-caption augmentation for at least a portion of the training data ($p_{cap} = 0.5$) leads to an improvement regardless of training dataset. Interestingly, unlike in MusicTextHQ, this trend does not extend to the scenario where we replace all text inputs with pseudo-captions ($p_{cap} = 1$) in LP-MusicCaps. In this case, we observe instead a slight degradation in performance on two of the evaluation datasets, compared to using only tags, or using captions half of the time. We posit that this divergence may be due to a gap in label quality between the two training sets, as pseudo-captions in LP-MusicCaps are generated based on sparse labels, with 50% of the items in the dataset paired to only three tags or less, and in many cases without being balanced across categories. Intuitively, this is likely to result in non-descript, or even inaccurate captions, as the LLM generation will be more prone to hallucinations and may therefore deviate substantially from the audio content. In contrast, MusicTextHQ provides strong grounding to the audio content, with multiple expert-provided tags per category (often three or more for *each* tag category). From this, we conclude that, while LLM-enabled text augmentation can provide a valuable strategy for enriching training data, it is not a substitute for adequate data curation, but rather a supplement. This is an important observation, as prior work has also found that specificity in captions is instrumental to effective multimodal contrastive learning [22, 31]. Since LLM-based augmentation, being bounded by the information content in the source data, cannot increase specificity, our results highlight an often overlooked shortcoming of synthetic text.

4.2 Training data: size vs quality

Next, we ask whether simply increasing dataset size can emphasise the benefits of tag-to-caption augmentation. To scale up the size of our training data, we include YT8M-MV, a subset of the YouTube8M dataset [1] tagged as *music video*, as an additional dataset to our training pool. For this, we follow [16] and employ tags from an automatic

music tagger and pseudo-captions generated following the same procedure described in Section 4.1. For simplicity, we refer to LP-MusicCaps, MusicTextHQ and YT8M-MV as Dataset_A (or simply A), Dataset_B (B) and Dataset_C (C), ordered by size as shown in Table 1. We also consider combining the two biggest datasets (B + C) and all three together (A + B + C). We note that each dataset differs not only in size, but also in audio and label quality.

Results In Figure 4 we showcase results from training on the datasets described above. Notably, we find that scaling dataset size does not consistently result in an improvement, signalling that the gap in quality between datasets can eclipse their size difference. Although we observe that combining all datasets yields better performance, likely due to overall increased diversity in the training data, the difference is not proportionate to the rise in training cost necessary to scale up. Instead, our results underscore the importance of data curation as a more efficient way of boosting performance, confirming that constructing a subset of highly curated examples, with descriptive and accurate captions, more positively contributes to learning in the contrastive setting [22].

5. IMPROVING DIVERSITY VIA TEXT AUGMENTATIONS

Having established that augmenting high-quality tags into captions offers a useful and inexpensive strategy to enrich training data, we explore this further and propose two augmentation-based techniques aimed at increasing data diversity and model robustness.

5.1 Augment, Drop & Swap

Augmented View Dropout First, building upon the tag-to-caption strategy described in Section 4.1, we explore text augmentation with the goal of constructing more effective views for contrastive learning, following the principle that optimal views should minimise mutual information between paired items while retaining a high degree of semantic alignment [25]. To this end, we propose *Augmented View Dropout*, where, for each item in our dataset, we randomly sample a subset of the tags, balanced by category (genre, mood, instrumentation) and produce a set of 10 different captions. Each can be thought of as a complementary, but partial view of the associated music track, as we mask a subset of all the ground-truth tags to produce each view. At training time, views are randomly sampled, effectively resulting in a further form of data augmentation.

Hard negatives via TextSwap Finally, we tackle another important challenge in contrastive learning, hard negative sampling, and propose to also address this through the lens of text augmentation, via a technique which we call TextSwap. In order to increase the rate of hard negatives beyond the natural rate found in the dataset, we create partially perturbed versions of the captions by stochastically swapping genre, mood or instrument keywords with alternative descriptors from a predefined dictionary (e.g. “a mellow *pop* track” becomes “a mellow *hip-hop* track”).

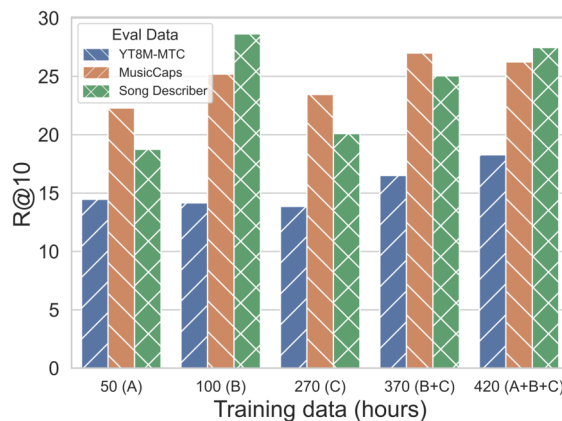


Figure 4: Retrieval performance across models trained on datasets that differ in size and annotation quality.

During training, for each positive pair, we then select a random subset of the negative captions in a batch and replace them with hard negatives by applying TextSwap once per descriptor category. This is illustrated in Figure 1, where we provide a visual guide for the full *Augment, Drop & Swap* pipeline. We hypothesise that the presence of hard negatives is particularly critical in later stages of training, once the model has already acquired basic features, and learning on “easy” negatives has saturated. Based on this, we follow a curriculum learning approach and linearly increase the probability of applying TextSwap from 0 to 15% over the course of 20 epochs, after a warm-up period of 5.

5.2 Experiments

Ablations In this set of experiments we examine the effect of each of the three components in our augmentation pipeline: tag-to-caption augmentation, Augmented View Dropout and TextSwap. We look at two scenarios: one where we want to measure their contribution in training two variants of our parameter-efficient DuET-MC framework, each with different degrees of audio pre-training and finetuning and locked text encoders, and one where we relax our computational requirements and explore whether our proposed method can be usefully applied in finetuning a general purpose audio-text embedding model (CLAP [29]), with limited paired music data.

Results We present our ablations on the proposed pipeline in Table 4, where we also compare to two audio-text contrastive baselines, CLAP [29] and TTMR [8], trained on general-purpose audio and music respectively. The table displays three different settings to which we apply our proposed pipeline: (1) training the audio encoder from scratch (shown in the HTS-AT + CLIP-T configuration), (2) training only 1% of the parameters in our locked audio-text encoder (MERT + CLIP-T), and (3) fine-tuning the full model on music, following general audio-text pre-training (CLAP-FT). From this, we observe that, while the vanilla version of DuET-MC (trained only on tags) exhibits at best comparable performance to the baselines, each additional component in our pipeline lifts performance across all model configurations, pre-training regimes and finetuning strategies. Among these, tag-to-caption augmentation

Model	Tag-to-caption	Augmented View Dropout	TextSwap	YT8M-MTC		MusicCaps		Song Describer		Avg R@10 ↑
				R@10 ↑	MR ↓	R@10 ↑	MR ↓	R@10 ↑	MR ↓	
<i>Baselines</i>										
CLAP [29]	-	-	-	11.9	80	40.3*	17*	19.8	53	24.0*
TTMR [8]	-	-	-	11.6	79	9.6	115	16.5	57	12.6
DuET-MC	✗	✗	✗	8.5	103	12.2	82	15.3	53	12.0
(HTS-AT + CLIP-T)	✓	✗	✗	8.0	104	13.4	76	14.1	57	11.8
	✓	✓	✗	<u>9.4</u>	<u>93</u>	15.1	<u>65</u>	<u>19.6</u>	49	<u>14.7</u>
	✓	✓	✓	<u>9.4</u>	<u>93</u>	<u>15.8</u>	66	17.4	<u>48</u>	14.2
DuET-MC	✗	✗	✗	10.8	82	18.3	56	20.2	45	16.4
(MERT + CLIP-T)	✓	✗	✗	11.7	69	21.3	41	23.4	36	18.8
	✓	✓	✗	13.4	65	<u>24.9</u>	36	27.7	32	22.0
	✓	✓	✓	<u>14.5</u>	<u>62</u>	24.6	<u>34</u>	<u>27.3</u>	29	<u>22.1</u>
CLAP-FT	✗	✗	✗	14.2	63	38.8*	18*	20.8	38	24.6*
	✓	✗	✗	14.6	61	42.3*	15*	23.5	34	26.8*
	✓	✓	✗	16.3	55	41.6*	16*	24.5	36	27.3*
	✓	✓	✓	15.7	57	43.5*	14*	26.3	<u>31</u>	28.5*

Table 4: Ablations. For each model, subsequent rows show the effect of introducing an additional step in our proposed *Augment, Drop & Swap* pipeline. We highlight best results for each model (underlined) and amongst all models (bold). * denotes values that may be inflated due to in-distribution bias.

and Augmented View Dropout emerge as the most influential, while the benefits of TextSwap are more prominent for model configurations where encoders have higher levels of pre-training, hinting at the necessity to increase the complexity of negatives later in training. This suggests that our *Augment, Drop & Swap* recipe provides a data-efficient strategy to improve music-text modelling under a variety of model configurations, at no additional computational cost. Importantly, this trend generalises across evaluation datasets, suggesting that it is beneficial to model robustness, and demonstrates that the lack of large-scale paired data in the music domain can be alleviated through augmentation-based techniques which enhance data quality instead of quantity. Finally, comparing retrieval scores of different family of models (TTMR, CLAP and DuET-MC), we note consistent differences between datasets, with CLAP-based models invariably showing a significant jump in performance on the MusicCaps dataset compared to MTC and SDD. We hypothesise that this may be a result of in-distribution bias, since there are several instances of non-music or noisy, low-quality recordings in MC. Since CLAP is trained to recognise everyday sounds, this points at a smaller shift from its training distribution, compared to SDD and MTC, which are exclusively composed of music recordings. We posit that further mismatches in the training and test distributions exist along the text dimension and investigate this through human evaluation.

Are metrics aligned with human preference? We recruit 35 participants to evaluate DuET-MC, CLAP and TTMR in a head-to-head pairwise comparison. Participants are presented with up to 24 text prompts, where each is a caption taken from one of the three evaluation datasets, and are asked to choose which one of two music tracks best aligns to the description. Through this qualitative evaluation, we find that DuET-MC does substantially better than TTMR, losing against it only 30.9% of the times, and largely mirroring our findings from Section

5.2. Surprisingly, the win and tie rate vs CLAP drops instead to 37.3% and 38.5% respectively. Looking at the breakdown of scores by dataset, this advantage in CLAP is predominantly observed on MC and MTC, while DuET-MC outperforms CLAP on SDD. Interestingly, DuET-MC is preferred or considered equivalent to the ground truth 38.9% of the times on MTC compared to 15.4 and 17.9% on the other two datasets. This points to significant differences in the level of alignment between caption and audio in the different datasets, signalling that evaluating on several datasets is paramount to understanding real-world performance. Additionally, it leads to an observation that complements our automatic evaluation in Table 4: the discrepancy between DuET-MC’s performance on MTC compared to MC and SDD may be ascribed to a higher degree of *vagueness* in MTC captions, which, as revealed through our qualitative evaluation, admit instead alternative matching tracks to those in the ground truth.

6. CONCLUSIONS

In this work we presented *Augment, Drop & Swap*, a training recipe for efficient music-text representation learning informed by our findings on training music-text contrastive models in resource-constrained scenarios. Through our experiments, we provide a practical guide to this family of models, and foreground their real-world use by focusing on multilingual support, computationally efficient techniques, and cross-dataset evaluation. Showing that data curation has a significant effect at modest data scales, we design each step in our pipeline to tackle specific aspects of the text used in training, such as descriptiveness and specificity, via data augmentations, leading to views that are more effective in multimodal contrastive learning. Through automatic and qualitative evaluations, we show the usefulness of our approach and reveal insights on the relation between measured performance and distribution shifts in the test data.

7. REFERENCES

- [1] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. “Youtube-8m: A large-scale video classification benchmark”. In: *arXiv preprint arXiv:1609.08675* (2016).
- [2] Andrea Agostinelli et al. “Musiclm: Generating music from text”. In: *arXiv preprint arXiv:2301.11325* (2023).
- [3] Dmitry Bogdanov, Minz Won, Philip Tovstogan, Alastair Porter, and Xavier Serra. “The MTG-Jamendo Dataset for Automatic Music Tagging”. In: *Machine Learning for Music Discovery Workshop, International Conference on Machine Learning (ICML)*. 2019.
- [4] Tom B Brown et al. “Language Models are Few-Shot Learners”. In: *Advances in Neural Information Processing Systems*. 2020.
- [5] Ke Chen, Xingjian Du, Bilei Zhu, Zejun Ma, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. “HTS-AT: A Hierarchical Token-Semantic Audio Transformer for Sound Classification and Detection”. In: *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022.
- [6] Tianyu Chen, Yuan Xie, Shuai Zhang, Shaohan Huang, Haoyi Zhou, and Jianxin Li. “Learning Music Sequence Representation From Text Supervision”. In: *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2022, pp. 4583–4587.
- [7] SeungHeon Doh, Keunwoo Choi, Jongpil Lee, and Juhan Nam. “LP-MusicCaps: LLM-Based Pseudo Music Captioning”. In: *Proceedings of the 24th International Society for Music Information Retrieval (ISMIR) Conference*. Milan, 2023.
- [8] SeungHeon Doh, Minz Won, Keunwoo Choi, and Juhan Nam. “Toward Universal Text-to-Music Retrieval”. In: *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022.
- [9] Benjamin Elizalde, Soham Deshmukh, Mahmoud Al Ismail, and Huaming Wang. “CLAP Learning Audio Concepts from Natural Language Supervision”. In: *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2023, pp. 1–5.
- [10] Josh Gardner, Simon Durand, Daniel Stoller, and Rachel Bittner. “LLark: A Multimodal Instruction-Following Language Model for Music”. In: *Proceedings of the 41st International Conference on Machine Learning (ICML)*. 2024.
- [11] Qingqing Huang, Aren Jansen, Joonseok Lee, Ravi Ganti, Judith Yue Li, and Daniel P. W. Ellis. “MuLan: A Joint Embedding of Music Audio and Natural Language”. In: *23rd International Society for Music Information Retrieval Conference (ISMIR 2022)*. 2022.
- [12] Edith Law, Kris West, Michael Mandel, Mert Bay, and J. Stephen Downie. “Evaluation of algorithms using games: The case of music tagging”. In: *Proceedings of the 10th ISMIR Conference*. 2009.
- [13] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. “Roberta: A robustly optimized bert pretraining approach”. In: *arXiv preprint arXiv:1907.11692* (2019).
- [14] Iliaria Manco, Emmanouil Benetos, Elio Quinton, and Gyorgy Fazekas. “Contrastive Audio-Language Learning for Music”. In: *23rd International Society for Music Information Retrieval Conference (ISMIR 2022)*. 2022.
- [15] Iliaria Manco et al. “The Song Descriptor Dataset: a Corpus of Audio Captions for Music-and-Language Evaluation”. In: *Machine Learning for Audio Workshop at NeurIPS 2023*. 2023.
- [16] Daniel McKee, Justin Salamon, Josef Sivic, and Bryan Russell. “Language-guided music recommendation for video via prompt analogies”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023, pp. 14784–14793.
- [17] Sanket Vaibhav Mehta, Darshan Patil, Sarath Chandar, and Emma Strubell. “An Empirical Investigation of the Role of Pre-training in Lifelong Learning”. In: *Journal of Machine Learning Research* 24.214 (2023), pp. 1–50.
- [18] Ryumei Nakada, Halil Ibrahim Gulluk, Zhun Deng, Wenlong Ji, James Zou, and Linjun Zhang. “Understanding Multimodal Contrastive Learning and Incorporating Unpaired Data”. en. In: *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*. PMLR, 2023, pp. 4348–4380.
- [19] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. “Representation learning with contrastive predictive coding”. In: *arXiv preprint arXiv:1807.03748* (2018).
- [20] Alec Radford et al. “Learning Transferable Visual Models From Natural Language Supervision”. In: *International Conference on Machine Learning*. PMLR, 2021.

- [21] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. “Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer”. In: *Journal of Machine Learning Research* 21.140 (2020), pp. 1–67.
- [22] Shibani Santurkar, Yann Dubois, Rohan Taori, Percy Liang, and Tatsunori Hashimoto. “Is a Caption Worth a Thousand Images? A Study on Representation Learning”. In: *The Eleventh International Conference on Learning Representations*. 2023.
- [23] Teven Le Scao, Angela Fan, and Christopher Akiki. *BLOOM: A 176B-Parameter Open-Access Multilingual Language Model*. 2023.
- [24] Didac Suris, Carl Vondrick, Bryan Russell, and Justin Salamon. “It’s Time for Artistic Correspondence in Music and Video”. en. In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2022, pp. 10554–10564.
- [25] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. “What Makes for Good Views for Contrastive Learning?” In: *Advances in Neural Information Processing Systems*. Vol. 33. 2020, pp. 6827–6839.
- [26] Minz Won, Keunwoo Choi, and Xavier Serra. “Semi-Supervised Music Tagging Transformer”. In: *International Society for Music Information Retrieval Conference (ISMIR)*. 2021.
- [27] Minz Won, Justin Salamon, Nicholas J. Bryan, Gautham J. Mysore, and Xavier Serra. “Emotion Embedding Spaces for Matching Music to Stories”. In: *International Society for Music Information Retrieval Conference (ISMIR)*. 2021.
- [28] Ho-Hsiang Wu, Prem Seetharaman, Kundan Kumar, and Juan Pablo Bello. “Wav2CLIP: Learning Robust Audio Representations from Clip”. In: *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2022, pp. 4563–4567.
- [29] Yusong Wu, Ke Chen, Tianyu Zhang, Yuchen Hui, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. “Large-Scale Contrastive Language-Audio Pretraining with Feature Fusion and Keyword-to-Caption Augmentation”. In: *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2023, pp. 1–5.
- [30] Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. “mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer”. In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, 2021, pp. 483–498.
- [31] Yihao Xue, Siddharth Joshi, Dang Nguyen, and Baharan Mirzasoleiman. “Understanding the Robustness of Multi-modal Contrastive Learning to Distribution Shift”. In: *The Twelfth International Conference on Learning Representations*. 2024.
- [32] Dongchao Yang, Jianwei Yu, Helin Wang, Wen Wang, Chao Weng, Yuexian Zou, and Dong Yu. “DiffSound: Discrete Diffusion Model for Text-to-Sound Generation”. In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 31 (2023), pp. 1720–1733.
- [33] Li Yizhi, Ruibin Yuan, Ge Zhang, Yinghao Ma, Xingran Chen, Hanzhi Yin, Chenghao Xiao, Chenghua Lin, Anton Ragni, Emmanouil Benetos, et al. “MERT: Acoustic Music Understanding Model with Large-Scale Self-supervised Training”. In: *The Twelfth International Conference on Learning Representations*. 2023.
- [34] Haoxuan You, Luowei Zhou, Bin Xiao, Noel Codella, Yu Cheng, Ruochen Xu, Shih-Fu Chang, and Lu Yuan. “Learning visual representation from modality-shared contrastive language-image pre-training”. In: *European Conference on Computer Vision*. Springer, 2022, pp. 69–87.
- [35] Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. “Lit: Zero-shot transfer with locked-image text tuning”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 18123–18133.
- [36] Qi Zhang, Yifei Wang, and Yisen Wang. “On the generalization of multi-modal contrastive learning”. In: *International Conference on Machine Learning*. PMLR. 2023, pp. 41677–41693.

MUSIC DISCOVERY DIALOGUE GENERATION USING HUMAN INTENT ANALYSIS AND LARGE LANGUAGE MODELS

SeungHeon Doh^b Keunwoo Choi[‡] Daeyong Kwon^b
 Taesu Kim[‡] Juhan Nam^b

^b Graduate School of Culture Technology, KAIST, South Korea

[‡] Prescient Design, Genentech, New York, NY, USA

[‡] Department of Industrial Design, KAIST, South Korea

{seungheondoh, ejmj63, tskind77, juhan.nam}@kaist.ac.kr, choi.keunwoo@gene.com

ABSTRACT

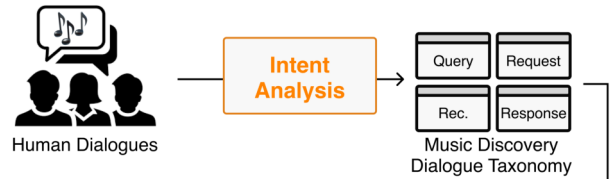
A conversational music retrieval system can help users discover music that matches their preferences through dialogue. To achieve this, a conversational music retrieval system should seamlessly engage in multi-turn conversation by 1) understanding user queries and 2) responding with natural language and retrieved music. A straightforward solution would be a data-driven approach utilizing such conversation logs. However, few datasets are available for the research and are limited in terms of volume and quality. In this paper, we present a data generation framework for rich music discovery dialogue using a large language model (LLM) and user intents, system actions, and musical attributes. This is done by i) dialogue intent analysis using grounded theory, ii) generating attribute sequences via cascading database filtering, and iii) generating utterances using large language models. By applying this framework to the Million Song dataset, we create – **LP-MusicDialog**, a Large Language Model based Pseudo Music Dialogue dataset, containing over 288k music conversations using more than 319k music items. Our evaluation shows that the synthetic dataset is competitive with an existing, small human dialogue dataset in terms of dialogue consistency, item relevance, and naturalness. Furthermore, using the dataset, we train a conversational music retrieval model and show promising results.¹

1. INTRODUCTION

In recent years, conversational systems have emerged as a promising solution to enhance user experience in various domains [1–4], including conversational music retrieval and recommendation [5, 6]. The goal of a conversational

¹ Our code is available at <https://github.com/seungheondoh/lp-music-dialog/>

Step1. Intent Analysis from Human Dialogue



Step2. Cascading Music Database Filtering



Step3. Music Discovery Dialogue Generation

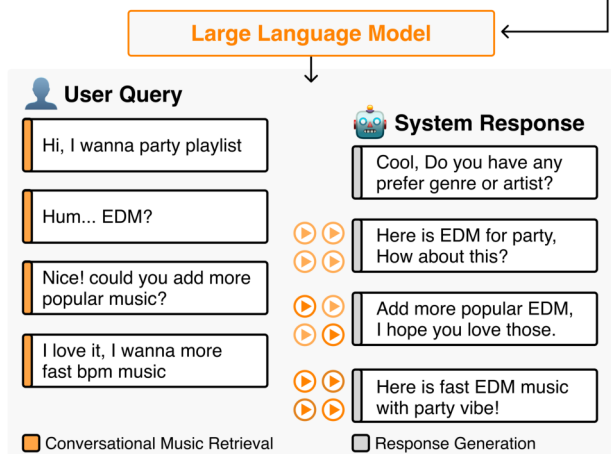


Figure 1. The generation process of pseudo musical dialogue.

music system is to assist users in finding their desired music through dialogues. Such a system should possess three key capabilities: i) to understand the *intents* and *musical needs* of users from their queries expressed in natural language, ii) to generate responses and facilitate human-like interaction, iii) to find music that aligns with the user’s preferences by taking previous dialogues into account.

Currently, the primary challenge of developing a conversational music retrieval system is the scarcity of large-scale public datasets. Chaganty et al. [5] introduce the Conversational Playlist Curation Dataset (CPCD). This crowd-sourced dataset comprises human-to-human dialogues that simulate the process of music discovery. How-

ever, as it relies on a manual process, the dataset is small and exhibits biases from the music streaming platforms used by the recommenders. To address this problem, Leszczynsk et al. [6] propose a dialogue generation framework through random walks in the music-text joint embedding space and dialogue inpainting [7]. However, this approach requires a high-quality music-text joint embedding and needs to use template-based system responses as input. As a result, the system’s responses are always composed of limited format utterances, leading to low naturalness in human evaluation.

In this paper, we introduce a framework for generating *human-like* music discovery dialogues using intent and a large language model (LLM). The proposed framework is based on the existing method [6], but we address their limitations by employing *cascading music database filtering* instead of a joint embedding and extensive *intent analysis* for naturalness. Using the grounded theory approach [8], we analyze a dataset of human music discovery dialogues and develop taxonomies for user intents, system actions, and musical attributes relevant to the task of music discovery. Furthermore, we introduce a model-free attribute sequence generation by applying cascading filtering to a multi-label music annotation database. Finally, we synthesize music discovery dialogues through an LLM using the created attribute sequences and human intents/actions.

Our contributions are threefold: First, we analyze music discovery dialogues and propose a taxonomy. Second, we introduce the LP-MusicDialog dataset, a large-scale synthetic dialogue dataset created using human intent and an LLM. Third, we present extensive objective and subjective evaluations to demonstrate the effectiveness of LLM-based pseudo-music dialogues.

2. RELATED WORK

Recently, there has been some progress in conversational music systems including language-based music retrieval [9–11]. However, existing systems are often limited to single-turn tasks. This means users are not able to refine their queries to obtain a highly satisfactory outcome. Towards multi-turn dialogues, Chaganty et al. [5] released the Conversational Playlist Curation Dataset (CPCD), which comprises 917 dialogues averaging 5.7 turns each. Unlike single-turn retrieval, conversational retrieval takes into account previous chat history to find relevant items. The model in [5] aggregates the context of history embedding and the current query embedding using average pooling, then uses contrastive loss to maximize the similarity between them. However, their model showed limited performance due to the small size of CPCD.

A solution to the data scale issue is synthesizing data using existing datasets and language models. Recently, synthetic datasets that bridge natural languages and music have been proposed to enhance music understanding [12, 13], captioning [13, 14], reasoning [13], and retrieval [11]. For the conversational music retrieval, Leszczynski et al. [6] proposed a two-stage data synthesis framework: musical attribute sequence generation via

random walk and utterance generation through dialogue inpainting [7]. The musical attribute sequence represents the evolution of user queries over turns (e.g., ask for workout music in the first turn, then refine the results to be also pop music). During utterance generation, a language model creates user queries using system responses as input, which include sampled musical attributes and a static template.² As a result, they created one million multi-turn music discovery dialogues: TtW Music, leveraging a private playlist dataset. However, this approach encounters issues with model errors in the music-text joint embedding space used for the random walk method and faces challenges with response consistency due to the reliance on manually created templates for system responses.

Deep understanding in music query has to be preceded aforementioned data generation. So far, query understanding has primarily focused on describing musical needs. Downie and Cunningham [15] analyzed 161 music queries and categorized them into 1) information needs, 2) desired outcomes, 3) intended uses for the information, and 4) social and contextual elements. Bainbridge et al. [16] utilized the grounded theory approach to analyze 502 real-world music queries, expanding upon prior research with 10 types of need descriptions. Lee [17] analyzed 1,705 Google Answers queries to propose a refined taxonomy for information needs and searching behavior. Despite these efforts, previous studies have been limited to the information needs (musical attributes) contained in queries, and intent in multi-turn queries has not received significant attention.

3. DIALOGUE INTENT ANALYSIS

3.1 Taxonomy Development

For the dialogue-specific music discovery taxonomy, we analyze the existing human-to-human music dialogue dataset (CPCD [5]) using the grounded theory approach [8]. Grounded theory is a qualitative approach that creates refined theory from unstructured real-world data. In detail, (1) we adopt the taxonomy from previous research as our initial taxonomy. For user intent and system action, we use the *conversational movie intent* taxonomy [18], and for musical needs, we use *music feature* taxonomy [17] as the initial taxonomy. (2) Three authors annotate ten dialogues using the initial taxonomy and discuss the limitations of the existing taxonomy. (3) We update the taxonomy and annotate a new randomly sampled five dialogues. This cycle of proposing, refining, and annotating was completed three times to ensure our taxonomy could capture the full range of scenarios present in the dialogue samples.

3.2 Analysis Results

3.2.1 Taxonomy for User Intents

In Table 1, we categorize user intents into four main categories, with eight detailed sub-intents. The **Initial Query**

² For example, “Of course! Let me add some songs described as <musical attributes>. What else?”

User Intent	Description	Example	%
<i>Start Dialogue</i>			
Initial Query	User initiates the inquiry with a specific request.	"Hi, I want to create a playlist for hiking."	18.5
Greeting	User initiates the dialog, often with greeting words.	"Hello, I would like... / "Good Morning! Let's start..."	12.7
<i>Item Discovery Query (Retrieval / Recommendation)</i>			
Positive Filter	User requests to include an additional criterion.	"I would like to add a bit of Rihanna."	76.7
Negative Filter	User requests to negative criterion in this turn.	"I do prefer them to not have any lyrics"	3.7
Continue	User requests for more songs with the current criteria.	"Those are good songs. More like these would be great."	6.8
<i>Item Understanding Query (Question Answering)</i>			
Item Attribute Question	User questions attributes of music.	"Do you know where Samer (Artist) is from?"	0.2
<i>Feedback Response</i>			
Accept Response	User responds positively to the recommendations.	"Thank you, they are perfect"	44.4
Reject Response	User responds negatively to the recommendations.	"I still didn't get any song suggestions..."	4.8

Table 1. Taxonomy for user intents. % represents the percentage of occurrences within the user query.

System Action	Description	Example	%
<i>Request</i>			
Feedback Request	System requests the user to evaluate recommendations.	"What about these ones?"	12.8
Detail Attribute Request	System requests for the user's needs or desires for recommendations.	"Are there any particular artists you want to see?"	20.8
<i>Item Discovery Response (Retrieval / Recommendation)</i>			
Passive Recommendation	System recommends music based on user's preferences.	"Here are some pop songs for kids."	71.4
Active Recommendation	System proactively recommends music without being explicitly asked.	"Nice! I added a couple more."	14.9
<i>Item Understanding Response (Question Answering)</i>			
Item Attribute Answer	System answers to user's question with musical attributes.	"Frederic Chopin was a Polish composer..."	0.1
<i>General Response</i>			
Parroting Response	System responds to the user's inquiry by mirroring.	"Here's some picks from The Who."	25.4
Sympathetic Response	System responds to the user's inquiry with human-like sympathy.	"Excellent! Looks like a great Kickstart!"	57.2

Table 2. Taxonomy for system actions. % represents the percentage of occurrences within the system response.

is part of the 'start-dialogue' category, serving as the kick-off point for recommendation dialogues. At times, users start interactions with **Greetings** to initiate the system into a more engaging mode of communication. The 'item-discovery-query' captures user preferences for music retrieval and recommendations, subdivided into **Positive Filter** for add preferences, **Negative Filter** for discarding existing ones, or **Continue** to sustain the current preferences. **Item Attribute Question**, where users inquire about the precise attributes of music tracks, such as their genre, mood, tempo, and key/mode. Feedback responses are outlined, with users either expressing satisfaction (**Accept Response**) or dissatisfaction (**Reject Response**) to the recommended music.

3.2.2 Taxonomy for System Actions

Table 2 shows the system action taxonomy, structured into four primary categories, and seven specific intents. The 'request' category enhances the search experience, either through post-recommendation **Feedback Request** to gauge satisfaction or **Detail Attribute Request** to clarify vague or incomplete user queries. To address user queries for music discovery, the system can adopt a **Passive Recommendation** approach to comply with user requests. This approach is similar to a retrieval task because there is an explicit query from the user. The system also proactively engage through **Active Recommendation**. It is similar to a recommendation task because it implicitly uses the context of the dialogue even without an explicit query from the user.

In **Item Attribute Answer**, the system responds to users' inquiries about specific musical attribute questions

such as genre, mood, tempo, and key/mode. For general responses, the system mimics user requests in its recommendations (**Parroting Response**) to affirm that user preferences are being considered, or it may adopt a more empathetic stance (**Sympathetic Response**) to foster a more human-like interaction.

3.2.3 Taxonomy for Musical Attribute

Musical attributes such as genre, mood, and artist are closely related to user preferences. They are categorized into (1) objective metadata produced when a track is registered on the platform, (2) subjective similarity with music entities, (3) user & listening context, and (4) music content information (Table 3). Metadata is mainly associated with entity recognition [19], where **Track** refers to requests for a single music recording entity, **Artist** denotes user requests for tracks released by a specific artist, **Year** reflects the era/year in which a piece of music was released, and **Popularity** indicates the level of attention a piece of music has received. **Culture** is related to the national or regional style of the music, often linked to the artist's nationality. Unlike objective metadata, **Similar Track** and **Similar Artist** are subjective musical attributes, representing connections to other track or artist entities. The user % listening context consist of **User**, which is related to the demographics of the listener, and **Theme**, which encompasses the location, time, usage, and activities associated with listening. **Mood** refers to the emotional tone conveyed by the music or the listener's emotional state. Lastly, the categories tied closely to music content itself include **Genre**, which relates to high-level music form and style, and aspects associated with timbre such as **Instru-**

Musical Attribute	Description	Example	%
<i>Metadata</i>			
Track	A single musical work, recording, performance	Can you add montero from lil nas?	4.5
Artist	A creator or performer of music.	How about some more Justin Timberlake?	43.1
Year	The time of music’s creation or release	Could you throw in some 90s hip hop?	6.6
Popularity	The widespread acclaim of music.	Awesome! How about more male disco hits?	0.7
Culture	The national or regional influences on music.	i like Nigerian songs	0.9
<i>Similar with Music Entity</i>			
Similar Track	The song is similar to the specific track.	Can I have more songs similar to Jessie’s Girl?	1.1
Similar Artist	The song is similar to the specific artist.	Adele and Sia type of music	5.0
<i>User & Listening Context</i>			
User	The listeners characterized by demographics.	I want to create a fun playlist for the kids	1.9
Theme	A context for music listening related to location, time, usage, and activity.	hi please i need a playlist to workout with	15.2
Mood	The emotional tone conveyed by music, or the emotional state of user.	I want to create playlist for when I’m sad	6.7
<i>Music Content Information</i>			
Genre	A category of music characterized in form, style, or subject matter.	I would love to start with classical music	17.2
Instrument	A tool or device designed to produce musical sounds	What other quartet music do you suggest?	1.1
Vocal	The singing voice in music, including styles, techniques, and expressions	try some female vocalists too	1.2
Tempo	The speed or pace at which a piece of music is played	Add me some slow slow songs from juice wrld	1.3
Key / Mode	The tonal information of music	(Not appear)	0.0

Table 3. Taxonomy for musical attributes. % represents the percentage of occurrences within the user query.

ment and **Vocal**, as well as **Tempo** related to rhythm, and **Key/Mode** for tonal information.

3.2.4 Dialogue intent annotation

After the development of the taxonomy, the three annotating authors proceeded to annotate user intent, system action, and musical attributes according to the proposed taxonomy within a new sampled 30 CPCD [5] dialogues. To measure the level of agreement among the annotators for the degree of concurrence, we employed Krippendorff’s alpha [20]. The results showed high agreement levels, with Krippendorff’s alpha scores of 0.83 for user intent, 0.85 for system action, and 0.71 for musical attributes. Following this agreement phase, each annotator independently annotated a portion of the total 888 CPCD dialogues, excluding 29 error samples out of the total 917 dialogues.

3.3 Intent Analysis and Findings

The proportion of categories in each taxonomy is on the rightmost columns of Tables 1 – 3. For user intent (in Table 1), the majority of item discovery queries progress through the *Positive Filter* (76.7%), whereas the opposite, *Negative Filter*, constitute a smaller portion (3.7%). *Item Attribute Questions* are almost non-existent (0.2%). The majority of recommendation are accepted (44.4% of *Accept Response* vs 4.8% of *Reject Response*). Regarding system actions, a notable observation is that the system predominantly provides recommendations passively, rather than actively offering suggestions (71.4% vs 14.9%). In the case of general responses, the system favors eliciting human-like conversations through sympathetic responses over merely parroting back information (57.2% vs 25.4%). In the case of music, categories such as artist (43.1%), genre (17.2%), and theme (15.2%) occupy a significant portion of the user queries. This may suggest that in a *Wizard of Oz* setting [21], where the recommender utilizes music streaming platforms, the platform’s search support may be limited to these categories. Conversely, similarity queries or music content queries beyond genre are rarely employed.

4. MUSIC DISCOVERY DIALOGUE GENERATION

In this section, we describe the framework for generating music discovery dialogue consisting (1) attribute sequence generation and (2) utterance generation. This two-step approach is an improved version of previous research [6] with model-free attribute sequence generation by *Cascading music database filtering*, and utterance generation by a LLM and *Intent Analysis*.

4.1 Musical Attribute Sequence Generation

For attribute sequence generation, we employ a series of cascading filters on a multi-label annotated dataset to extract samples with overlapping semantics. Inspired by [22], which utilized *functional programs* to construct a visual reasoning dataset, we apply this approach to the domain of music discovery dialogue using user intents such as add filter, remove filter, and continue. Figure 2 illustrates an example of a cascading data filter. For example, a user may initially request songs in the *EDM* genre, narrow it down to a *party* theme, and finally specify a need for *fast tempo* music. This sequence can be represented by a series of connected functional program filters: *filter(tempo:fast, filter(theme:party, filter(genre:edm, database)))*. This method requires the types of filters and musical attributes. We derive filter types from the annotated intents (Section 3.2). For musical attributes, we initially conducted random sampling. Subsequently, to ensure diverse sampling with high co-occurrence with previous attributes, we employed top-*k* sampling ($k=20$), which involves randomly selecting a word from the top *k* words with the highest frequency. During top-*k* sampling, we include attributes from the *metadata* and *similar with music entity* categories.

4.2 Utterance Generation via Language Model

The sequence of musical attributes and annotated intent becomes prompts for LLMs to generate user and system

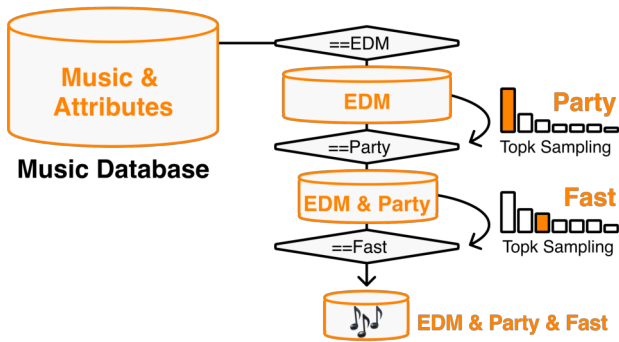


Figure 2. Cascading database filtering.

utterances. Our proposed utterance generation follows the formulation: $x_{text} = f_{LLM}(\mathcal{P}, y_{intent}, y_{music})$, where y_{intent} and y_{music} refer to the annotated intent/action (Sec 3.2) and the sampled musical attribute (Sec 4.1), respectively, and \mathcal{P} is the prompt for dialogue generation. We assumed that dialogue intent and musical attribute can serve as effective conditions for human-like dialogue generation.

5. DATASET: LP-MusicDialog

5.1 Data Source

To construct our synthetic dialogue dataset, we utilize the Million Song Dataset (MSD) [23], which has rich metadata including track details, artist information, release year, and artist familiarity. We quantize year and artist familiarity into decades and popularity, respectively.³ To cover a wide range of musical attributes as listed in Table 3, we interlink multiple annotation datasets using track IDs. For culture, mood, theme, genre, instrument, and vocals, we utilize tags from Tagtraum [24], Last.fm [25, 26], and AllMusic [27].⁴ For similar track attribute, we incorporate merged data from the Art of the Mix playlist [28] and EchoNest taste profiles. We utilize a weighted matrix factorization technique [29] to create a similarity matrix of item vectors. We then annotate the top- k similar tracks for each track ($k=128$). For artist similarity, we use cultural similarity annotation from the OLGA [30] dataset. Finally, for key/mode and tempo, we extract beats per minute (BPM), 24 key/mode using the pretrained classifier [31, 32].⁵

5.2 Creation Process

Based on the proposed pseudo dialogue generation method, we created LP-MusicDialog, an LLM-based Pseudo Music Dialogue dataset. We integrate user intents and system actions annotated in CPCD dialogues (Sec.3.2) with musical attributes obtained through cascading music database filtering (Sec.4.1) as inputs for GPT 3.5-turbo. At each cascading filtering step, we randomly sample 10

³ We categorize the top 10% of artist familiarity as high popularity, 10-30% as mid popularity, and the lower 30% as low popularity.

⁴ As the ‘style’ category in AllMusic is structured as sub-genres, we merged it with ‘genre’ category.

⁵ BPM was quantized into three text label: songs below 70 BPM were classified as slow, those between 70 BPM and 130 BPM as moderate, and those above 130 BPM as fast.

Data Source	# of Track	Musical Attributes
Million Song Dataset [23]	1,000,000	Track, Artist, Year, Popularity
TagTraum [24]	280,831	Genre, Instrument, Vocal, Mood, Theme, Culture
Last.fm [25, 26]	344,865	
AllMusic [27]	507,435	
Art of the Mix [28]	119,686	Similar Track
TasteProfile [23]	380,462	
OLGA [30]	542,364	Similar Artist
Madmom Key/Mode [32]	992,525	Key/Mode
Madmom Tempo [31]	978,759	Tempo

Table 4. Data sources for the dialogue generation

	CPCD [5]	TtW [6]	LPMD (Ours)
# of Dialog	917	1,037,701	287,675
# of Tracks	106,736	332,594	391,465
# of Vocab	9872	N/A	105,832
Avg.# of turns	5.7	5.6	4.97
Avg. query len.	54.4	80.3	63.8
Avg. response len.	45.8	N/A	87.8
Public Available	Yes	No	Yes

Table 5. Statistics of conversational music retrieval datasets. LPMD stands for LP-MusicDialog.

tracks to link with the dialogue turn. As a result, we acquire 287,675 user query, system response, music item triplets for each turn of the dialogue.

As detailed in Table 5, LP-MusicDialog is significantly larger and more diverse than existing datasets. Compared to the human dialogue dataset CPCD [5], it contains $\times 313$ larger dialogues, a more than $\times 10$ diverse vocabulary, and nearly four times more tracks. While remaining open to the public, our dataset is on par with a private dataset, TtW [6], in many aspects, with plans to expand further upon confirming active usage. In contrast, LP-MusicDialog is not only publicly available but also offers an extensive collection of connected tracks. Figure 3 shows the musical attribute ratio in dialogue. Unlike CPCD [5], where a significant portion is occupied by artist and genre due to platform bias, LP-MusicDialog shows a higher proportion of dialogues concerning music content and user listening context. Although the proportion of queries for entity recognition such as track and artist has decreased, the percentage of similarity queries has increased.

5.3 Human Evaluation

Following previous work [6], we assess the quality of our generated data through human evaluation, focusing on three key aspects: 1) Consistency - evaluating if the user preferences are coherent across dialogue turns; 2) Relevance - determining the alignment between the retrieved music items and the user query; 3) Naturalness - assessing the likelihood of such a conversation occurring in real life. Unlike previous work, we adhere to mean opinion score (MOS) that uses a 5-point Likert scale instead of a 3-point scale. A total of 26 raters evaluated 10 randomly sampled dialogues each. Within 260 total ratings, we only reported dialogues assessed by three or more raters.

Table 6 presents the MOS evaluations for dialogues from CPCD [5] and LP-MusicDialog. To understand the

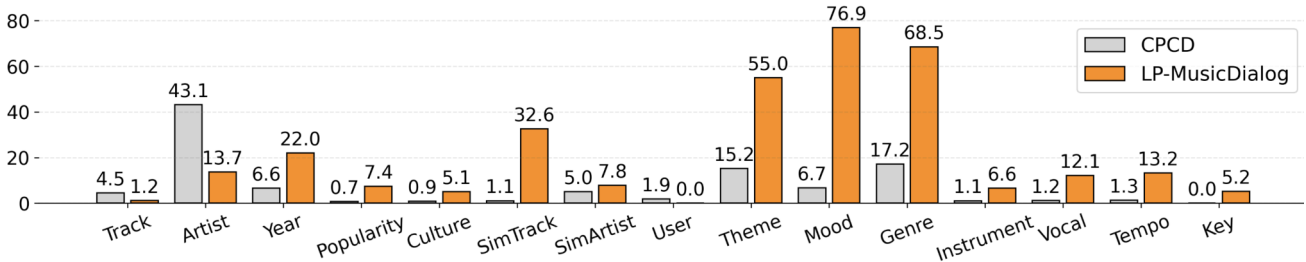


Figure 3. The ratios of musical attributes in music discovery dialogues.

	Consistency	Relevance	Naturalness
LP-MusicDialog	3.79 ± 0.56	4.04 ± 0.74	3.87 ± 0.57
+ Intent / Action	3.88 ± 0.87	4.05 ± 0.53	4.24 ± 0.38
Human Dialogue [5]	3.90 ± 0.70	4.16 ± 0.68	4.29 ± 0.43

Table 6. Mean opinion scores of the generated dialogues (LP-MusicDialog) and human dialogues (CPCD [5])

impact of user intent and system action, we conducted an ablation study synthesizing dialogues with only musical attributes and prompts. Comparing the first and second rows, we find minimal differences in consistency and relevance, as both sets of generated dialogues utilize identical musical attributes. However, a notable distinction arises in naturalness, suggesting that LLMs can foster more human-like dialogue synthesis by incorporating intents and actions. In comparing human dialogues with our generated dialogues, we found that the generated dialogues perform comparably to the human dataset across all three metrics, within the standard deviation.

6. CONVERSATIONAL MUSIC RETRIEVAL

In this section, we present a benchmark of conversational music retrieval models. Unlike prior studies [5, 6] that have music embeddings solely relying on metadata-based text modality,⁶ we expand it to include audio modality. We use a pre-train audio-text joint embedding model (TTMR++ [11]), that consists of a text encoder that handles user queries and system responses and an audio encoder that takes music tracks. We freeze TTMR++ and add a trainable MLP layer for both text and audio encoder. To handle chat history, we use a chat embedding created by average-pooling the current query, previous queries, responses, and music embeddings. Two encoders are trained to maximize the cosine similarity between the chat embedding and target music embedding using the InfoNCE [33] loss. In the inference stage, we extract chat embeddings for each turn in the same way as in the training stage and measure the similarity score with all music embeddings in the train and test splits. Based on the similarity score, we retrieve the most similar k items by nearest neighbor search.

We chose the CPCD dataset as the compared dataset and report Hit@ K as evaluation metric ($k=\{10,20,100\}$).⁷ Our baseline models are as follows: (1) BM25 [34], as a sparse retrieval baseline; (2) Contriever [35], an unsu-

⁶ {title} by {artist} from {album}

⁷ Excluding the lost audio due to YouTube crawling, we use 98,738 out of 106,736 tracks for evaluation. We use the official evaluation codebase: <https://github.com/google-research-datasets/cpcd>

Model	Type	Hit@10	Hit@20	Hit@100
BM25 [34]	-	0.180	0.251	0.433
Contriever [35]	Zeroshot	0.176	0.255	0.344
TTMR++ [11]	Zeroshot	0.201	0.275	0.505
+ LPMD only	Finetune _{OD}	0.173	0.253	0.479
+ CPCD only	Finetune _{ID}	0.209	0.295	0.530
+ LPMD & CPCD	Finetune _{OD+ID}	0.219	0.304	0.533

Table 7. Conversational music retrieval performance on CPCD Dataset. **OD** stands for out of domain. **ID** stands for in domain.

pervised dense retrieval baseline; and (3) TTMR++ [11], an audio-text joint embedding baseline. Table 7 shows the performance of conversational retrieval. Among the baselines, TTMR++ shows superior performance over BM25 and Contriever, highlighting the importance of the audio modality in the music domain. Furthermore, training TTMR++ with only LP-MusicDialog (i.e., inter-dataset evaluation) somewhat leads to a performance decrease. This is presumably due to the musical entity difference between the two datasets. Specifically, the LP-MusicDialog dataset derives from the MSD, which contains music up to the year 2010, while the CPCD dataset includes music extending up to the year 2023. Nonetheless, training with both LP-MusicDialog and CPCD results in performance improvement over training only with CPCD, suggesting the usefulness of the proposed dataset.

7. CONCLUSION

We proposed a musical dialogue generation approach with i) dialogue intent analysis using the grounded theory, ii) generating attribute sequences via cascading database filtering, and iii) generating utterances using a large language model. Our intent analysis underpins the synthesis of human-like conversations, demonstrating strengths in naturalness. Cascading filtering allows us to utilize music attributes from external music databases to generate new dialogues. The outcome, the proposed **LP-MusicDialog** covers a broader range of musical attributes and aids in the conversational music retrieval task.

However, the proposed methods have several limitations. The first is that cascading filtering is sensitive to annotation errors [36]. The second is that top-k sampling, by following the tag distribution, inevitably leads to a data imbalance problem. We hope that these limitations of cascading filtering will be addressed in future research by incorporating balanced sampling.

8. REFERENCES

- [1] K. Christakopoulou, F. Radlinski, and K. Hofmann, "Towards conversational recommender systems," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016.
- [2] R. Li, S. Ebrahimi Kahou, H. Schulz, V. Michal-ski, L. Charlin, and C. Pal, "Towards deep conversational recommendations," *Advances in neural information processing systems (NeurIPS)*, 2018.
- [3] D. Jannach, A. Manzoor, W. Cai, and L. Chen, "A survey on conversational recommender systems," *ACM Computing Surveys (CSUR)*, 2021.
- [4] Z. He, Z. Xie, R. Jha, H. Steck, D. Liang, Y. Feng, B. P. Majumder, N. Kallus, and J. McAuley, "Large language models as zero-shot conversational recommenders," in *Proceedings of the 32nd ACM international conference on information and knowledge management*, 2023.
- [5] A. T. Chaganty, M. Leszczynski, S. Zhang, R. Ganti, K. Balog, and F. Radlinski, "Beyond single items: Exploring user preferences in item sets with the conversational playlist curation dataset," in *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2023.
- [6] M. Leszczynski, S. Zhang, R. Ganti, K. Balog, F. Radlinski, F. Pereira, and A. T. Chaganty, "Talk the walk: Synthetic data generation for conversational music recommendation," *arXiv preprint arXiv:2301.11489*, 2023.
- [7] Z. Dai, A. T. Chaganty, V. Y. Zhao, A. Amini, Q. M. Rashid, M. Green, and K. Guu, "Dialog inpainting: Turning documents into dialogs," in *International Conference on Machine Learning (ICML)*, 2022.
- [8] B. G. Glaser, A. L. Strauss, and E. Strutzel, "The discovery of grounded theory; strategies for qualitative research," *Nursing research*, 1968.
- [9] I. Manco, E. Benetos, E. Quinton, and G. Fazekas, "Contrastive audio-language learning for music," *International Society for Music Information Retrieval (ISMIR)*, 2022.
- [10] S. Doh, M. Won, K. Choi, and J. Nam, "Toward universal text-to-music retrieval," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023.
- [11] S. Doh, M. Lee, D. Jeong, and J. Nam, "Enriching music descriptions with a finetuned-llm and metadata for text-to-music retrieval," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024.
- [12] S. Liu, A. S. Hussain, C. Sun, and Y. Shan, "Music understanding llama: Advancing text-to-music generation with question answering and captioning," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024.
- [13] J. Gardner, S. Durand, D. Stoller, and R. M. Bittner, "Llark: A multimodal foundation model for music," *arXiv preprint arXiv:2310.07160*, 2023.
- [14] S. Doh, K. Choi, J. Lee, and J. Nam, "LP-musiccaps: LLM-based pseudo music captioning," *International Society for Music Information Retrieval (ISMIR)*, 2023.
- [15] J. S. Downie and S. J. Cunningham, "Toward a theory of music information retrieval queries: System design implications," 2002.
- [16] D. Bainbridge, S. J. Cunningham, and J. S. Downie, "How people describe their music information needs: A grounded theory analysis of music queries," 2003.
- [17] J. H. Lee, "Analysis of user needs and information features in natural language queries seeking music information," *Journal of the American Society for Information Science and Technology*, 2010.
- [18] W. Cai and L. Chen, "Predicting user intents and satisfaction with dialogue-based conversational recommendations," in *Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization*, 2020.
- [19] E. Epure and R. Hennequin, "A human subject study of named entity recognition in conversational music recommendation queries," in *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, 2023.
- [20] K. Krippendorff, "Testing the reliability of content analysis data," *The content analysis reader*, 2009.
- [21] E. Dinan, S. Roller, K. Shuster, A. Fan, M. Auli, and J. Weston, "Wizard of wikipedia: Knowledge-powered conversational agents," *arXiv preprint arXiv:1811.01241*, 2018.
- [22] J. Johnson, B. Hariharan, L. Van Der Maaten, L. Fei-Fei, C. Lawrence Zitnick, and R. Girshick, "Clevr: A diagnostic dataset for compositional language and elementary visual reasoning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.
- [23] T. Bertin-Mahieux, D. P. Ellis, B. Whitman, and P. Lamere, "The million song dataset," in *International Society for Music Information Retrieval (ISMIR)*, 2011.
- [24] H. Schreiber, "Improving genre annotations for the million song dataset," in *International Society for Music Information Retrieval (ISMIR)*, 2015.

- [25] M. Won, S. Oramas, O. Nieto, F. Gouyon, and X. Serra, "Multimodal metric learning for tag-based music retrieval," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.
- [26] M. Won, K. Choi, and X. Serra, "Semi-supervised music tagging transformer," in *International Society for Music Information Retrieval (ISMIR)*, 2021.
- [27] A. Schindler and P. Knees, "Multi-task music representation learning from multi-label embeddings," in *Proc. International Conference on Content-Based Multimedia Indexing (CBMI)*, 2019.
- [28] B. McFee and G. R. Lanckriet, "Hypergraph models of playlist dialects," in *International Society for Music Information Retrieval (ISMIR)*, 2012.
- [29] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative filtering for implicit feedback datasets," in *2008 Eighth IEEE international conference on data mining*, 2008.
- [30] F. Korzeniowski, S. Oramas, and F. Gouyon, "Artist similarity with graph neural networks," in *International Society for Music Information Retrieval (ISMIR)*, 2021.
- [31] S. Böck, M. E. Davies, and P. Knees, "Multi-task learning of tempo and beat: Learning one to improve the other," in *International Society for Music Information Retrieval (ISMIR)*, 2019.
- [32] F. Korzeniowski and G. Widmer, "Genre-agnostic key classification with convolutional neural networks," *arXiv preprint arXiv:1808.05340*, 2018.
- [33] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv:1807.03748*, 2018.
- [34] S. Robertson, S. Walker, and S. Jones, "M. hancock-beaulieu, m., and gatford, m.(1995). okapi at trec-3," in *The Third Text REtrieval Conference (TREC-3)*, 1994.
- [35] G. Izacard, M. Caron, L. Hosseini, S. Riedel, P. Bojanowski, A. Joulin, and E. Grave, "Unsupervised dense information retrieval with contrastive learning," *arXiv preprint arXiv:2112.09118*, 2021.
- [36] K. Choi, G. Fazekas, K. Cho, and M. Sandler, "The effects of noisy labels on deep convolutional neural networks for music tagging," *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2018.

STONE: SELF-SUPERVISED TONALITY ESTIMATOR

Yuexuan Kong^{1,2}
Stella Wong

Vincent Lostanlen²
Mathieu Lagrange²

Gabriel Meseguer-Brocal¹
Romain Hennequin¹

¹ Deezer Research, Paris, France

² Nantes Université, École Centrale Nantes, CNRS, LS2N, UMR 6004, F-44000 Nantes, France

ykong@deezer.com

ABSTRACT

Although deep neural networks can estimate the key of a musical piece, their supervision incurs a massive annotation effort. Against this shortcoming, we present STONE, the first self-supervised tonality estimator. The architecture behind STONE, named ChromaNet, is a convnet with octave equivalence which outputs a “key signature profile” (KSP) of 12 structured logits. First, we train ChromaNet to regress artificial pitch transpositions between any two unlabeled musical excerpts from the same audio track, as measured as cross-power spectral density (CPSD) within the circle of fifths (CoF). We observe that this self-supervised pretext task leads KSP to correlate with tonal key signature. Based on this observation, we extend STONE to output a structured KSP of 24 logits, and introduce supervision so as to disambiguate major versus minor keys sharing the same key signature. Applying different amounts of supervision yields semi-supervised and fully supervised tonality estimators: i.e., Semi-TONEs and Sup-TONEs. We evaluate these estimators on FMAK, a new dataset of 5489 real-world musical recordings with expert annotation of 24 major and minor keys. We find that Semi-TONE matches the classification accuracy of Sup-TONE with reduced supervision and outperforms it with equal supervision.

1. INTRODUCTION

Self-taught musicians can tell whether two pieces go “in tune” or “out of tune”. To do so, they do not need to know the name of every key [1]. Meanwhile, in music information retrieval (MIR), current tonality estimators depend on a vocabulary of labels such as $C : \text{maj}$ or $F : \text{min}$.

In this context, we aim to develop models which “learn by ear” like humans; i.e., from little or no annotated data. This goal is justified in practice by the fact that online digital music corpora are larger and more musically diverse than established MIR datasets, yet often lack expert metadata.

To overcome the need for large amount of labeled data, self-supervised learning (SSL) has emerged as an alternative paradigm to supervised learning, with numerous applications in speech and music processing [2–5].

The design of pretext tasks is a long-standing issue in SSL for audio. On one hand, some of them are meant as pretraining step for general-purpose representation learning: contrastive predictive coding [6], deep metric learning [7], and self-distillation [8], to name a few. On the other hand, another family of pretext tasks is designed to suit a particular downstream task, such as tempo and pitch estimation [9, 10]. In this context, the concept of *equivariance* plays a central role. Loosely speaking, equivariance means that a certain parametric transformation of the input data forms a simple trajectory in the space of learned representations. Yet, equivariant SSL has never been used to study tonality, for lack of an adequate pretext task.

The main idea of our paper is that, even so *absolute* key labels are unknown, we can construct paired samples in which *relative* harmonic progressions serve as a learning signal for tonality estimation. Our contributions are:

STONE. To our knowledge, the first SSL framework whose model predictions correlates with key signatures. It comprises a new equivariant neural network named ChromaNet and a noncontrastive loss function based on cross-power spectral density (CPSD).

Downstream task. We extend STONE into Semi-TONE, a semi-supervised model that is tailored for 24-way key estimation. Semi-TONE performs on par with a supervised counterpart (Sup-TONE) while reducing dependency on annotated data by 90%¹.

FMAK. A new large dataset of 5489 real-world music recordings, collected from the Free Music Archive (FMA) and annotated by an expert for 24 major and minor keys, is available for free download².

2. RELATED WORK

2.1 Equivariant self-supervised learning in music

Equivariant SSL learns task-specific embeddings by representing the transformations which underlie its factors of



© Y. Kong, V. Lostanlen, G. Meseguer-Brocal, S. Wong, M. Lagrange and R. Hennequin. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Y. Kong, V. Lostanlen, G. Meseguer-Brocal, S. Wong, M. Lagrange and R. Hennequin, “STONE: Self-supervised Tonality Estimator”, in *Proc. of the 25th Int. Society for Music Information Retrieval Conf.*, San Francisco, United States, 2024.

¹ Companion website: <https://github.com/deezer/stone>

² FMAK dataset (version 2): <https://zenodo.org/records/12759100>

variability: e.g., variations in pitch or tempo [10–12]. In particular, PESTO is a monophonic pitch estimator trained by learning the pitch shift of the same sample [9]. However, its extension to multipitch tracking is an open problem [13].

2.2 Computational models of tonality

Tonality estimation remains a relatively under-researched field, due to the scarcity of labeled data available for both training and evaluation purposes. The earliest methods were based on template matching [14–17]. Later, convolutional neural networks (convnets) and transformers appeared, treating the task as a supervised 24-class classification problem [18–20]. Among studies employing the same dataset, the convnet of Korzeniowski *et al.* [21] achieves the best performance. Against the lack of annotated data, prior work proposed to integrate key estimation with unsupervised autoencoding [18, 19]. However, their approach is computationally intensive and its evaluation is limited.

2.3 Annotated datasets for key estimation

Key detection datasets face a conundrum between diversity and reproducibility. The Million Song Dataset [22] offers key labels for diverse commercial music but lacks public audio. GiantSteps MTG Key (GSMK)³, GiantSteps Key (GSK) [23] and McGill Billboard datasets [24] offer public data, yet they are restricted in terms of genres and quantity. In particular, GSMK and GSK serve in the training and evaluation of supervised SOTA [21].

3. METHODS

Figure 1 illustrates our proposed method for STONE.

3.1 Artificial pitch transpositions of the CQT

We compute a constant- Q transform (CQT) with $Q = 12$ bins per octave and center frequencies ranging between $\xi_{\min} = 27.5$ Hz and $\xi_{\max} = 2^{99/12}\xi_{\min} = 8.37$ kHz.

Given a CQT matrix \mathbf{x} and an integer $c \leq 15$, we reduce the number of rows from 99 down to 84 (7 octaves) by trimming the c lowest-frequency bins and $(15 - c)$ highest-frequency bins. This is tantamount by a pitch transposition by c semitones [9]. We denote the result by $T_c\mathbf{x}$ where $T_c\mathbf{x}[p, t] = \mathbf{x}[p - c, t]$ for each frequency $p < 84$ and time t .

3.2 ChromaNet: a convnet with octave equivalence

The cropped CQT matrix $T_c\mathbf{x}$ has a frequency range of $QJ = 84$ semitones with $Q = 12$ and $J = 7$ octaves. We define a 2-D fully convolutional network f_{θ} with trainable parameters θ , operating on $T_c\mathbf{x}$ with no pooling over the frequency dimension. The last layer has a single channel and performs global average pooling over the time dimension.

The architecture f_{θ} composes seven blocks, each of them composing a ConvNeXT block [25] and a time downsampling block, and layer normalization. It returns a vector in dimension QJ . While ConvNeXT blocks leaves the input

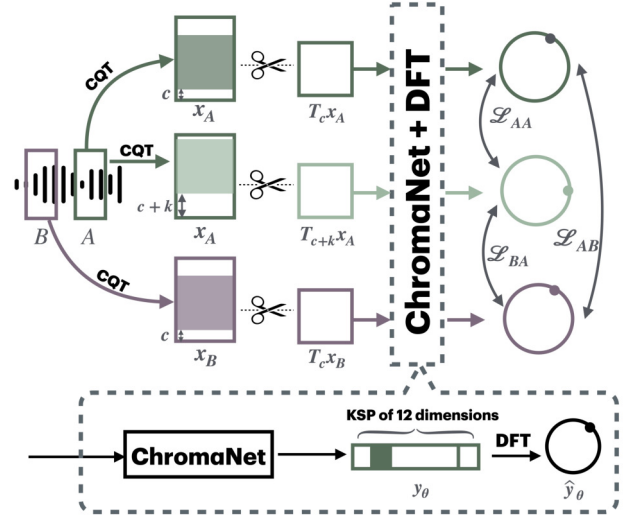


Figure 1. Overview of the equivariant pretext task in STONE. Given two segments A and B from an unlabeled musical recording, we compute their constant- Q transforms (CQT) and apply random crops by c and $(c + k)$ to simulate pitch transpositions. We feed them to ChromaNet, an equivariant neural network with octave equivalence, yielding a learned key signature profile (KSP) of 12 chromas. We compute the discrete Fourier transform (DFT) of each KSP and derive pairwise cross-power spectral densities (CPSD). Self-supervised losses \mathcal{L}_{AA} , \mathcal{L}_{AB} , and \mathcal{L}_{BA} are formulated as CPSD regression residuals in the complex domain.

resolution unchanged, time downsampling blocks decrease time resolution while preserving frequency resolution.

We compose the convnet f_{θ} with a non-trainable operator g whose role is to guarantee octave equivalence. We roll the log-frequency axis into a spiral which makes a full turn at every octave, thus aligning coefficients in $f_{\theta}(T_c\mathbf{x})$ of the form $(p \pm Qj)$ for integer j . The operator g sums these coefficients across octaves j for each pitch class q and applies a softmax transformation. We obtain a Q -dimensional vector \mathbf{y}_{θ} whose entries are nonnegative and sum to one. We propose to call this vector a “key signature profile” (KSP):

$$\begin{aligned} \mathbf{y}_{\theta}[q] &= (g \circ f_{\theta})(T_c\mathbf{x})[q] \\ &= \frac{\exp\left(\sum_{j=0}^{J-1} f_{\theta}(T_c\mathbf{x})[Qj + q]\right)}{\sum_{q'=0}^{Q-1} \exp\left(\sum_{j=0}^{J-1} f_{\theta}(T_c\mathbf{x})[Qj + q']\right)}. \end{aligned} \quad (1)$$

For brevity, we do not recall the dependency of \mathbf{y}_{θ} upon \mathbf{x} nor c . Equation 1 resembles the extraction of chroma features [26], hence the proposed name of ChromaNet.

3.3 DFT over key signature profiles

With $Q = 12$, the discrete Fourier transform (DFT) of the KSP \mathbf{y}_{θ} is

$$\hat{\mathbf{y}}_{\theta}[\omega] = \mathcal{F}\{\mathbf{y}_{\theta}\}[\omega] = \sum_{q=0}^{11} \mathbf{y}_{\theta}[q] e^{-2\pi i \omega q / 12}, \quad (2)$$

where ω is an integer between 0 and 11 that is coprime to 12 for a full circular distribution over all 12 pitches. With

³ <https://github.com/GiantSteps/giantsteps-mtg-key-dataset>

$\omega = 7$, a circular shift of \mathbf{y}_θ by seven chromas corresponds to a multiplication of $\hat{\mathbf{y}}_\theta[\omega]$ by $e^{2\pi i 49/12} = e^{2\pi i/12}$. Hence, the phase of the complex number $\hat{\mathbf{y}}_\theta[\omega]$ denotes a key modulation in the circle of fifths (CoF). Alternatively, $\omega = 1$ would correspond to a circle of semitones. Our paper evaluates both settings but only describes the CoF setting ($\omega = 7$) for the sake of conciseness.

3.4 Cross-power spectral density (CPSD)

Let us split the CQT matrix \mathbf{x} into two disjoint time segments of equal length: $\mathbf{x} = (\mathbf{x}_A, \mathbf{x}_B)$. We denote the ChromaNet response for A by $\mathbf{y}_{\theta,A} = (g \circ f_\theta)(T_c \mathbf{x})$ idem for B. The circular cross-correlation between $\mathbf{y}_{\theta,A}$ and $\mathbf{y}_{\theta,B}$ is

$$\mathbf{R}_{\mathbf{y}_{\theta,A}, \mathbf{y}_{\theta,B}}[k] = \sum_{q=0}^{Q-1} \mathbf{y}_{\theta,A}[q] \mathbf{y}_{\theta,B}[(q+k) \bmod Q] \quad (3)$$

for $0 \leq k < 12$. Taking the DFT of the equation above yields the circular cross-power spectral density (CPSD)

$$\hat{\mathbf{R}}_{\mathbf{y}_{\theta,A}, \mathbf{y}_{\theta,B}}[\omega] = \mathcal{F}\{\mathbf{R}_{\mathbf{y}_{\theta,A}, \mathbf{y}_{\theta,B}}\}[\omega] = \hat{\mathbf{y}}_{\theta,A}[\omega] \hat{\mathbf{y}}_{\theta,B}^*[\omega], \quad (4)$$

where the asterisk denotes complex conjugation.

3.5 Differentiable distance over the circle of fifths

Given a constant DFT frequency $\omega = 7$ and an arbitrary musical interval k in semitones, we compute the CPSD associated to the pair $(\mathbf{y}_{\theta,A}, \mathbf{y}_{\theta,B})$ and measure its half squared Euclidean distance to $e^{-2\pi i \omega k / Q}$ in the complex domain:

$$\mathcal{D}_{\theta,k}(\mathbf{x}_A, \mathbf{x}_B) = \frac{1}{2} \left| e^{-2\pi i \omega k / Q} - \hat{\mathbf{R}}_{\mathbf{y}_{\theta,A}, \mathbf{y}_{\theta,B}}[\omega] \right|^2. \quad (5)$$

Intuitively, in the case where $\hat{\mathbf{y}}_{\theta,A}[\omega]$ and $\hat{\mathbf{y}}_{\theta,B}[\omega]$ are both one-hot-encoding of 12 dimensions, they will be mapped as complex numbers of module 1 on the border of the CoF, $\hat{\mathbf{R}}_{\mathbf{y}_{\theta,A}, \mathbf{y}_{\theta,B}}[\omega]$ measures the difference of phases on the CoF. Then, $\mathcal{D}_{\theta,k}(\mathbf{x}_A, \mathbf{x}_B)$ measures its deviation from the DFT basis vector $e^{-2\pi i \omega k / Q}$, which corresponds to the actual pitch shift k on the CoF. This distance is differentiable with respect to the weight vector θ .

3.6 Invariance loss

Although the contents of \mathbf{x}_A versus \mathbf{x}_B may differ in terms of melody, rhythm, and instrumentation, we assume them to be in the same key. This implies that ChromaNet responses $\mathbf{y}_{\theta,A}$ and $\mathbf{y}_{\theta,B}$ should be maximally correlated at the unison interval $k = 0$ and decorrelated for $k \neq 0$. In other words, the CPSD at the frequency ω should be maximal; i.e., equal to one. Thus, given an arbitrary pitch interval c , we define an *invariance loss* \mathcal{L}_{AB} , defined as the distance between $T_c \mathbf{x}_A$ and $T_c \mathbf{x}_B$ on the CoF. We obtain⁴:

$$\mathcal{L}_{AB}(\theta | \mathbf{x}, c) = \mathcal{D}_{\theta,0}(T_c \mathbf{x}_A, T_c \mathbf{x}_B) \quad (6)$$

⁴ In this paper, we use the vertical bar notation so as to clearly separate neural network parameters on the left versus data on the right.

3.7 Equivariance loss

We want the model f_θ to be equivariant to pitch transpositions. Hence, we define an *equivariance loss* $\mathcal{L}_{c,k}^{AA}$ as the distance between $T_c \mathbf{x}_A$ and $T_{(c+k)} \mathbf{x}_A$ on the CoF:

$$\mathcal{L}_{AA}(\theta | \mathbf{x}, c, k) = \mathcal{D}_{\theta,k}(T_c \mathbf{x}_A, T_{c+k} \mathbf{x}_A). \quad (7)$$

In theory, setting the architecture of f_θ to a ChromaNet should lead to equivariance by design, for any value of the weight vector θ . Yet, in practice, we observed that some values of θ break this property of equivariance, likely due to boundary artifacts in 2-D convolutions—a similar observation to PESTO [9]. For STONE, only minimizing the invariance loss \mathcal{L}_c^{AB} causes the ChromaNet to collapse and predict a constant one-hot vector regardless of audio input \mathbf{x} under certain hyperparameter choices, particularly for $\omega = 1$. To prevent this collapse, we penalize f_θ with the equivariance loss in Equation 7.

3.8 Combined invariance and equivariance loss

In addition, we penalize f_θ according to the following loss:

$$\mathcal{L}_{BA}(\theta | \mathbf{x}, c, k) = \mathcal{D}_{\theta,k}(T_c \mathbf{x}_B, T_{c+k} \mathbf{x}_A), \quad (8)$$

i.e., the distance between ChromaNet responses $T_c \mathbf{x}_B$ and $T_{(c+k)} \mathbf{x}_A$ on the CoF. Observe that both these responses are already available after Equations 6 and 7. Therefore, the inclusion of Equation 8 in the loss comes at almost no extra computational cost during gradient backpropagation.

4. SELF-SUPERVISED KEY SIGNATURE PROFILES

4.1 Training on real-world unlabeled data

We collect 60k songs from the catalog of a music streaming service, with due permission. For each of them, we extract two disjoint segments \mathbf{x}_A and \mathbf{x}_B of duration equal to 15 seconds each. Following prior knowledge in music cognition [27], we set this duration to be as large as possible, considering the memory constraints of GPU hardware.

We implement ChromaNet and CPSD in PyTorch. The interval c (see Section 3.1) varies between zero and 15 semitones while the interval k (see Section 3.7) varies between -12 and 12 semitones. We define a CPSD-based stochastic loss function by combining Equations 6, 7, and 8:

$$\mathcal{L}^{\text{CPSD}}(\theta | \mathbf{x}, k, c) = \mathcal{L}_{AB}(\theta | \mathbf{x}, c) + \mathcal{L}_{AA}(\theta | \mathbf{x}, k, c) + \mathcal{L}_{BA}(\theta | \mathbf{x}, k, c), \quad (9)$$

where CQT samples \mathbf{x} and intervals c and k are drawn independently and uniformly at random. We train the ChromaNet for 50 epochs using a cosine learning rate schedule with a linear warm-up. We use an AdamW optimizer with a learning rate of 10^{-3} and a batch size of 128.

The self-supervised procedure above learns an approximately equivariant mapping from CQT to key signature profiles (KSP). After training, we observe informally that for each input \mathbf{x} , most of the softmax activation in the KSP \mathbf{y}_θ is concentrated on a single pitch class. In other words, the loss $\mathcal{L}^{\text{CPSD}}$ (Equation 9) is sparsity-promoting.

4.2 Calibration on a C major scale

By learning to predict pitch transpositions between segments, STONE learns the notion of relative tonality, just like the relative pitch of musicians; however, it lacks a notion of absolute tonality. Thanks to the equivariant property of ChromaNet, we only need to introduce this notion via a single recording of a C major scale paired with C major chords. This calibration procedure resembles previous work in self-supervised fundamental frequency estimation [9, 11].

Denoting the C:maj calibration sample by \mathbf{x}_{cal} , we look up the index of its highest KSP coefficient in STONE:

$$q_{\text{cal}}(\boldsymbol{\theta}) = \arg \max_{0 \leq q < Q} (g \circ f_{\boldsymbol{\theta}})(\mathbf{x}_{\text{cal}})[q]. \quad (10)$$

Then, given a CQT matrix \mathbf{x} from the test set, we realign its ChromaNet response $\mathbf{y} = (g \circ f_{\boldsymbol{\theta}})(\mathbf{x})$ via a pitch transposition of the learned KSP by $q_{\text{cal}}(\boldsymbol{\theta})$ semitones:

$$h_{\boldsymbol{\theta}}(\mathbf{y})[q] = \mathbf{y}[(q - q_{\text{cal}}(\boldsymbol{\theta})) \bmod Q]. \quad (11)$$

4.3 Evaluation on real-world labeled data (FMAK)

FMAK is a subset of Free Music Archive dataset [28] containing 5489 songs that present a clear key and are in major or minor modes. We label these songs by ear. Each of the 24 keys is represented in FMAK by at least 89 songs. C:maj and A:min are the best represented, while G#:maj and G#:min are the least represented. Songs are distributed in major and minor modes evenly. Rock and electronic dance music are the best represented genres; jazz and blues are the least represented. To our knowledge, FMAK stands as the largest and most diverse MIR dataset with key annotation.

4.4 Key signature estimation accuracy (KSEA)

In compliance with MIREX [29], we propose the following figure of merit for key signature estimation:

$$\text{KSEA}(\boldsymbol{\theta}) = \frac{1}{N} \sum_{n=0}^{N-1} \left(\boldsymbol{\delta}[s_n(\boldsymbol{\theta}) - s_n^{\text{ref}}] + \frac{1}{2} \boldsymbol{\delta}[|s_n(\boldsymbol{\theta}) - s_n^{\text{ref}}| - 6| - 1] \right), \quad (12)$$

where $s_n(\boldsymbol{\theta}) = \arg \max_{0 \leq q < Q} (h_{\boldsymbol{\theta}} \circ g \circ f_{\boldsymbol{\theta}})(\mathbf{x}_n)[q]$ and $\boldsymbol{\delta}$ is the Kronecker symbol. KSEA assigns a full point to the prediction if it matches the reference and a half point if the prediction is one perfect fifth above or below the reference.

4.5 Results on key signature estimation

We evaluate two variants of STONE on FMAK, all other things being equal: $\omega = 7$ (CoF) and $\omega = 1$. For comparison, we also evaluate the work of Korzeniowski *et al.*, the supervised state of the art (SOTA) for this task [21], a convnet trained on GSMK. Lastly, we evaluate a feature engineering pipeline, requiring no supervision: i.e., we take the global average of the chromagram representation and extract the pitch class with highest energy.

Table 1 summarizes our results. We observe that, for both values of the CPSD frequency ω , STONE outperforms the feature engineering baseline. Furthermore, for $\omega = 1$, the KSEA approaches that of the supervised SOTA.

Table 1. Evaluation of self-supervised models on FMAK. KSEA denotes key signature estimation accuracy. We also report the supervised state of the art (SOTA) for comparison.

	Correct	Fifth	KSEA
Feature engineering	1599	981	38%
STONE ($\omega = 7$)	3587	1225	77%
STONE ($\omega = 1$)	3883	920	79%
Supervised SOTA [21]	4090	741	81%

Table 2. Ablation study of STONE ($\omega = 7$) on FMAK. CPSD denotes cross-power spectral density. KSEA denotes key signature estimation accuracy. We report a naive baseline (i.e., predict the key signature of C:maj and A:min for every sample) for comparison.

	Correct	Fifth	KSEA
STONE ($\omega = 7$)	3587	1225	77%
w/o octave equivalence	1052	1267	31%
w/o CPSD	1049	1267	31%
Baseline (predict C)	1049	1267	31%

4.6 Ablation study

The two main novel components of STONE are the ChromaNet (Section 3.2) on one hand and cross-power spectral density (CPSD) over learned key signature profiles (KSP) on the other hand. In order to evaluate their relative performance, we conduct an ablation study: i.e., we substitute them by more conventional alternatives.

First, we replace the non-learned octave equivalence layer g (Equation 1) by a fully connected layer with same output size. Secondly, we replace the three CPSD-based losses (Equation 9) by 12-class cross-entropy losses. Intuitively, the first ablation disables equivariance in $f_{\boldsymbol{\theta}}$ while the second disables equivariance in \mathcal{L} . We observe that both ablations cause a collapse of SSL, leading it to predict the majority class (i.e., C:maj) on almost every sample. This suggests that both octave equivalence and CPSD are essential to the success of STONE.

5. SELF-SUPERVISED TONALITY ESTIMATION

5.1 Structured prediction

After having established that STONE learns to represent key signatures without any supervision, we turn to study its transferability to the well-known MIR problem of key estimation. For this purpose, we must accommodate the distinction between major and minor keys, thus doubling the output dimension of the ChromaNet from 12 to 24.

We note that key signature and mode are orthogonal concepts: each major key has exactly one relative minor and vice versa. These considerations suggest that the downstream task of key estimation may be formulated as structured prediction: i.e., in a 2-D label space. We encode structured labels in a matrix \mathbf{Y} with 12 rows and two columns.

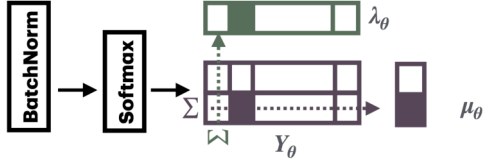


Figure 2. We modify the ChromaNet architecture of Figure 1 to accommodate structured prediction key signature and mode. We apply batch normalization per mode m and softmax over all coefficients, yielding a 12×2 matrix $\mathbf{Y}_\theta(\mathbf{x})$. Summing $\mathbf{Y}_\theta(\mathbf{x})$ over modes m yields a learned key signature profile $\lambda_\theta(\mathbf{x})$ in dimension 12; summing $\mathbf{Y}_\theta(\mathbf{x})$ over chromas q yields a pitch-invariant 2-dimensional vector $\mu_\theta(\mathbf{x})$.

5.2 Batch normalization across key signatures

We modify the last layer of the ChromaNet f_θ to output two channels instead of one. We also redefine the non-learnable operator g for octave equivalence to accommodate two channels, apply batch normalization with non-learnable parameters on each channel, and a softmax nonlinearity over all batch-normalized coefficients. This procedure normalizes each channel to null mean and unit variance over the training set, thus ensuring that both channels are activated and thus prevents a form of collapse during self-supervision.

The composition of g and f_θ , under their new definitions, yields a matrix $\mathbf{Y}_\theta(\mathbf{x})$ with $Q = 12$ rows and two columns. By property of the softmax in g , all 24 coefficients in $\mathbf{Y}_\theta(\mathbf{x})$ are positive and sum to one. As illustrated in Figure 2, we take advantage of this property to derive a key signature estimator λ_θ and a mode estimator μ_θ , respectively defined as row-wise and column-wise partial sums of $\mathbf{Y}_\theta(\mathbf{x})$:

$$\lambda_\theta(\mathbf{x})[q] = \sum_{m=0}^1 \mathbf{Y}_\theta(\mathbf{x})[q, m] \quad (13)$$

$$\mu_\theta(\mathbf{x})[m] = \sum_{q=0}^{11} \mathbf{Y}_\theta(\mathbf{x})[q, m]. \quad (14)$$

We verify that the 12-dimensional vector $\lambda_\theta(\mathbf{x})$ is positive, sums to one, and is *equivariant* to pitch transpositions in $\mathbf{Y}_\theta(\mathbf{x})$. Conversely, the 2-dimensional vector $\mu_\theta(\mathbf{x})$ is positive, sums to one, and is *invariant* to pitch transpositions in $\mathbf{Y}_\theta(\mathbf{x})$. We use λ_θ as a substitute for $(g \circ f_\theta)$ in $\mathcal{L}^{\text{CPSD}}$.

5.3 Self-supervised mode estimation

We now introduce a loss for self-supervised mode estimation. To this aim, we posit that mode is not only constant throughout the musical piece, but also remains invariant by pitch transposition. Therefore, going back to the notations from Section 4: $T_c \mathbf{x}_A$, $T_c \mathbf{x}_B$, and $T_{c+k} \mathbf{x}_B$ should elicit the same value of the mode estimator μ_θ .

We recall the definition of binary cross-entropy (BCE) for 2-D vectors whose entries are positive and sum to one:

$$\text{BCE}(\boldsymbol{\mu}, \boldsymbol{\mu}') = -\boldsymbol{\mu}[0] \log \boldsymbol{\mu}'[0] - \boldsymbol{\mu}[1] \log \boldsymbol{\mu}'[1]. \quad (15)$$

We compute the pairwise BCE between mode estimator responses associated to the three predictions of the self-supervised ChromaNet (see Figure 1)⁵:

$$\begin{aligned} \mathcal{L}^{\text{BCE}}(\boldsymbol{\theta} | \mathbf{x}, c, k) &= \text{BCE}(\mu_\theta(T_c \mathbf{x}_B), \mu_\theta(T_c \mathbf{x}_A)) \\ &\quad + \text{BCE}(\mu_\theta(T_c \mathbf{x}_A), \mu_\theta(T_{c+k} \mathbf{x}_A)) \\ &\quad + \text{BCE}(\mu_\theta(T_c \mathbf{x}_B), \mu_\theta(T_{c+k} \mathbf{x}_A)). \end{aligned} \quad (16)$$

We add the BCE-based loss in Equation 16 to the CPSD-based loss in Equation 9, thus yielding a full-fledged loss for self-supervised tonality estimation (STONE):

$$\mathcal{L}(\boldsymbol{\theta} | \mathbf{x}, c, k) = \mathcal{L}^{\text{CPSD}}(\boldsymbol{\theta} | \mathbf{x}, c, k) + \mathcal{L}^{\text{BCE}}(\boldsymbol{\theta} | \mathbf{x}, c, k). \quad (17)$$

We train the modified ChromaNet to minimize \mathcal{L} with the same optimization hyperparameters as in Section 4.1. The resulting model, named 24-STONE, performs key signature estimation with λ_θ and mode estimation with μ_θ . However these estimators are uncalibrated: i.e., λ_θ only contains information of relative tonalities and μ_θ may swap relative major and minor. We calibrate them by means of a C:maj scale, via the same procedure as in Section 4.2.

5.4 Results on key and mode classification

We evaluate two variants of 24-STONE on FMAK: $\omega = 1$ and $\omega = 7$; as well as the supervised SOTA. We also evaluate the template matching algorithm of [17], requiring behavioral data but no supervision.

Table 3 summarizes our results. The 24-STONE model with $\omega = 7$ is best in the unsupervised category, although well below the supervised SOTA. However, setting ω to 1 dramatically hurts the MIREX score of 24-STONE, placing it below the naive baseline. Thus, formulating CPSD regression over the CoF (see Section 3.5) seems necessary for STONE to transfer to key and mode estimation, even so it is outperformed by $\omega = 1$ in KSEA (see Section 4.4). With this result in mind, we set $\omega = 7$ in the rest of this paper.

6. SEMI-SUPERVISED TONALITY ESTIMATION

6.1 Supervising the ChromaNet

Thanks to structured prediction, the ChromaNet accommodates supervised training in the same label space as self-supervised training. Note that the 24-STONE losses $\mathcal{L}^{\text{CPSD}}$ and \mathcal{L}^{BCE} involves pairwise comparisons between three items belonging to the same \mathbf{x} : i.e., two transposed versions of the same segment \mathbf{x}_A and one from another segment \mathbf{x}_B . In this context, a simple way to introduce supervision is to replace the responses $\lambda_{\theta, B}$ and $\mu_{\theta, B}$ by “oracles” λ_{ref} and μ_{ref} which are informed by the ground truth.

Given the ground truth key signature q_{ref} and mode m_{ref} , one-hot encoding yields the sparse vectors $\lambda_{\text{ref}, c}(\mathbf{x})[q] = \boldsymbol{\delta}[(q - q_{\text{ref}} - c) \bmod 12]$ and $\mu_{\text{ref}}(\mathbf{x})[m] = \boldsymbol{\delta}[m - m_{\text{ref}}]$, where c is a pitch interval in semitones (see Section 3.1). We use these oracles to write supervised variants of losses

⁵ Compared to $\mathcal{L}^{\text{CPSD}}$, we have swapped \mathbf{x}_A with \mathbf{x}_B in the first term. This is for compatibility with the supervised setting, as described in Section 6.1, so as to avoid an undefined BCE due to a logarithm of zero.

	Correct	Fifth	Relative	Parallel	Wrong	MIREX
Template matching [17]	2398	631	390	506	1564	53.4%
24-STONE ($\omega = 1$)	421	535	399	253	3881	15.6%
24-STONE ($\omega = 7$)	2443	628	1320	115	983	57.9%
Supervised SOTA [21]	3586	482	504	165	752	73.1%
Baseline (predict C : ma j)	551	568	498	286	3586	19.0%

Table 3. Evaluation of self-supervised models for key and mode estimation on FMAK. We also report the supervised state of the art (SOTA) [21] and a naive baseline (i.e., predict C : ma j for every sample) for comparison. See Section 5.4 for details.

\mathcal{L}^{CPSD} and \mathcal{L}^{BCE} . This seamless switch from SSL to supervised learning requires no change of architecture nor optimizer. Thus, instead of using supervision as fine-tuning, we propose an alternated scheme: one epoch of SSL followed by one epoch of supervised learning, and so on.

6.2 Semi-TONE and Sup-TONE

Introducing supervision into 24-STONE yields a semi-supervised tonality estimator, or Semi-TONE for short. We alternate between self-supervised epochs on 60k unlabeled recordings (see Section 4.1) and supervised epochs on the 1159 songs in GSMK in which annotators agree. For the sake of comparison, we experiment with disabling SSL and training the ChromaNet directly on GSMK: hence a fully supervised tonality estimator, or Sup-TONE for short.

To compare their ability to learn from limited labeled data, we retrain Semi-TONE and Sup-TONE after subsampling GSMK at random by factors of 10 and 100.

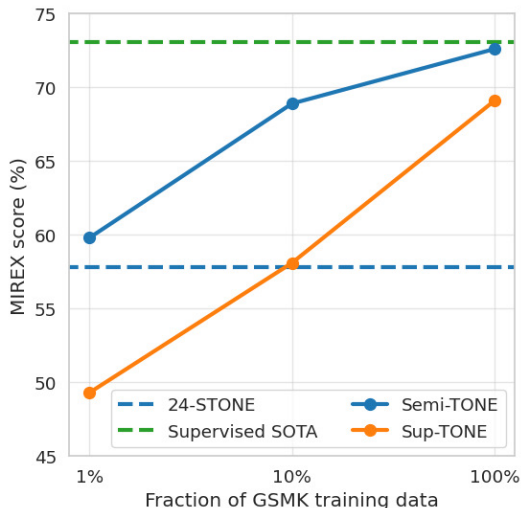


Figure 3. Evaluation of self-supervised (dashed blue), semi-supervised (solid blue), and supervised models (orange) on FMAK. All models use $\omega = 7$. We also report the supervised state of the art (SOTA) [21] in dashed green.

6.3 Results on key and mode classification

Figure 3 summarizes our results. We observe that Semi-TONE systematically outperforms Sup-TONE at any amount of training data. In particular, training Semi-TONE with 10% of GSMK leads to a comparable MIREX score as training Sup-TONE with 100% of GSMK. This result

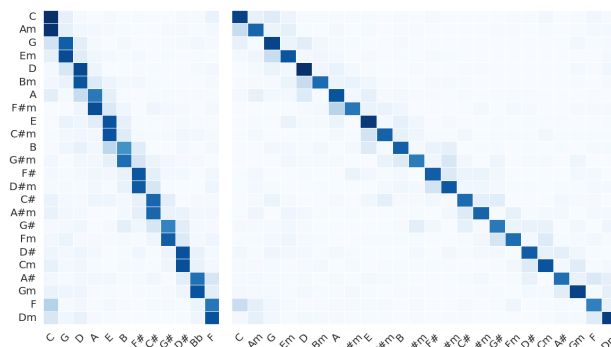


Figure 4. Confusion matrices of STONE (left, 12 classes) and Semi-TONE (right, 24 classes) on FMAK, both using $\omega = 7$. The axis correspond to model prediction and reference respectively, keys arranged by proximity in the CoF and relative modes. Deeper colors indicate more frequent occurrences per relative occurrence per reference key.

confirms the interest of our proposed pretext tasks towards the overarching goal of reducing human annotation effort.

Training Semi-TONE on the full GSMK dataset yields a MIREX score of 72.6%; i.e., roughly on par with the supervised SOTA (73.1%). Figure 4 shows the confusion matrix of calibrated STONE and Semi-TONE on FMAK. Although our methods do not outperform the SOTA on key estimation, it brings insights into a novel framework that does not require high supervision for training. Moreover, we note that self-supervision remains beneficial even when the full GSMK dataset is available for training. Therefore, a promising avenue of research is to scale up the dataset of unlabeled recordings (see Section 4.1), thus widening the gap between Semi-TONE and Sup-TONE on FMAK.

7. CONCLUSION

STONE learns key signature profiles (KSP) via equivariant self-supervised learning in the time–frequency domain. We have seen that a semi-supervised extension of STONE (semi-TONE) reduces expert annotation by 90% less at no loss of MIREX score compared to the fully supervised variant (sup-TONE). The primary limitation of our work resides in the inability of the STONE objective (CPSD, i.e., cross-power spectral density) to distinguish major keys from minor keys. Future work will study how STONE can be adapted to other pitch-relative MIR tasks.

8. REFERENCES

- [1] D. Müllensiefen, B. Gingras, J. Musil, and L. Stewart, “The musicality of non-musicians: An index for assessing musical sophistication in the general population,” *PLOS ONE*, vol. 9, no. 2, p. e89642, 2014.
- [2] H. Zhu, Y. Niu, D. Fu, and H. Wang, “MusicBERT: A self-supervised learning of music representation,” in *Proceedings of the ACM International Conference on Multimedia (MM)*, 2021, pp. 3955–3963.
- [3] J. Spijkervet and J. A. Burgoyne, “Contrastive learning of musical representations,” in *Proceedings of the International Society for Music Information Retrieval Late-Breaking/Demo Session (ISMIR-LBD)*, 2021.
- [4] D. Desblancs, V. Lostanlen, and R. Hennequin, “Zero-note samba: Self-supervised beat tracking,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.
- [5] G. Meseguer-Brocal, D. Desblancs, and R. Hennequin, “An experimental comparison of multi-view self-supervised methods for music tagging,” in *Proceedings of the IEEE International Conference on Audio, Speech and Signal Processing (ICASSP)*, 2024.
- [6] S. Schneider, A. Baeviski, R. Collobert, and M. Auli, “wav2vec: Unsupervised Pre-training for Speech Recognition,” in *Proceedings of INTERSPEECH*, 2019.
- [7] A. Saeed, D. Grangier, and N. Zeghidour, “Contrastive learning of general-purpose audio representations,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021.
- [8] D. Niizumi, D. Takeuchi, Y. Ohishi, N. Harada, and K. Kashino, “BYOL for audio: Self-supervised learning for general-purpose audio representation,” in *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN)*, 2021.
- [9] A. Riou, S. Lattner, G. Hadjeres, and G. Peeters, “PESTO: Pitch estimation with self-supervised transposition-equivariant objective,” in *Proceedings from the International Society for Music Information Retrieval Conference (ISMIR)*, 2023.
- [10] E. Quinton, “Equivariant self-supervision for musical tempo estimation,” in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2022.
- [11] B. Gfeller, C. Frank, D. Roblek, M. Sharifi, M. Tagliasacchi, and M. Velimirović, “SPICE: Self-supervised pitch estimation,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 1118–1128, 2020.
- [12] G. Morais, M. E. Davies, M. Queiroz, and M. Fuentes, “Tempo vs. pitch: understanding self-supervised tempo estimation,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.
- [13] F. Cwitkowitz and Z. Duan, “Toward fully self-supervised multi-pitch estimation,” *arXiv preprint arXiv:2402.15569*, 2024.
- [14] K. Noland and M. Sandler, “Signal processing parameters for tonality estimation,” in *Proceedings of the Audio Engineering Society Convention (AES)*, 2007.
- [15] S. Pauws, “Musical key extraction from audio,” in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2004.
- [16] Á. Faraldo, E. Gómez, S. Jordà, and P. Herrera, “Key estimation in electronic dance music,” in *Proceedings of the European Conference on Information Retrieval (ECIR)*, 2016.
- [17] C. L. Krumhansl, *Cognitive foundations of musical pitch*. Oxford University Press, 2001.
- [18] Y. Wu, E. Nakamura, and K. Yoshii, “A variational autoencoder for joint chord and key estimation from audio chromagrams,” in *Proceedings of the Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2020.
- [19] Y. Wu and K. Yoshii, “Joint chord and key estimation based on a hierarchical variational autoencoder with multi-task learning,” *APSIPA Transactions on Signal and Information Processing*, 2022.
- [20] H. Schreiber and M. Müller, “Musical tempo and key estimation using convolutional neural networks with directional filters,” in *Proceedings of the International Sound and Music Computing Conference (SMC)*, 2019.
- [21] F. Korzeniowski and G. Widmer, “Genre-agnostic key classification with convolutional neural networks,” in *Proceedings of the International Society on Music Information Conference (ISMIR)*, 2018.
- [22] T. Bertin-Mahieux, D. P. Ellis, B. Whitman, and P. Lamere, “The million song dataset,” in *Proceedings of the International Society on Music Information Retrieval Conference (ISMIR)*, 2011.
- [23] P. Knees, Á. Faraldo, P. Herrera, R. Vogl, S. Böck, F. Hörschläger, and M. L. Goff, “Two data sets for tempo estimation and key detection in electronic dance music annotated from user corrections,” in *International Society for Music Information Retrieval Conference*, 2015. [Online]. Available: <https://api.semanticscholar.org/CorpusID:15836728>
- [24] J. A. Burgoyne, J. Wild, and I. Fujinaga, “An expert ground truth set for audio chord recognition and music analysis,” in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, A. Klapuri and C. Leider, Eds. University of Miami, 2011.

- [25] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, “A ConvNet for the 2020s,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, 2022, pp. 11 976–11 986.
- [26] M. Müller, D. P. Ellis, A. Klapuri, and G. Richard, “Signal processing for music analysis,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 6, pp. 1088–1110, 2011.
- [27] P. Loui and D. Wessel, “Harmonic expectation and affect in Western music: Effects of attention and training,” *Perception & Psychophysics*, vol. 69, pp. 1084–1092, 2007.
- [28] M. Defferrard, K. Benzi, P. Vandergheynst, and X. Bresson, “FMA: A dataset for music analysis,” in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2017.
- [29] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, D. P. Ellis, and C. C. Raffel, “mir_eval: A Transparent Implementation of Common MIR Metrics.” in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2014.

BEAT THIS!

ACCURATE BEAT TRACKING WITHOUT DBN POSTPROCESSING

Francesco Foscarin^{*1,2}

Jan Schlüter^{*1}

Gerhard Widmer^{1,2}

¹ Johannes Kepler University, Linz, Austria

² LIT AI Lab, Linz Institute of Technology, Austria

firstname.lastname@jku.at

ABSTRACT

We propose a system for tracking beats and downbeats with two objectives: generality across a diverse music range, and high accuracy. We achieve generality by training on multiple datasets – including solo instrument recordings, pieces with time signature changes, and classical music with high tempo variations – and by removing the commonly used Dynamic Bayesian Network (DBN) postprocessing, which introduces constraints on the meter and tempo. For high accuracy, among other improvements, we develop a loss function tolerant to small time shifts of annotations, and an architecture alternating convolutions with transformers either over frequency or time. Our system surpasses the current state of the art in F1 score despite using no DBN. However, it can still fail, especially for difficult and underrepresented genres, and performs worse on continuity metrics, so we publish our model, code, and preprocessed datasets, and invite others to beat this.

1. INTRODUCTION

Beat tracking is the task of estimating the temporal locations of musical beats in an audio signal. It is often combined with the downbeat tracking task, which targets a higher metrical level: tracking the beginning of each measure. Despite being one of the long-standing problems in the Music Information Retrieval (MIR) field, it still attracts attention and several approaches were proposed in recent years [1–8]. Most recent work follows a common pipeline: the audio files are transformed into some spectrogram-like representation, then a deep neural network predicts frame-wise beat and downbeat probabilities, which are postprocessed to obtain the final predictions. The most widely used postprocessing technique is the Dynamic Bayesian Network (DBN) in the form proposed by Böck et al. [9]. It addresses four tasks: variable threshold peak-picking, forcing the tempo to stay in a certain range (i.e., limiting the allowed distance

between beats), limiting sudden tempo changes, and (for downbeat tracking) ensuring that the downbeat falls every n beats, where n is constant for a piece of music and is selected from a limited list of values.

We argue that the DBN is a problematic component because it is inherently bound to fail for several music pieces: pieces with time signature changes, pieces whose tempo falls outside of the tempo range (or that slow down/speed up outside the tempo range), and pieces whose number of beats per measure are not included in the list of supported values. Moreover, it has a fixed parameter controlling allowed tempo variations, although we can expect, for example, classical music to have bigger tempo variability than rock music. Finally, even the hypotheses of having periodic beats and downbeats may be invalid, for example, for songs where the players make mistakes or audio tracks containing multiple concatenated songs.

Still, the DBN performs well on most pieces commonly used to train and evaluate beat tracking systems: music with a constant time signature of 3/4 or 4/4 and a stable, medium tempo. This can be seen from the default DBN parameters which most systems use,¹ i.e., tempo range [55, 215] BPM, beats per measure [3, 4], and a tempo variability optimised on pop, rock and dance datasets. Pieces outside these specifications are likely to be mispredicted by the system, but form a minority in typical datasets. Therefore, in terms of evaluation metrics, it usually does not pay to remove the DBN. However, working in these “simplified” conditions blocks research from solving corner cases in existing datasets and targeting more challenging or diverse data.

Our first goal is thus to replace the DBN with minimal postprocessing, free of the aforementioned musical assumptions. A recent attempt to remove the DBN was made by Chen and Su [2]. However, their system may not look appealing to practitioners requiring beat tracking for downstream tasks, or researchers seeking a system to improve, as its accuracy falls clearly behind DBN-based ones.

Our second goal is to provide a powerful basis for practitioners and researchers. The current best-performing system (which uses a DBN) from Hung et al. [10] falls short in this regard, as its code or a pretrained model is not public, its architecture is very complex, and (to the best of our knowledge) the results could not be reproduced by others.

* Equal contribution.



¹ We could verify that [3, 4, 8] use these parameters, since their code is publicly available, and we assume [1, 10] do as well, since they do not mention any details in their paper.

In this paper, we present an open-source system that obtains new state-of-the-art F1 scores without a DBN. It is based on a rotary transformer [11] applied on spectrograms, with the following novelties: (1) We design a frontend alternating convolutions with a transformer variant by Lu et al. [12] that attends alternatively over frequency bins or time frames. (2) We train with a shift-tolerant binary cross-entropy (BCE) that can cope with small deviations in the beat/downbeat annotations, and with weights on beat/downbeat frames to balance their relative scarcity. (3) We propose an approach that encourages downbeat predictions to be a subset of beat predictions, and (4) a data augmentation masking input segments to encourage the network to consider a longer context. Our code, pretrained models, and preprocessed datasets are openly available.²

2. RELATED WORK

The currently best-performing model (on the GTZAN dataset [13] commonly used for evaluation) is by Hung et al. [10] and serves as a point of comparison. It uses a complex neural network architecture named SpecTNT which alternates computing frequency-related features with a frequency-oriented transformer, and processing a virtual extra frequency band with a time-oriented transformer. This runs in parallel with a more widely used Temporal Convolutional Network (TCN, a fully convolutional network with dilated convolutions), and the outputs of the two networks are merged for the final predictions. Unfortunately, the approach is not open source, and to the best of our knowledge, no other research group has managed to reproduce its results. Moreover, it still uses the DBN, which, as argued in the introduction, limits the system’s generality.

Although no other work could reach the accuracy reported by Hung et al., two other recent beat tracking papers brought new interesting ideas [3, 4]. Both perform instrument separation (with a pretrained network) and feed the separate stems (bass, drums, vocals, other, for [3] also piano) into the model, mixing their information in cross-instrument attention blocks. While this approach is very reasonable from a music perception standpoint, it reduces the generality of the system, since it assumes that the input pieces will contain such instruments, at least to some extent. Another proposal of [3, 4] is the use of dilated attention, following the successful use of dilated convolutions in beat-tracking architectures to increase the receptive field without adding computations. We find that flash attention [14] enables us to train with dense attention over a satisfactorily large input size, and leave experiments with dilated attention for future work.

Chen and Su [2] try to remove the DBN and propose a set of improvements. The most impactful is to replace the BCE with the Dice [15] and Focal [16] loss, inspired by their common usage for medical image segmentation. While these losses improve results, possibly due to their inherent ability to handle unbalanced classes, we found that a BCE with weights on the positive (beat and downbeat)

classes outperforms them. We suspect this is because, in contrast to medical image segmentation, our positive examples are single frames, and the Dice and Focal loss perform better when the area of positive predictions is larger [17]. Another proposal by Chen and Su is to predict the phase of the beat/downbeat instead of a single binary value, following [18]. Although this seems promising, the results on both papers (and ours) do not show any consistent improvement.

Many recent approaches [2, 3, 18, 19] use the additional task of tempo prediction (a single tempo target for each input excerpt) in a multi-task setting. While this improves their results, it goes against our goals of generality, since it assumes an (almost) constant tempo in the file excerpt, which is not the case for many kinds of music.

Other recent papers do not align with the goal of this paper: [1] explores the usage of different time resolutions between the input audio and output predictions (only addressing beats); [8] performs unsupervised beat tracking; [20] focuses on online beat tracking; [5] notices the problems of the DBN for music with tempo changes, and proposes a different postprocessing method targeted specifically to classical music; [7] focuses on fine-tuning existing systems, and changing the DBN parameters for targeting specific underrepresented genres.

3. METHOD

Our beat tracker is based on a neural network with ~ 20 M parameters. It starts from 30 seconds of mono audio sampled at 22.05 kHz and converts it to a 128-bin mel spectrogram from 30 Hz to 10 kHz, with a window size of 1024 and hop size of 441 samples (yielding 50 frames per second), and magnitudes scaled via $\ln(1 + 1000x)$ (similar to $\ln(\max(10^{-3}, x))$, but maps silence to zero). These hyperparameters were optimised in preliminary experiments. Our model processes this into frame-wise beat and downbeat probabilities, followed by minimal post-processing to derive beat and downbeat locations.

3.1 Model

Our model (Figure 1) processes a $T \times 128$ spectrogram into $T \times 2$ probabilities; T being the number of input frames (1500 in case of 30 s). It consists of three components: a frontend converting the spectrogram into a sequence of feature vectors, a transformer processing these vectors, and two task heads computing the output probabilities.

3.1.1 Frontend

The frontend’s role is to integrate information across the 128 frequency bands into feature vectors. Typically, this is done via 2d convolutions gradually reducing the number of bands to 1 while increasing the number of channels [3, 10]. We adopt this, but found it helps to interleave convolutions with *Partial Transformers*, which treat the time and frequency axis independently. Overall, our frontend consists of a stem, three identical blocks, and a linear projection.

The stem (Figure 1, top right) starts with a batch normalisation that processes each frequency band separately

²https://github.com/CPJKU/beat_this

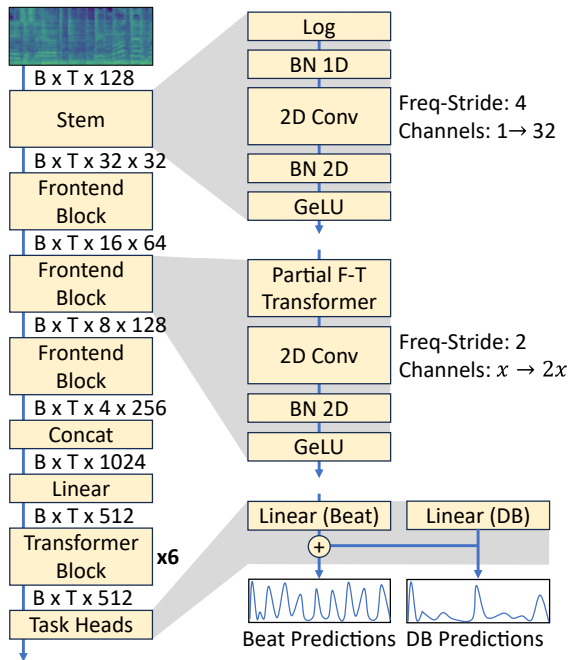


Figure 1: Full model architecture.

to homogenise them, followed by a 2d convolution of 3×4 kernels, regular batch normalisation, and GeLU nonlinearity. The convolution is strided to reduce the number of frequency bands to a fourth and creates 32 channels.

Each block (Figure 1, middle right) consists of two partial transformers, a strided 2d convolution halving the number of bands while doubling the number of channels, batch normalisation and GeLU. The first partial transformer is *frequency-directed*, i.e., it processes the $T \times F \times C$ tensor by treating each time frame as a sequence of length F (the number of bands), the second one is *time directed* and treats each frequency band as a sequence of length T (the number of frames), an idea adopted from the Band Split RoFormer [12]. Each transformer has a head size of 32 (one head in the first frontend block, two in the second, four in the third), rotary positional embedding [11], a sigmoid gate per head [21, Sec. 4.2], and includes a usual pointwise feedforward network with a hidden size of four times the channel count.

After three frontend blocks, the resulting $T \times 4 \times 256$ tensor (4 bands, 256 channels) is reshaped to a $T \times 1024$ tensor and linearly projected to 512 features.

3.1.2 Transformer

The transformer makes up the bulk of our model’s parameters and compute. It consists of 6 stacked transformer blocks processing the 512-dimensional vectors with 16 heads of size 32, rotary positional embedding, sigmoid gating, and a pointwise feedforward network of 2048 hidden units. This matches the configuration in the frontend transformer blocks, but as it processes a single sequence of 512-dimensional feature vectors, it is a regular transformer over time without separately considering a frequency dimension. Its goal is to map the 512 input features to a space that relates to beats and downbeats. Due to the attention

mechanism, the transformer’s receptive field covers the full sequence, and it could therefore produce an output that has characteristics that we want in the beat predictions, for example, regularity.

3.1.3 Task Heads

The output of the final transformer block is processed by two linear layers, one for beats and one for downbeats. Initially, we used the common approach of passing their output into 2 sigmoid functions to produce a probability for each input frame, and then threshold this probability at 0.5 to obtain "hard" beat predictions. However, we observe that this sometimes produces downbeat predictions not coinciding with a beat prediction, which is allowed under the evaluation metrics but is a musically invalid and unusable output. This problem is solved when using a DBN to jointly process beats and downbeats. However, we noticed that several works, e.g., [3,6], use two independent DBNs to predict beats and downbeats (and others [2,10] do not specify). To our surprise, this leads to better metrics, but it severely limits practical use.

To mitigate this problem, we propose a *Sum Head* that sums the output of the beat and downbeat layers, and treats this as the beat logits (for prediction and training). This is a very simple way of helping the network produce a beat when there is a downbeat (though it does not enforce that; a highly negative output of the beat layer can still counter the downbeat layer). We explored other ways of aggregating the beat and downbeat logits, like taking their maximum, but this hampered training due to the sparser gradients. On the GTZAN dataset, the sum head almost halves the percentage of downbeats that are more than 70 ms away from the closest beat, from 1.1% to 0.62%, compared to directly using the output of the linear layers. We observe that the remaining unmatched downbeats are in pieces with very erratic predictions that would be incorrect anyway.

Some systems circumvent the problem by using a 3-way classifier (beat vs. downbeat vs. none) instead of the two binary classifiers (beat vs. none, and downbeat vs. none). However, to be able to train on datasets that do not include downbeat annotations, we stick to binary classifiers.

3.2 Postprocessing

To obtain beat/downbeat locations, we pick all frames assigned the highest beat/downbeat probability within a neighborhood of ± 3 frames (± 70 ms), and probability > 0.5 . In case adjacent frames are assigned the same probability, we report their center. Finally, we move all downbeat predictions to the closest beat prediction to correct the remaining mismatches described in the previous section. For music pieces longer than 30 s, we concatenate predictions over non-overlapping 30-second excerpts.

3.3 Loss

The model is trained by gradient descent on a loss function that compares the frame-wise beat and downbeat predictions with frame-wise binary annotations. The usual loss for binary classification is Binary Cross-Entropy,

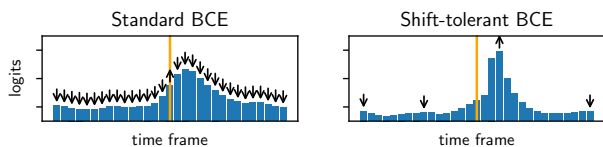


Figure 2: The standard binary cross-entropy loss (left plot) encourages high network outputs (upward arrow) at beat annotations (vertical line), and low outputs for all other frames (downward arrows). Max-pooling the predictions over time redistributes gradients to local maxima (right plot). This way, slightly shifted annotations do not affect learning, and the network produces confident sharp peaks.

$L_{\text{bce}}(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_t y_t \log(\hat{y}_t) + (1 - y_t) \log(1 - \hat{y}_t)$. Training with BCE leads to unconfident predictions since the problem is heavily imbalanced. To counter this, we can weight positive examples by a factor w as $L_{\text{wbce}}(\mathbf{y}, \hat{\mathbf{y}}, w) = -\sum_t w y_t \log(\hat{y}_t) + (1 - y_t) \log(1 - \hat{y}_t)$. We found that setting w to the number of negative examples divided by the number of positive examples (over the training set) yields the best result and is crucial when not using a DBN.

Another problem persists: annotations are not precise down to our spectrogram resolution, due to annotators’ imprecision, players’ asynchronicity, or simply the limits of human perception. This is taken into account during the evaluation, e.g., the typically used F1 score accepts predictions in a ± 70 ms window around labels. During training, the BCE loss punishes close positive predictions (Figure 2, left) even though they may be correct, thus creating two problems: training is slower and the network learns to predict wide “blurred” peaks. This is commonly addressed by adding two extra positive labels around each annotation weighted by 0.5, but this only mitigates the former problem without helping with the latter. Instead, we max-pool predictions over time (7 frames, stride 1) before comparing them to the labels. In this way, only the largest positive prediction ± 3 frames from each label is considered (Figure 2, right). The loss for negative examples is ignored ± 6 frames from each label, as this is how far a max-pooled prediction 3 frames away from a label spreads. Denoting max-pooling of k frames with $m_k(\cdot)$, we can formalise our *Shift-tolerant weighted BCE* as: $L_{\text{st}}(\mathbf{y}, \hat{\mathbf{y}}, w) = -\sum_t w y_t \log(m_7(\hat{\mathbf{y}})_t) + (1 - m_{13}(\mathbf{y})_t) \log(1 - m_7(\hat{\mathbf{y}})_t)$.

3.4 Data Augmentation

Masking. To encourage the model to not only rely on local information for its predictions, we mask 0 to 6 areas of 0.5 to 2 s. Each masked area is randomly divided into 5 to 10 parts which are randomly reordered. This destroys local correspondence between audio and beats without changing local input statistics, and works better than zero masking as employed in SpecAugment [22]. We assume our approach makes it harder for the network to learn a dedicated behaviour for masked areas.

Pitch and time. We speed up and slow down every song by 20, 16, 12, 8, and 4%, and transpose by at most +6 and -5 semitones. We precompute these augmentations (us-

ing Pedalboard [23]) so experiments become reproducible without access to the original audio. To limit storage use, tempo and pitch augmentations are not combined, giving 22 variations for each song. We verified that our limited tempo augmentation gives comparable results to the commonly used approach by Böck and Davies [19] of performing on-the-fly augmentations by randomly changing the hop size of the STFT, at the advantage of not requiring audio access.

4. EXPERIMENTS

We perform 8-fold cross-validation experiments on multiple datasets, compute results on the test-only GTZAN dataset, and do an ablation study.

We use the standard beat-tracking metrics: F1, CMLt, and AMLt and compute them using the `mir_eval` package [24] with default parameters.³ CMLt and AMLt are called continuity metrics, and only consider a beat/downbeat as correct if both it and the previous beat/downbeat are correct; AMLt also accepts different metrical levels such as half or double time, and offbeats [25]. The metrics and their settings match those used by Hung et al. [10]; this enables a comparison with their reported results, though it is unclear which 8-fold dataspit they use, and we cannot run any statistical significance tests since their code is not reproducible. Therefore, any comparison needs to be taken only as an indication.

4.1 Datasets

We train and validate with several datasets: Simac [26], SMC [27], Hainsworth [28], Ballroom [26, 29], HJDB [30], Beatles [31], Harmonix [32], RWC [33, 34] (classical, pop, royalty-free, and jazz), TapCorrect [35], JAAH [36], Filosax [37], ASAP [38], Groove MIDI [39], GuitarSet [40], Candombe [41]. The first two datasets contain only beat annotations, all others both beat and downbeats. We discard one Beatles piece which does not contain downbeat annotations and one with empty beat annotations, resulting in a total of 4556 tracks. For comparison, Hung et al. [10] train with only the first 7 datasets reported above (Simac to Harmonix), plus RWC pop, for a total of 3144 pieces (when assuming the same handling of missing annotations). We use the GTZAN [13] dataset (993 pieces discarding one unannotated track and 6 tracks that miss downbeat annotations) for testing only.

We use only the backing tracks of Filosax without the saxophone solos. For ASAP, we discard the tracks that contain the “rubato” beat annotations. In Groove MIDI, we keep all pieces that are longer than 20 seconds and use the provided audio renderings. We only use the comping tracks of GuitarSet, discarding the solos.

We employ the *8-fold cross-validation splits* published by Böck and Davies [19] for the datasets they used and produce new ones for our added datasets, ensuring different versions of the same piece are not spread across folds, and

³This includes “trim_beats” of 5s that discards the first 5 seconds during the evaluation, which is a choice we do not necessarily approve of, but we use it to be consistent with what seems the standard way of evaluating.

stratifying by metadata when possible. We also produce a new *single split* with $\sim 15\%$ of the pieces on each dataset as validation (again taking care of different versions of the same piece).

4.2 Training

We train for 100 epochs with gradient accumulation over 8 batches of size 8,⁴ AdamW optimizer [42], weight decay of 0.01 (excluding biases and learned norms), learning rate warm-up [43] of 1000 steps to a maximum of 0.0008, and cosine annealing. During each epoch, we randomly sample 30 seconds of each piece, and pad if the total piece duration l is less than 30 seconds. We draw k samples from pieces that are longer than 30 seconds, following the equation $k = \text{round}(\alpha \cdot l/30)$ with $\alpha = 0.65$, since we observe it leads to faster convergence than using one random sample per piece, or $l/30$ non-overlapping samples. On average, this yields ~ 3 samples per piece. During training, every time a sample is drawn, we randomly select a precomputed augmentation described in Section 3.4, and apply masking. The full training takes around 8 hours on a single NVIDIA RTX 2080 Ti, 6 hours on A40, and 4 hours on A100.

During our experimentation, we found that to achieve good results without a DBN, we need our network to be overconfident in its predictions. This may seem to violate usual deep learning practice, but can be explained by a closer look at the beat tracker’s desirable output. We do not want our network to produce probabilities close to 0.5 when unsure, since this will lead to random oscillations between positive and negative predictions, and thus erratic beats. Instead, we want it to give steady, high-probability predictions even when unsure, exactly like the DBN would. To achieve this, we keep training even after the validation loss starts increasing, which would typically indicate overfitting. Indeed, we see that the validation F1 score continues to improve even with increasing validation loss. This means that even with our modifications, the BCE loss is not a good indicator of the F1 score, and further research into alternative losses may be valuable.

The reader may wonder why, once we obtain our well-performing network, we do not use the DBN to increase the metrics even more. By having overconfident predictions, we reduce the benefits of such a postprocessing method. With a degree of simplification, we can imagine the DBN as using a model’s most high-confident predictions to infer beats in low-confidence areas. By avoiding the low-confidence predictions, we are disrupting this mechanism.

4.3 Main Results

We report the results on our 18 datasets in the commonly used 8-fold cross-validation setting: each dataset is split into 8 parts, we jointly train on $18 \cdot 7$ parts (all but one per dataset) and predict on the remaining 18, after 8 such runs we covered all pieces and average metrics over pieces by dataset. We observe that our model outperforms Hung et al. [10], except for Harmonix and RWC Pop (downbeat). In

	Beat F1		Downbeat F1	
	Our	Hung	Our	Hung
ASAP	76.3	-	61.2	-
Ballroom	97.5	96.2	95.3	93.7
Beatles	94.5	94.3	88.8	87.0
Candombe	99.7	-	99.7	-
Filosax	99.5	-	98.5	-
Groove MIDI	93.7	-	82.1	-
GuitarSet	92.0	-	88.1	-
Hainsworth	91.9	87.7	80.0	74.8
Harmonix	95.8	95.3	90.7	90.8
HJDB	98.2	-	96.6	-
JAAH	95.1	-	85.0	-
RWC Classical	77.1	-	66.3	-
RWC Jazz	83.3	-	80.7	-
RWC Pop	96.1	95.0	93.7	94.5
RWC RF	94.5	-	91.9	-
Simac	77.9	-	-	-
SMC	62.7	60.5	-	-
TapCorrect	93.0	-	86.4	-

Table 1: Results with 8-fold cross-validation.

our results, the lowest downbeat performance is obtained in the ASAP and RWC Classical datasets, confirming the well-known difficulty of beat-tracking classical music [2, 5]. Performance on SMC (where only beat annotations are accessible) is also very low, consistent with the outcomes of other systems, highlighting the substantial room for improvement that beat tracking systems continue to hold.

We also report the results on the GTZAN dataset in Table 2. We compute these results with a single model trained on the entirety of our training-val dataset (note that we do not perform any early stopping or other techniques for which the validation dataset may still be necessary). All our runs are computed 3 times with different random seeds, and we report the means and standard deviations of the computed metrics over the 3 seeds. We notice that even when training on the reduced collection of datasets by Hung et al. (third row in the table), we still outperform their F1 score without a DBN, proving the effectiveness of our design choices. Our main model has ~ 20 M parameters, 5 times more than Hung et al. with 4 M, so we also show the results for a smaller model with the hidden dimension of the main transformer blocks reduced from 512 to 128, and the number of heads from 16 to 4. This small model has ~ 2 M parameters and still gives SOTA F1 scores.

Disappointingly, we notice that the continuity metrics (CMLt and AMLt) are lower than those of Hung et al. From qualitative inspections of the results, we notice that for complex or underrepresented pieces, our network introduces non-periodic beats, which drastically lower the continuity metrics. We are then brought to wonder why our network cannot learn a supposedly obvious behaviour, such as only producing periodic-like output, and we can propose two explanations. Firstly, our loss does not specifically penalise non-periodic predictions, but treats each beat individually.

⁴ This enables training with under 8 GiB of GPU memory.

	Beat			Downbeat		
	F1	CMLt	AMLt	F1	CMLt	AMLt
Hung et al. [10]	88.7	81.2	92.0	75.6	71.5	88.1
Our system	89.1 ± 0.3	79.8 ± 0.6	89.8 ± 0.4	78.3 ± 0.4	67.3 ± 0.8	79.1 ± 0.6
– limited to data of [10]	88.9 ± 0.1	79.9 ± 0.4	89.4 ± 0.2	75.5 ± 0.5	60.8 ± 1.2	75.5 ± 0.5
– smaller model	88.8 ± 0.2	79.4 ± 0.4	89.0 ± 0.4	77.2 ± 0.2	65.3 ± 0.3	78.0 ± 0.3
– with DBN	88.1 ± 0.3	80.5 ± 0.4	91.1 ± 0.2	77.4 ± 0.2	73.3 ± 0.2	87.8 ± 0.5

Table 2: Evaluation on the test dataset (GTZAN). The results for Hung et al. [10] are taken from their paper.

	Beat F1	Downbeat F1
Our system	92.6 ± 0.1	85.4 ± 0.1
No sum head	92.6 ± 0.1	85.0 ± 0.1
No tempo augmentation	92.5 ± 0.1	84.9 ± 0.1
No mask augmentation	92.2 ± 0.0	84.5 ± 0.3
No partial transformers	92.2 ± 0.1	83.9 ± 0.2
No shift tolerance	91.2 ± 0.2	82.2 ± 0.4
No pitch augmentation	88.3 ± 0.4	80.8 ± 0.5
No shift tol., no weights	79.5 ± 0.7	68.7 ± 0.8

Table 3: Ablation studies on the single split validation dataset, ordered by decreasing downbeat F1.

This results in a discrepancy between what is preferred by continuity metrics and what the network learns to predict in difficult parts to minimise the loss. Secondly, our datasets contain several non-periodic annotations, some due to quality issues (see Section 4.5), some in correctly annotated pieces such as tapcorrect_10 or beatles_Wild-Honey-Pie, where a 2/4 measure in the middle of a 4/4 piece disrupts the assumption of periodicity for downbeats. Finally, one could also question the generality of the AMLt metric as a tool to quantify double/half-time errors, since the computations of different metrical levels assume that the time signature and the tempo do not change and that a measure can always be divided into 2 or 3 parts.

Using a DBN increases our CMLt downbeat performance by correcting some of the (wrongly) non-periodic outputs, but it reduces our F1 performance, by changing other otherwise correct predictions that fall outside the DBN assumptions. The AMLt score does not increase since our network is overconfident in its predictions, and the DBN cannot easily switch to another metrical level.

4.4 Ablation Studies

We ablate multiple components of our model on the single split described in Section 4.1. We perform every experiment 3 times with different seeds and report the mean and standard deviation on the validation set in Table 3. The usage of our Sum Head shows little impact, but we use it to have a musically valid output, rather than to increase the F1 score. Pitch, mask, and tempo augmentations help, in this order of importance. The usage of partial transformers in our front-end proves more effective than only having convolutions. Our most impactful design choice is the weighted shift-

tolerant loss. Using a normal BCE with positive example weights results in decreased performance, which decreases even further when the weights are removed.

4.5 Notes on Dataset Quality

While exploring the datasets, we found multiple problems in the annotations, and we think this hinders the development of better models, especially for downbeat predictions. Even the GTZAN dataset, which is commonly used for evaluation, is not immune to quality problems. Some of them are evident and not debatable, like jazz_00000, jazz_00002, jazz_00083, blues_00015, reggae_00095, classical_00077, rock_00067. Furthermore, there are pieces where even for experts it would be hard to agree on a unique beat and downbeat annotation, like metal_00081 or classical_00056, and multiple annotations would be necessary. Finally, some pieces question the primary assumption of beat tracking, i.e., that there is a beat/downbeat to track, like pop_00064, and jazz_00003.

5. CONCLUSIONS AND OUTLOOK

In this paper, we presented a new beat tracking system which obtained a state-of-the-art F1 score on a very diverse set of music, with minimal assumptions about the tempo, time signature, and their changes over time. Remarkably, we do not use the DBN postprocessing, which was employed by all recent high-accuracy models. However, removing the DBN hurts the CMLt and AMLt metrics. A study on how this trade-off affects human perception, alternative metrics, and a direct comparison with DBN-based models on complex pieces is left for future work.

We emphasise that beat tracking is not a solved problem, even for commonly targeted genres such as rock or electronic music, especially for the downbeat tracking task. We provide an open-sourced model that can be used as a starting point, and we invite future researchers to improve it. Potential directions are: reducing the model parameters, developing new losses that enforce periodicity during training, using other data augmentation techniques to make the system more robust to multiple sound conditions, fine-tuning it on specific genres, and training on larger datasets. The contribution of people with musical expertise will also be essential, as we think that correcting the existing commonly used datasets, and producing new annotated data for under-represented genres is a crucial step for further development.

6. ACKNOWLEDGEMENTS

This work was supported by the European Research Council (ERC) under the EU’s Horizon 2020 research & innovation programme, grant agreement No. 101019375 (*Whither Music?*), and the Federal State of Upper Austria (LIT AI Lab). The computational results presented were achieved in part using the Vienna Scientific Cluster (VSC).

7. REFERENCES

- [1] T. Cheng and M. Goto, “Transformer-based beat tracking with low-resolution encoder and high-resolution decoder,” in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2023.
- [2] T.-P. Chen and L. Su, “Toward postprocessing-free neural networks for joint beat and downbeat estimation,” in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2022.
- [3] J. Zhao, G. G. Xia, and Y. Wang, “Beat transformer: Demixed beat and downbeat tracking with dilated self-attention,” in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2022.
- [4] T. Kim and J. Nam, “All-in-one metrical and functional structure analysis with neighborhood attentions on demixed audio,” in *Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2023.
- [5] C.-Y. Chiu, M. Müller, M. E. Davies, A. W.-Y. Su, and Y.-H. Yang, “Local periodicity-based beat tracking for expressive classical piano music,” *Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 2922–2934, 2023.
- [6] L. Maia, M. Rocamora, L. W. P. Biscainho, and M. Fuentes, “Adapting meter tracking models to latin american music,” in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2023.
- [7] A. S. Pinto and G. Bernardes, “Bridging the rhythmic gap: A user-centric approach to beat tracking in challenging music signals,” in *Proceedings of the International Symposium on Computer Music Interdisciplinary Research (CMMR)*, 2023.
- [8] D. Desblancs, V. Lostanlen, and R. Hennequin, “Zero-note samba: Self-supervised beat tracking,” *Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 2922–2934, 2023.
- [9] S. Böck, F. Krebs, and G. Widmer, “Joint beat and downbeat tracking with recurrent neural networks,” in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2016.
- [10] Y.-N. Hung, J.-C. Wang, X. Song, W.-T. Lu, and M. Won, “Modeling beats and downbeats with a time-frequency transformer,” in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022.
- [11] J. Su, M. Ahmed, Y. Lu, S. Pan, W. Bo, and Y. Liu, “RoFormer: Enhanced transformer with rotary position embedding,” *Neurocomputing*, vol. 568, 2024.
- [12] W.-T. Lu, J.-C. Wang, Q. Kong, and Y.-N. Hung, “Music source separation with band-split RoPE transformer,” in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024.
- [13] U. Marchand and G. Peeters, “Swing ratio estimation,” in *Digital Audio Effects (Dafx)*, 2015.
- [14] T. Dao, D. Fu, S. Ermon, A. Rudra, and C. Ré, “FlashAttention: Fast and memory-efficient exact attention with IO-awareness,” *Advances in Neural Information Processing Systems*, vol. 35, 2022.
- [15] F. Milletari, N. Navab, and S.-A. Ahmadi, “V-Net: Fully convolutional neural networks for volumetric medical image segmentation,” in *Proceedings of the International Conference on 3D vision (3DV)*, 2016.
- [16] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” in *Proceedings of the International Conference on 3D vision (3DV)*, 2017.
- [17] N. Abraham and N. M. Khan, “A novel focal Tversky loss function with improved attention U-Net for lesion segmentation,” in *Proceedings of the International Symposium on Biomedical Imaging (ISBI)*, 2019.
- [18] T. Oyama, R. Ishizuka, and K. Yoshii, “Phase-aware joint beat and downbeat estimation based on periodicity of metrical structure,” in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2021.
- [19] S. Böck and M. E. Davies, “Deconstruct, analyse, reconstruct: How to improve tempo, beat, and downbeat estimation,” in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2020.
- [20] C.-C. Chang and L. Su, “BEAST: Online joint beat and downbeat tracking based on streaming transformer,” in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024.
- [21] Y. Bondarenko, M. Nagel, and T. Blankevoort, “Quantizable transformers: Removing outliers by helping attention heads do nothing,” *Advances in Neural Information Processing Systems*, vol. 36, 2023.

- [22] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, “SpecAugment: A simple data augmentation method for automatic speech recognition,” in *Proceedings of the Interspeech Conference*, 2019.
- [23] P. Sobot, “Pedalboard,” 2021. [Online]. Available: <https://doi.org/10.5281/zenodo.7817838>
- [24] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, D. P. Ellis, and C. C. Raffel, “mir_eval: A transparent implementation of common MIR metrics,” in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2014.
- [25] M. E. Davies, S. Böck, and M. Fuentes, *Tempo, Beat and Downbeat Estimation*, 2021. [Online]. Available: <https://tempobeatdownbeat.github.io/tutorial/intro.html>
- [26] F. Gouyon, “A computational approach to rhythm description — Audio features for the computation of rhythm periodicity functions and their use in tempo induction and music content processing,” Ph.D. dissertation, Universitat Pompeu Fabra, 2006.
- [27] A. Holzapfel, M. E. Davies, J. R. Zapata, J. L. Oliveira, and F. Gouyon, “Selective sampling for beat tracking evaluation,” *Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 9, pp. 2539–2548, 2012.
- [28] S. W. Hainsworth and M. D. Macleod, “Particle filtering applied to musical tempo tracking,” *EURASIP Journal on Advances in Signal Processing*, vol. 2004, pp. 1–11, 2004.
- [29] F. Krebs, S. Böck, and G. Widmer, “Rhythmic pattern modeling for beat and downbeat tracking in musical audio,” in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2013.
- [30] J. Hockman, M. E. Davies, and I. Fujinaga, “One in the jungle: Downbeat detection in hardcore, jungle, and drum and bass,” in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2012.
- [31] M. E. Davies, N. Degara, and M. D. Plumbley, “Evaluation methods for musical audio beat tracking algorithms,” *Queen Mary University of London, Centre for Digital Music, Tech. Rep. C4DM-TR-09-06*, 2009.
- [32] O. Nieto, M. C. McCallum, M. E. Davies, A. Robertson, A. M. Stark, and E. Egozy, “The Harmonix set: Beats, downbeats, and functional segment annotations of western popular music,” in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2019.
- [33] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, “RWC music database: Popular, classical and jazz music databases,” in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2002.
- [34] M. Goto, “AIST annotation for the RWC music database,” in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2006.
- [35] J. Driedger, H. Schreiber, W. B. de Haas, and M. Müller, “Towards automatically correcting tapped beat annotations for music recordings,” in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2019.
- [36] V. Eremenko, E. Demirel, B. Bozkurt, and X. Serra, “Audio-aligned jazz harmony dataset for automatic chord transcription and corpus-based research,” in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2018.
- [37] D. Foster and S. Dixon, “Filosax: A dataset of annotated jazz saxophone recordings,” in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2021.
- [38] F. Foscarin, A. Mcleod, P. Rigaux, F. Jacquemard, and M. Sakai, “ASAP: a dataset of aligned scores and performances for piano transcription,” in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2020.
- [39] J. Gillick, A. Roberts, J. Engel, D. Eck, and D. Bamman, “Learning to groove with inverse sequence transformations,” in *Proceedings of the International Conference on Machine Learning (ICML)*, 2019.
- [40] Q. Xi, R. M. Bittner, J. Pauwels, X. Ye, and J. P. Bello, “GuitarSet: A dataset for guitar transcription,” in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2018.
- [41] M. Rocamora, L. Jure, B. Marengo, M. Fuentes, F. Lanzaro, and A. Gómez, “An audio-visual database of Candombe performances for computational musicological studies,” in *Congreso Internacional de Ciencia y Tecnología Musical (CICTeM)*, 2015.
- [42] I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” in *International Conference on Learning Representations (ICLR)*, 2019.
- [43] X. S. Huang, F. Perez, J. Ba, and M. Volkovs, “Improving transformer optimization through better initialization,” in *Proceedings of the International Conference on Machine Learning (ICML)*, 2020.

Papers – Session VII

SCORING TIME INTERVALS USING NON-HIERARCHICAL TRANSFORMER FOR AUTOMATIC PIANO TRANSCRIPTION

Yujia Yan, Zhiyao Duan

University of Rochester, Rochester, New York, USA,
yujia.yan@rochester.edu, zhiyao.duan@rochester.edu

ABSTRACT

The neural semi-Markov Conditional Random Field (semi-CRF) framework has demonstrated promise for event-based piano transcription. In this framework, all events (notes or pedals) are represented as closed time intervals tied to specific event types. The neural semi-CRF approach requires an interval scoring matrix that assigns a score for every candidate interval. However, designing an efficient and expressive architecture for scoring intervals is not trivial. This paper introduces a simple method for scoring intervals using scaled inner product operations that resemble how attention scoring is done in transformers. We show theoretically that, due to the special structure from encoding the non-overlapping intervals, under a mild condition, the inner product operations are expressive enough to represent an ideal scoring matrix that can yield the correct transcription result. We then demonstrate that an encoder-only non-hierarchical transformer backbone, operating only on a low-time-resolution feature map, is capable of transcribing piano notes and pedals with high accuracy and time precision. The experiment shows that our approach achieves the new state-of-the-art performance across all subtasks in terms of the F1 measure on the Maestro dataset. **See appendix for post-camera-ready updates.**

1 Introduction

Automatic Music Transcription (AMT) transforms the audio signal of music performances into symbolic representations [1]. In this work, we focus on transcribing piano performance audio into its piano roll representation.¹ The piano roll representation, as formulated in [2], can be abstracted as consisting of sets of non-overlapping time intervals of the form [onset, offset], with each set corresponding to one particular event type, e.g., a specific note or pedal.

Recent strategies to handle the problem of outputting this structured representation fall into three main categories: 1) Keypoint detection and assembly: This approach involves identifying the onsets, offsets, and frame-wise activations of notes and then assembling these elements together with a handcrafted post-processing step. Examples include [3–5];

¹ Code: <https://github.com/Yujia-Yan/Transkun>



© Y. Yan, and Z. Duan. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Y. Yan, and Z. Duan, “Scoring Time Intervals using Non-Hierarchical Transformer for Automatic Piano Transcription”, in *Proc. of the 25th Int. Society for Music Information Retrieval Conf.*, San Francisco, United States, 2024.

2) Structured prediction with a probabilistic model: Models in this category use a probabilistic model to ensure the structure of the output to be sets of non-overlapping intervals, e.g., [2, 6, 7]; 3) Sequence-to-sequence (Seq2Seq) methods²: These methods, such as [8], treat music transcription as a machine translation problem, which translates audio to tokens that encode the target symbolic representation.

Our study focuses on the neural semi-Markov Conditional Random Field (semi-CRF) framework [2] from the second category, which directly models each music event (note or pedal) as a closed time interval associated with a specific event type. The approach employs a neural network to score interval candidates and uses dynamic programming to decode non-overlapping intervals. This framework eliminates the need for separate keypoint detection and assembly steps in the first category but outputs the events (intervals) in a single stage. Compared to other methods in the second category, e.g. [6, 7], it does not need hand-crafted state definitions and state transitions. Additionally, it benefits from optimal decoding in a non-autoregressive fashion as opposed to the slow autoregressive and suboptimal decoding in Seq2Seq methods (the third category).

This paper builds upon, simplifies, and improves the neural semi-CRF framework [2] for piano transcription. Our major contributions are as follows. First, we replace the original scoring module that assigns a score for every possible interval with a simpler and more efficient pairwise inner product operation. Specifically, we prove that due to the special structure of encoding non-overlapping intervals, under a mild condition, the inner product operation is expressive enough to represent an ideal scoring matrix that can yield the correct transcription decoding. Second, inspired by the resemblance between the proposed inner product operation and the attention mechanism in the transformer [9], we use the transformer architecture to produce the interval representation for inner product scoring. We demonstrate that an encoder-only non-hierarchical transformer backbone, operating only on a low-time-resolution feature map, is capable of transcribing notes with high accuracy and time precision. Third, we compare our method against state-of-the-art piano transcription systems on the Maestro v3 dataset, showing that our method establishes the new state of the art across all subtasks in terms of the F1 score.

² Strictly speaking, the Seq2Seq approach can also be categorized as a probabilistic model for structured prediction. We isolate it here for simplifying the discussion.

2 Related Work

2.1 Neural Semi-CRF for Piano Transcription

Previous work of [2] introduced a neural semi-Markov Conditional Random Field (semi-CRF) framework for event-based piano transcription, where each event (note or pedal) is represented as a closed interval associated with a specific event type. The approach employs a neural network to score interval candidates and uses dynamic programming to decode non-overlapping intervals. After interval decoding, interval-based features are used to estimate event attributes, such as *MIDI velocity* and *refined onset/offset positions*³.

The neural semi-CRF can be viewed as a general output layer, similar to a softmax layer, but tailored for handling non-overlapping intervals. For a sequence of T frames, let \mathcal{Y} denote a set of non-overlapping closed intervals. The semi-CRF layer for \mathcal{Y} takes two inputs for each event type:

1. $score(i, j)$: A $T \times T$ triangular matrix that scores every candidate interval $[i, j]$ for inclusion in \mathcal{Y} . The diagonal values $score(i, i)$ represent single-frame events.
2. $score_\epsilon(i - 1, i)$: A $(T - 1)$ -dimensional vector that assigns a score to every interval $[i - 1, i]$ not covered by any interval in \mathcal{Y} , serving as an inactivity score.

Both $score(i, j)$ and $score_\epsilon(i - 1, i)$ are computed using a neural network from the audio input \mathcal{X} . The total score for \mathcal{Y} , given \mathcal{X} , is:

$$\Phi(\mathcal{Y}|\mathcal{X}) = \sum_{[i,j] \in \mathcal{Y}} score(i, j) + \sum_{\substack{[i-1,i] \\ \text{not covered} \\ \text{in } \mathcal{Y}}} score_\epsilon(i - 1, i). \quad (1)$$

For inference, maximum a posteriori (MAP) is used to infer the optimal set of non-overlapping intervals \mathcal{Y}^* :

$$\mathcal{Y}^* = \arg \max_{\mathcal{Y}} \Phi(\mathcal{Y}|\mathcal{X}). \quad (2)$$

For training, the maximum likelihood approach is used, with the conditional log-likelihood defined as:

$$\log p(\mathcal{Y}|\mathcal{X}) = \Phi(\mathcal{Y}|\mathcal{X}) - \log \sum_{\mathcal{Y}'} \exp \Phi(\mathcal{Y}'|\mathcal{X}). \quad (3)$$

Here, $\arg \max$ in Eq. (2), and the summation in the second term in Eq. (3) are over all possible sets of non-overlapping intervals. We refer the readers to [2] for algorithmic details.

To make predictions for all event types (88 keys + pedals), multiple instances of semi-CRF are used in parallel, each corresponding to a specific event type.

2.2 Vision Transformer and YOLOs

The Vision Transformer (ViT) [10] introduced a significant shift in computer vision, offering an alternative to traditional CNN models. ViT processes images as sequences of fixed-size patches using transformer layers [9], proving successful across various tasks. For end-to-end object detection, YOLOs [11] demonstrated a minimal, non-hierarchical encoder-only design that appends [DET] tokens (representing object slots) directly to image patch tokens as input to the transformer encoder. Our architecture adopts a similar encoder-only design for event-based music transcription.

³ For dequantizing onset/offset positions from quantized positions.

3 Revisiting Interval Scoring for Semi-CRFs

The neural semi-CRF framework crucially relies on modeling the interval scoring matrix, $score(i, j)$, which assigns a score to each candidate interval. The size of the matrix, which grows quadratically with the sequence length, poses a challenge to designing an efficient and expressive model architecture. For this discussion, $score_\epsilon$ will be excluded due to its minimal impact on model performance from our observation and negligible modeling challenges.

3.1 Interval Scoring in [2]

In [2], a backbone model first transforms the input sequence $\mathcal{X} = [\mathbf{x}_0, \dots, \mathbf{x}_{T-1}]$ into a sequence of feature vectors $[\mathbf{h}_0, \dots, \mathbf{h}_{T-1}]$. Each interval $[i, j]$ is scored by applying an MLP to features computed from the interval, with the output dimension being the number of event types. For simplicity, assuming only one event type to predict, the score is computed as

$$score(i, j) = MLP([\mathbf{h}_i, \mathbf{h}_j, \mathbf{h}_i \odot \mathbf{h}_j, \mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3]), \quad (4)$$

where \mathbf{h}_i and \mathbf{h}_j are feature vectors corresponding to the interval's onset and offset, \odot denotes element-wise multiplication, and $\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3$ are the first, second, and third statistical moments over the interval $[i, j]$.

After producing the initial interval scoring matrices for all event types, a shallow CNN is applied, treating the interval endpoints as spatial coordinates and event types as channels. This refinement step slightly improves the result.

Directly computing Eq. (4) and the subsequent refinement step are memory intensive. The official implementation processes the scoring matrix in segments and applies gradient checkpointing during training, reducing peak memory usage at the cost of increased computational time. Consequently, the MLP and CNN layers' depth and width are constrained, potentially limiting the model's capacity and increasing susceptibility to local pattern overfitting.

3.2 Interval Scoring with Inner Product

We propose to use the following method for interval scoring:

$$score(i, j) = \frac{|j - i|}{\sqrt{D}} \langle \mathbf{q}_i, \mathbf{k}_j \rangle + b_i \delta(i, j), \quad (5)$$

where $\delta(i, j)$ is the Kronecker delta, which is 1 if $i = j$ and 0 otherwise. $\mathbf{q}_i \in \mathbb{R}^D$, $\mathbf{k}_i \in \mathbb{R}^D$ and $b_i \in \mathbb{R}$ are computed from the embedding vector \mathbf{h}_i using a linear layer f :

$$[\mathbf{q}_i, \mathbf{k}_i, b_i] = f(\mathbf{h}_i). \quad (6)$$

The interval scoring matrix computed from Eq. (5) takes a low-rank plus diagonal structure. This method, termed **Scaled Inner Product Interval Scoring**, computes the score of an event as the scaled inner product between vectors \mathbf{q}_i and \mathbf{k}_j representing the start and the end of the interval.

Despite its simplicity and resemblance to the attention mechanism in transformers, one question arises about the expressiveness of the inner product for capturing the transcription result. We answer this question by constructing a family of interval scoring matrices that can yield the correct decoded result, and then show that this family of matrices can be represented in the form of pairwise inner product under certain conditions.

Without loss of generality, we ignore the intervals of form $[i, i]$, which correspond to the diagonal values in the interval scoring matrix; they can be added back as diagonals as in Eq. (5). Additionally, since only the upper triangular part of the interval scoring matrix is used, we use the notation for a full matrix to simplify the derivation. We begin by defining a set of nonoverlapping closed intervals.

Definition 3.1. Let \mathcal{Y} be a set of closed intervals defined on $\mathbb{N} \cap [0, T - 1]$, i.e., T steps. It is a set of *non-overlapping* intervals if for any two intervals $[i_0, j_0] \in \mathcal{Y}$ and $[i_1, j_1] \in \mathcal{Y}$, $i_0 \geq j_1$ or $i_1 \geq j_0$, and, additionally, $\forall [i, j] \in \mathcal{Y}, i < j$.

Definition 3.2. An ideal interval scoring matrix for \mathcal{Y} over T steps, i.e., $\mathbf{S}_Y \in \mathbb{R}^{T \times T}$, is a matrix such that

$$\begin{aligned} \mathbf{S}_Y(i, j) &> 0, & \forall [i, j] \in \mathcal{Y}, \\ \mathbf{S}_Y(i, j) &= -\epsilon, & \text{otherwise} \end{aligned}$$

where $\epsilon > 0$.

With an ideal scoring matrix \mathbf{S}_Y , it is clear that the MAP decoding will yield \mathcal{Y} , since the exclusion of $\forall [i, j] \in \mathcal{Y}$ or the inclusion of $\forall [i, j] \notin \mathcal{Y}$ will decrease the total score.

Lemma 3.1. *The rank of an ideal interval scoring matrix \mathbf{S}_Y for a set of non-overlapping intervals, \mathcal{Y} , is $M + 1$, where $M = |\mathcal{Y}|$, which is the number of intervals.*

Proof. By definition, the first column is $-\epsilon \mathbf{1}$, that is, $\forall i, \mathbf{S}_Y(i, 0) = -\epsilon$. Subtracting the first column from all columns gives \mathbf{S}'_Y such that

$$\begin{aligned} \mathbf{S}'_Y(i, j) &> \epsilon, & \forall [i, j] \in \mathcal{Y}, \\ \mathbf{S}'_Y(i, j) &= 0, & \text{otherwise} \end{aligned}$$

Given that no two non-zero entries in \mathbf{S}'_Y share a row or column (as per the definition of set of non-overlapping intervals), and there are M non-zero entries, the rank of \mathbf{S}'_Y is M . Since there are at most $T - 1$ non-overlapping intervals across T frames, we have $M \leq T - 1$, and the number of nonzero entries in \mathbf{S}'_Y is smaller than or equal to $T - 1$. As a result, $-\epsilon \mathbf{1}$ (T non-zeros) cannot be represented by a linear combination of other nonzero columns in \mathbf{S}'_Y , therefore $\text{rank}(\mathbf{S}_Y) = \text{rank}(\mathbf{S}'_Y) + 1 = M + 1$. \square

Theorem 3.2. *Let \mathcal{Y} be a set of non-overlapping closed intervals over T steps, with cardinality M . An ideal interval scoring matrix \mathbf{S}_Y can be represented as pairwise inner products between two 1d sequences $(\mathbf{k}_i)_i$ and $(\mathbf{q}_i)_i$ of vectors:*

$$\mathbf{S}_Y(i, j) = \langle \mathbf{q}_i, \mathbf{k}_j \rangle, \quad (7)$$

provided that $\text{rank}(\mathbf{Q}_Y) > M$ and $\text{rank}(\mathbf{K}_Y) > M$ where $\mathbf{Q}_Y = [\mathbf{q}_0, \dots, \mathbf{q}_{T-1}]$, and $\mathbf{K}_Y = [\mathbf{k}_0, \dots, \mathbf{k}_{T-1}]$.

Proof. By Lemma 3.1, the rank of \mathbf{S}_Y is $M + 1$. Then it directly follows the rank factorization of a matrix. \square

Theorem 3.2 establishes a minimum rank requirement for \mathbf{Q}_Y and \mathbf{K}_Y to represent an ideal scoring matrix. This leads to two key observations:

1. The vector dimensions D of \mathbf{k}_i and \mathbf{q}_i must exceed the total number of intervals, $|\mathcal{Y}|$.

2. Consider a linear upsampling operator u_c , which is a special case of a 1-d transposed convolutional layer. It works by dividing each step of a vector sequence into c equal parts when the sequence is upsampled c times. Suppose we want to represent \mathbf{Q}_Y and \mathbf{K}_Y using low-resolution 1-d vector sequences: $\mathbf{Q}'_Y = [\mathbf{q}'_0, \dots, \mathbf{q}'_{T'-1}]$ and $\mathbf{K}'_Y = [\mathbf{k}'_0, \dots, \mathbf{k}'_{T'-1}]$ where $T' < T$, and this representation is achieved by applying u_c to \mathbf{Q}'_Y and \mathbf{K}'_Y , resulting in $\mathbf{Q}_Y = u_c(\mathbf{Q}'_Y)$, and $\mathbf{K}_Y = u_c(\mathbf{K}'_Y)$, where $c = T/T'$ represents the upsampling factor. For this representation to be valid, the vector dimension D' for the low-resolution sequence, i.e., \mathbf{q}'_i and \mathbf{k}'_i should exceed $c|\mathcal{Y}|$.

These observations highlight that the dimensionality requirement depends solely on the count of intervals in \mathcal{Y} and the downsampling (upsampling) factor $c = T/T'$ along the time axis. This analysis reveals sufficient conditions to guarantee the expressiveness of the inner product interval scoring method. From Theorem 3.2, by applying a scaling factor⁴ and reintegrating diagonal terms, we can recover Eq. (5).

3.3 Comparison with Attention Mechanism

Comparing the neural semi-CRF with the inner product scoring to the attention mechanism reveals interesting parallels. Both of them have quadratic time complexity in the length of the input. The original score module, as in [2], resembles an additive attention mechanism, as introduced by [12]. However, attention mechanisms based on inner products [13] have become preferred for their simplicity and computational efficiency. Similarly, the proposed inner product scoring for neural semi-CRFs efficiently scores intervals. However, in contrast to attention mechanisms that score sequence positions and normalize posteriors for each position, neural semi-CRFs score intervals and normalize posteriors globally over sets of non-overlapping intervals.

The Transformer architecture can be viewed as inherently refining a sequential representation for inner product scoring. Inspired by these similarities, we utilize the transformer architecture to produce the 1-d sequence representations $(\mathbf{h}_i^{\text{eventType}})$ for each event type, termed **event tracks**, which will be used for inner product interval scoring.

4 Proposed System

Figure 1 summarizes the proposed system. The input is an oversampled log-mel spectrogram, as in [2]. The spectrogram is downsampled using 2-d strided convolutional layers, followed by the addition of spatial position embeddings (Section 4.2). Event tracks for all event types (notes and pedals) are initialized with their own spatial position embeddings and concatenated with the downsampled spectrogram representations. The concatenated features are processed by a transformer encoder. Subsequently, only the event track embeddings are upsampled using one 1-d transposed convolutional layer. The upsampled event tracks are used for inner product interval scoring (Eq. (5)) to generate

⁴Note that applying a length-dependent scaling on the ideal scoring matrix does not change the decoded result.

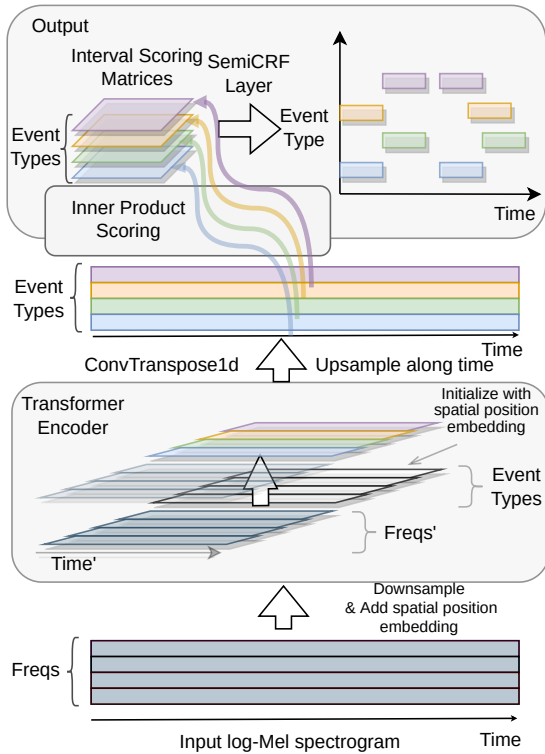


Figure 1: Overview of the proposed system. Inner product scoring follows Eq. (5).

interval scoring matrices, which are then fed to the neural semi-CRF layer for log-likelihood calculation or inference.

4.1 Rethinking Downsampling

Existing studies on Vision Transformers (ViTs) demonstrate the effectiveness of a non-hierarchical design that uses highly downsampled, low-resolution feature maps even for tasks requiring dense predictions, e.g., [14], challenging the dominance of hierarchical models like UNET [15]. However, state-of-the-art (SOTA) piano transcription systems, including [2, 4, 5, 8], retain full resolution along the time axis. These approaches preserve the temporal detail of the input frames, but at the cost of increased training time and reduced model scalability.

This choice might be explained by concerns over losing temporal precision when locating events. However, we argue that the high dimensionality of the embeddings makes the low temporal resolution feature map still capable of processing with enough information.

In our approach, we use strided convolutional layers to downsample the input spectrogram, along both the time and frequency axes, transforming it from its original spatial dimensions (T, F) to a low-resolution feature map with dimensions $(T', F') = (\frac{T}{c_T}, \frac{F}{c_F})$. In line with the ViT literature, we refer to this reduced feature map as *patch embeddings* for $c_T \times c_F$ patches. The choice of patch size (c_T, c_F) may present a trade-off between computational efficiency and the model’s capacity to capture dense events in the input spectrogram. As an initial exploration, we use a patch size of 8×4 to keep the training time within our expected range.

To upsample event tracks to the original temporal res-

olution of frames, we utilize a single transposed 1-d convolutional layer. We found that this simple upsampling layer efficiently prepares representations for inner product scoring at the desired resolution.

4.2 Transformer Encoder Architecture

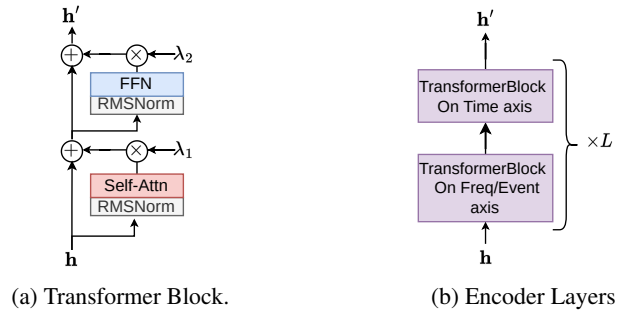


Figure 2: Building Blocks for the Transformer Encoder **Spatial Position Embedding.** We use learnable Fourier features for spatial position embeddings [16] for both time-frequency representations with coordinates $(frameIdx, freqIdx)$, and event tracks with coordinates $(frameIdx, eventTypeIdx)$. This position embedding is chosen for its simplicity and broad compatibility with transformer architectures. Our formula differs slightly from [16] as we follow the formula in the original random Fourier features paper [17]. We compute the position embedding $\mathbf{y} \in \mathbb{R}^E$ from a multidimensional coordinate $\mathbf{x} \in \mathbb{R}^C$ as:

$$\mathbf{y} = g\left(\sqrt{\frac{2}{B}} \cos(\mathbf{W}_r \mathbf{x} + \mathbf{b})\right), \quad (8)$$

where \mathbf{W}_r is a learnable matrix $\mathbb{R}^{B \times C}$, initialized from $\mathcal{N}(0, \gamma^{-2})$; B is the dimension for the Fourier features; γ is a hyperparameter; $\mathbf{b} \in \mathbb{R}^B$ is the learnable bias term, initialized from $\mathcal{U}(-\pi, +\pi)$; $g: \mathbb{R}^B \rightarrow \mathbb{R}^E$ is a two-layer perceptron. This position embedding functions like an MLP that takes coordinates as input, with the first nonlinearity being a scaled cosine function.

The Transformer Encoder Layer. Figure 2a illustrates the basic transformer block. This block first applies *RMSNorm* [18] before the self-attention and feed-forward layers. To enhance training stability, we use ReZero [19] which applies a learnable scaling factor λ , initially set to 0.01, before adding to the skip connection. As in Figure 2b, for reducing computational cost, we alternate attention within each transformer block along the time and frequency/eventType axes; similar ideas are often used for efficient transformer architectures [20–22].

4.3 Segment-Wise Processing

Longer audio is transcribed using segments with 50% overlap. Unlike [2], which discards events that exceed the segment boundary during training, we truncate such events to fit within the segment. We introduce two binary attributes, *hasOnset* and *hasOffset*, to indicate whether an event’s onset or offset has been truncated.

For each event type within a segment, decoding starts from either: (1) the current segment’s boundary, or (2) the offset of the last event in the result set with *hasOffset = true*, whichever is later. Events decoded in the current

segment are then processed as follows: (1) non-overlapping events with $hasOnset = true$ are directly added to the result set; (2) for events overlapping with the last event of the same type in the result set: if the current event has $hasOnset = true$, it replaces the last event⁵; otherwise, the two events are merged.

4.4 Attribute Prediction

Attributes associated with each event include *velocity*, *refined onset/offset positions* (for dequantizing frame positions), and the binary flags *hasOnset* and *hasOffset*. To predict these attributes for an event extracted from the event track $(\mathbf{h}_i^{\text{eventType}})_{i=0}^{T-1}$, e.g., $[a, b]$, we use a two-layer MLP that takes $\mathbf{h}_a^{\text{eventType}}$ and $\mathbf{h}_b^{\text{eventType}}$ as input. The MLP outputs the parameters of the probability distributions for each attribute. Specifically, $velocity \in \{0 \dots, 127\}$ is modeled as a categorical distribution, $refined\ onset/offset\ positions \in (-0.5, 0.5)$ are modeled as continuous Bernoulli distributions [23] shifted by -0.5 , and $hasOnset/hasOffset \in \{0, 1\}$ are modeled as Bernoulli distributions.

5 Experiment

5.1 Dataset

Maestro v3.0.0 [24]. This dataset contains about 200 hours of piano performances, including audio recordings and corresponding MIDI files captured using Yamaha Disklavier pianos. We use the standard train/validation/test splits.

MAPS [25]. The MAPS dataset includes both synthesized and real piano recordings, with the real recordings captured by MIDI playback on Yamaha Disklavier. We evaluate our model on the Disklavier subset (ENSTDkAm/MUS and ENSTDkCl/MUS) of the MAPS dataset, which consists of 60 recordings and is commonly used for cross-dataset evaluation. However, we discovered systematic alignment issues in the ground-truth annotations for both notes and pedals, affecting both onset and offset locations. Onset alignment issues have been previously reported in [26] but are not widely known in the community⁶.

SMD [27]. Similar to Maestro dataset, the SMD dataset was created by recording human performance on a Yamaha Disklavier. We use SMD version 2. The dataset contains 50 recordings. We found that both the onset and offset annotations in SMD are better aligned compared to MAPS.

5.2 Model Specification

The key model specifications are summarized in Table 1. Training takes about 6 days on 2 *NVIDIA RTX 4090*.

Input Mel Spectrogram	sr: 44100 Hz, hop: 1024, window size: 4096, subwindows:5, mels: 229, freq: 30-8000 Hz, segment: 16s,
Patch	shape: 8×4 , embedding size: 256
Strided Conv. Layers for Downsampling	initial proj. size: 64, added with freq. embeddings. out channels: [128, 256, 256, 256], kernel size: 3, strides: [(2,1), (2,2), (2,2), (1,1)]. Each followed by GroupNorm, groups = 4, and GELU (except for the last conv.)
Position Embedding	$\gamma = 1$, $ B = 256$, MLP hidden size 1024
Transformer Encoder	8 heads, 6 layers (=12 blocks), FNN size: 1024
Upsampling	1d. transposed conv, out: 128, kernel size:8, stride:8
Attribute Prediction	two layer MLP, hidden size: 512, dropout 0.1
Batch Size	12
Optimizer	Adabelief [28], maximum learning rate: $4e-4$
Weight Decay	$1e-2$, excluding bias, norm., and pos. embedding
Learning Rate Schedule	500k iterations, 5% warm-up phase, cosine anneal.
Gradient Clipping	Clipping norms at 80% quantile of past 10,000 iterations

Table 1: Model Specification.

5.3 Evaluation Metrics

We compute precision, recall, and f1 score averaged over recordings for both activation level (from [2], equivalent to frame level with infinitesimal hop size), and note level metrics (*Note Onset*, *Note w/Offset*, and *Note w/Offset & Vel.*, using *mir_eval* [29], default settings). All metrics are directly computed from transcribed MIDIs. For details on these metrics, readers can refer to the supplementary material of [2], and the documentation of *mir_eval* [30].

Due to the ground-truth alignment issues discussed in Section 5.1 and space constraints, we only report activation-level and onset-only note-level metrics for MAPS and SMD.

5.4 Results

Our results on the Maestro v3 test set are presented in Table 2. The proposed model achieves state-of-the-art performance across all metrics in terms of f1 score, surpassing previous methods by a significant margin. We also report results for soft pedal transcription which has not been previously explored. The low event-level metrics suggest that accurately determining soft pedal onset and offset times is more challenging than for notes and sustain pedals. We conjecture this is because soft pedals are typically engaged for longer durations and appear significantly less frequently in the dataset than sustain pedals.

Scoring Methods Comparison. We conducted an ablation study to compare our proposed inner product scoring with the more complex scoring method from [2]. We trained a model with an identical architecture but replaced the inner product scoring with the scoring module from [2]. To ensure a fair comparison, we adjusted the hidden sizes of the scoring module to keep the training time for a single iteration within a factor of two of our proposed system. Specifically, all event tracks were projected to a single sequence with a dimension of 512, and the hidden size of the scoring module was set to 512. As shown in Table 2, our inner product scoring outperforms the more complex scoring method, demonstrating its effectiveness and efficiency.

Furthermore, we compared two variants of the inner product scoring: a linear layer and an MLP for computing the $k/q/b$ vectors (f in Eq. (6)). The results demonstrate that the linear layer yields better performance than the MLP.

⁵ For overlapping events between segments: (1) The first event must have $hasOffset = false$. (2) A continuing second event must have $hasOnset = false$. (3) If the second event’s $hasOnset = true$, the first event is replaced by the second event as it’s not supported by the second.

⁶ A piece-dependent onset latency around 15 ms has been previously discussed in [26]. Due to the electro-mechanical playback mechanism, this latency could also be note/pedal dependent. Offset deviation (up to approximately 70 ms) appears more complex and may be influenced by pedal-/note-dependent mechanical latency or undocumented specific piano model’s response to non-binary pedal values.

Method	# Param	Activation			Note Onset			Note w/ Offset			Note w/ Offset & Vel.		
		P(%)	R(%)	F ₁ (%)	P(%)	R(%)	F ₁ (%)	P(%)	R(%)	F ₁ (%)	P(%)	R(%)	F ₁ (%)
Notes													
SemiCRF [2]	9.8M	93.79	88.36	90.75	98.69	93.96	96.11	90.79	86.46	88.42	89.78	85.51	87.44
hFT, reported in [5]	5.5M	92.82	93.66	93.24	99.64	95.44	97.44	92.52	88.69	90.53	91.43	87.67	89.48
hFT [5] ⁸	5.5M	95.37	90.82	92.93	99.62	95.41	97.43	92.22	88.40	90.23	91.21	87.44	89.24
Ours with scoring method in [2]	11.0M	93.79	92.40	93.06	98.61	95.92	97.23	91.69	89.23	90.43	91.08	88.64	89.83
Ours with MLP kqb mapping	13.0M	95.66	94.79	95.20	99.54	96.91	98.19	94.39	91.92	93.12	93.84	91.40	92.59
Ours w/o incomplete events	12.9M	93.76	94.46	95.07	99.56	97.10	98.30	94.66	92.36	93.48	94.12	91.83	92.95
Ours	12.9M	95.75	95.01	95.35	99.53	97.16	98.32	94.61	92.39	93.48	94.07	91.87	92.94
Sustain Pedals													
Kong et al., reported in [4]	20.2M	94.30	94.42	94.25	91.59	92.41	91.86	86.36	87.02	86.58	-	-	-
Kong et al. [4] ^{8 9}	20.2M	94.14	94.29	94.11	77.43	78.19	77.71	73.56	74.21	73.81	-	-	-
SemiCRF [2]	9.8M	95.17	88.33	90.98	82.18	75.81	78.52	78.75	72.74	75.30	-	-	-
Ours w/o incomplete events	12.9M	96.69	92.92	94.47	89.10	83.96	86.28	86.33	81.40	83.63	-	-	-
Ours	12.9M	96.67	94.46	95.40	88.96	84.22	86.37	86.19	81.66	83.71	-	-	-
Soft Pedals													
Ours w/o incomplete events	12.9M	74.41	28.77	36.54	20.24	9.08	11.69	17.19	7.51	9.76	-	-	-
Ours	12.9M	86.42	83.12	84.09	24.32	17.39	19.46	18.51	13.40	15.06	-	-	-

Table 2: Transcription Result on Maestro v3.0.0 Dataset Test Split.

Interestingly, this aligns with how k and q are computed in transformers.

Effect of omitting incomplete events. We found that omitting steps of handling incomplete events at segment boundaries (Section 4.3) only cause noticeable performance impact for pedals, particularly the soft pedal (Table 2). This can be explained by the fact that pedal events, especially soft pedals, can often exceed the segment length, while notes are normally shorter than the segment length we choose.

Results on MAPS/SMD. We evaluated our model on the MAPS dataset using three different ground-truth annotations: (1) Original, (2) Ad hoc Align, where the median deviation from the initial evaluation is subtracted from all notes for each piece and then re-evaluated, and (3) Cogliati, which subtracted a latency value per recording for ENST-DkCL as provided by [26]. For the SMD dataset, only the original annotation is used. Table 3 presents the results.

All methods exhibit low activation-level F1 scores on MAPS. Using the onset-corrected annotation (Cogliati) on MAPS increases the onset F1 score but degrades the activation-level F1 score due to the uncorrected offset biases. In fact, the Cogliati annotation achieves similar or lower activation-level F1 scores compared to all listed methods when evaluated against the original annotation.

All methods achieve F1 scores on SMD that are more comparable to those evaluated on Maestro. However, performance decreases significantly on MAPS, even with corrected annotations. This suggests that the dataset issue may be more complex than a simple piece-dependent timing shift.

Notably, the corrected annotations can lead to different conclusions compared to the original annotation. For example, while the data-augmented Onsets&Frames model achieves a higher note onset F1 score than hFT using the original annotation, it scores lower than hFT when evaluated using the ad hoc correction and the Cogliati annotation.

These observations highlight the need for caution when evaluating models on datasets created using mechanisms that may involve systematic biases, e.g., electromechanical playback. Despite these complications, our proposed system, with or without data augmentation⁷, achieves the highest note onset F1 score among the compared methods on both SMD and MAPS with Ad hoc/Cogliati correction.

⁷ Data augmentation: pitch shifting ± 20 cents, adding noise from [31],

Method	Dataset	Groudtruth	Activation			Note Onset		
			P(%)	R(%)	F ₁ (%)	P(%)	R(%)	F ₁ (%)
Onsets	MAPS	Original	90.27	80.33	84.87	87.40	85.56	86.41
&Frames [24]	MAPS	Ad hoc Align	90.50	80.53	85.08	88.79	86.93	87.78
w. Data Aug. ⁸	MAPS	Cogliati	64.75	82.83	71.60	87.57	84.97	86.19
hFT [5]. ⁸	MAPS	Original	91.53	71.03	79.81	84.63	85.75	85.13
	MAPS	Ad hoc Align	91.77	71.25	80.04	87.32	88.48	87.84
	MAPS	Cogliati	68.83	74.07	70.24	89.94	90.10	89.97
	SMD	Original	93.18	89.82	91.35	98.71	95.58	97.09
Ours	MAPS	Original	88.41	82.29	85.08	84.31	88.10	86.10
	MAPS	Ad hoc Align	88.69	82.57	85.36	86.63	90.53	88.47
	MAPS	Cogliati	65.74	84.69	72.78	89.60	91.39	90.44
	SMD	Original	92.36	95.24	93.73	98.16	97.65	97.89
Ours	MAPS	Original	94.11	84.63	89.00	92.11	88.78	90.38
w. Data Aug.	MAPS	Ad hoc Align	94.35	84.84	89.22	94.21	90.76	92.41
	MAPS	Cogliati	67.77	87.39	75.03	94.66	91.43	92.98
	SMD	Original	93.38	95.91	94.57	99.77	97.68	98.70
Between Ground Truths								
Cogliati [26]	MAPS	Original	98.86	69.22	80.17	100	100	100

Table 3: Transcription Result on MAPS and SMD. See Text for discussion of dataset issues.

6 Conclusion

This paper introduces a simple and efficient method for scoring time intervals using scaled inner product operations for the neural semi-CRF framework for piano transcription. We demonstrate that the proposed scoring method is not only simple and efficient but also theoretically expressive for yielding the correct transcription result. Inspired by the similarity between the proposed scoring method and the attention mechanism, we employ a non-hierarchical, encoder-only transformer backbone to produce event track representations. Our method achieves state-of-the-art performance on the Maestro dataset across all subtasks. Due to resource constraints, we have not evaluated the effect of patch and embedding sizes, which is left for future work. Additionally, future research could explore more advanced transformer architectures, investigate the interaction between transformer architecture and the neural semi-CRF layer, and extend the approach to other instruments and multi-instrument music transcription tasks.

applying randomized 8 band EQ and impulse response from [32].

⁸ Use their provided code and pretrained weights. Recomputed from transcribed MIDIs.

⁹ Previous SOTA for sustain pedals. Their released code indicates a 200 ms onset tolerance for pedal evaluation, contrary to the reported 50 ms in their paper. Here, we use a 50 ms onset tolerance, which explains the large discrepancy between the numbers here and their reported results.

7 Acknowledgement

This work is supported in part by National Science Foundation (NSF) grants 1846184 and 2222129.

8 References

- [1] E. Benetos, S. Dixon, Z. Duan, and S. Ewert, "Automatic music transcription: An overview," *IEEE Signal Processing Magazine*, vol. 36, pp. 20–30, 2019.
- [2] Y. Yan, F. Cwitkowitz, and Z. Duan, "Skipping the frame-level: Event-based piano transcription with neural semi-CRFs," in *Advances in Neural Information Processing Systems*, 2021.
- [3] C. Hawthorne, E. Elsen, J. Song, A. Roberts, I. Simon, C. Raffel, J. Engel, S. Oore, and D. Eck, "Onsets and frames: Dual-objective piano transcription," in *Proceedings of the 19th International Society for Music Information Retrieval Conference (ISMIR)*, 2018.
- [4] Q. Kong, B. Li, X. Song, Y. Wan, and Y. Wang, "High-resolution piano transcription with pedals by regressing onset and offset times," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3707–3717, 2020.
- [5] K. Toyama, T. Akama, Y. Ikemiya, Y. Takida, W. Liao, and Y. Mitsufuji, "Automatic piano transcription with hierarchical frequency-time transformer," in *International Society for Music Information Retrieval Conference*, 2023.
- [6] R. Kelz, S. Böck, and G. Widmer, "Deep polyphonic adsr piano note transcription," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 246–250.
- [7] T. Kwon, D. Jeong, and J. Nam, "Polyphonic piano transcription using autoregressive multi-state note model," in *Proceedings of the 19th International Society for Music Information Retrieval Conference*, 2018.
- [8] C. Hawthorne, I. Simon, R. Swavely, E. Manilow, and J. Engel, "Sequence-to-sequence piano transcription with transformers," in *International Society for Music Information Retrieval Conference*, 2021.
- [9] A. Vaswani, N. M. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Neural Information Processing Systems*, 2017.
- [10] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2021.
- [11] Y. Fang, B. Liao, X. Wang, J. Fang, J. Qi, R. Wu, J. Niu, and W. Liu, "You only look at one sequence: Rethinking transformer in vision through object detection," in *Neural Information Processing Systems*, 2021.
- [12] D. Bahdanau, K. H. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *3rd International Conference on Learning Representations, ICLR 2015*, 2015.
- [13] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 1412–1421.
- [14] Y. Li, H. Mao, R. Girshick, and K. He, "Exploring plain vision transformer backbones for object detection," in *European conference on computer vision*, 2022, pp. 280–296.
- [15] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proceedings of the 18th International Conference on Medical Image Computing and Computer-assisted Intervention (MICCAI)*, 2015, pp. 234–241.
- [16] Y. Li, S. Si, G. Li, C.-J. Hsieh, and S. Bengio, "Learnable fourier features for multi-dimensional spatial positional encoding," in *Advances in Neural Information Processing Systems*, 2021.
- [17] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *Neural Information Processing Systems*, 2007.
- [18] B. Zhang and R. Sennrich, "Root Mean Square Layer Normalization," in *Advances in Neural Information Processing Systems*, 2019.
- [19] T. Bachlechner, B. P. Majumder, H. Mao, G. Cottrell, and J. McAuley, "Rezero is all you need: fast convergence at large depth," in *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, vol. 161, 2021, pp. 1352–1361.
- [20] J. Ho, N. Kalchbrenner, D. Weissenborn, and T. Salimans, "Axial attention in multidimensional transformers," *ArXiv*, vol. abs/1912.12180, 2019.
- [21] N.-C. Ristea, R. T. Ionescu, and F. S. Khan, "Septr: Separable transformer for audio spectrogram processing," in *Proceedings of INTERSPEECH*, 2022, pp. 4103–4107.
- [22] W.-T. Lu, J.-C. Wang, Q. Kong, and Y.-N. Hung, "Music source separation with band-split rope transformer," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 481–485.
- [23] G. Loaiza-Ganem and J. P. Cunningham, "The continuous bernoulli: fixing a pervasive error in variational autoencoders," in *Advances in Neural Information Processing Systems*, 2019.

- [24] C. Hawthorne, A. Stasyuk, A. Roberts, I. Simon, C.-Z. A. Huang, S. Dieleman, E. Elsen, J. Engel, and D. Eck, “Enabling factorized piano music modeling and generation with the MAESTRO dataset,” in *International Conference on Learning Representations*, 2019.
- [25] V. Emiya, N. Bertin, B. David, and R. Badeau, “Maps - a piano database for multipitch estimation and automatic transcription of music,” 2010.
- [26] A. Cogliati, Z. Duan, and B. Wohlberg, “Context-dependent piano music transcription with convolutional sparse coding,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 12, pp. 2218–2230, 2016.
- [27] M. Müller, V. Konz, W. Bogler, and V. Arifi-Müller, “Saarland music data (SMD),” in *Late-Breaking and Demo Session of the 12th International Conference on Music Information Retrieval (ISMIR)*, 2011.
- [28] J. Zhuang, T. Tang, Y. Ding, S. Tatikonda, N. Dvornek, X. Papademetris, and J. Duncan, “Adabelief optimizer: Adapting stepsizes by the belief in observed gradients,” *Conference on Neural Information Processing Systems*, 2020.
- [29] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, and D. P. W. Ellis, “Mir_eval: A transparent implementation of common mir metrics,” in *International Society for Music Information Retrieval Conference*, 2014.
- [30] C. Raffel, “mir_eval documentation on transcription metrics,” https://craffel.github.io/mir_eval/#id46, Accessed: 10-April-2024.
- [31] K. J. Piczak, “ESC: Dataset for Environmental Sound Classification,” in *Proceedings of the 23rd Annual ACM Conference on Multimedia*. ACM Press, 2015, pp. 1015–1018.
- [32] “Echo thief,” <http://www.echothief.com/>, accessed: 2023-05-07.

PERTOK: EXPRESSIVE ENCODING AND MODELING OF SYMBOLIC MUSICAL IDEAS AND VARIATIONS

Julian Lenz
Lemonaide Research
Barcelona, ES

Anirudh Mani
Lemonaide Research
Boston, US

ABSTRACT

We introduce *Cadenza*, a new multi-stage generative framework for predicting expressive variations of symbolic musical ideas as well as unconditional generations. To accomplish this we propose a novel MIDI encoding method, *PerTok* (Performance Tokenizer) that captures minute expressive details whilst reducing sequence length up to 59% and vocabulary size up to 95% for polyphonic, monophonic and rhythmic tasks. The proposed framework comprises of two sequential stages: 1) *Composer* and 2) *Performer*. The Composer model is a transformer-based Variational Autoencoder (VAE), with Rotary Positional Embeddings (RoPE) [1] and an autoregressive decoder modified to more effectively integrate the latent codes of the input musical idea. The Performer model is a bidirectional transformer encoder that is separately trained to predict velocities and microtimings on MIDI sequences. Objective and human evaluations demonstrate *Cadenza*'s versatile capability in 1) matching other unconditional state-of-the-art symbolic models in musical quality whilst sounding more expressive, and 2) composing new, expressive ideas that are both stylistically related to the input whilst providing novel ideas to the user. Our framework is designed, researched and implemented with the objective of ethically providing inspiration for musicians.

1. INTRODUCTION

The creative endeavor in present-day music production is inherently complex and multifaceted. However, it can be broadly categorized into distinct phases that include 1) initiation, 2) evolution and development, and 3) completion of musical ideas into a finished musical outcome. Modern generative models have had a major impact in every creative domain, none the least in music creation. MIDI, and therefore Symbolic AI research approaches for single-track MIDI generation are especially applicable to the contemporary music producer. The motivation behind our investigation arises from a gap in this current landscape to facilitate the crucial middle phase of the creative pro-

cess: absence of a comprehensive, adaptable modeling framework specifically engineered for generating expression variations from a given MIDI file input. Our proposed solution, *Cadenza*, addresses this by focusing on the 'development' phase of music creation while unveiling a framework that is designed for flexibility and efficiency. *Cadenza* utilises a multi-stage generative process, the *composer* and the *performer*, to create novel ideas and variations while emulating the nuanced performance characteristics that can define a given musical style. We choose to call our framework 'Cadenza', inspired by the improvised musical passage played by soloists, creating new and exciting variations of the original motifs of the piece being performed.

1.1 Encoding Symbolic Music

Alongside transformer-based architectures a number of methods have been proposed to encode, or *tokenize* MIDI files into a discrete sequence of tokens. As transformers suffer from quadratic memory complexity in relation to sequence lengths [2], particular focus is placed on capturing relevant MIDI information whilst minimizing the total number of tokens. Popular tokenizers include REMI [3], TSD [4] and Structured [5], amongst many others. However, these approaches suffer from a common drawback: they rely on singular tokens to denote the position of each note event on an evenly-spaced temporal grid. In comparison to MIDI files, which typically utilise a time resolution of 220 or 440 *ticks-per-quarter* [note], these tokenizers are typically employed with just four intervals per quarter note. This has two negative effects: first, note-values outside this range are immediately *quantized*, such as quarter-triplets, eighth-triplets, quintuplets, and thirty-second notes. In addition, any rhythmic performance attributes, expressed as subtle deviations from the fixed-grid, are immediately lost. As a result, the current state-of-the-art in MIDI tokenizers are unable to accurately capture the full range of rhythmic values and expressive performances.

1.2 Expressive Modeling

In both digital and physical contexts, it is common to divide the act of *composing* music and *performing* it. The composition (or score) contains the raw musical idea, whereas the performance will typically embellish it with additional details, such as varying volumes (velocities in MIDI) and subtle timing deviations. Symbolic datasets can



be categorized broadly as:

- **Score:** The sequences contain quantized rhythmic values and minimal/no volume information.
- **Time-Performance:** Dynamics and expressive timing are captured. The performer(s) play without a fixed tempo, resulting in a *time*-based encoding (typically milliseconds), such as [6].
- **Beat-Performance:** Dynamics and expressive timing are captured. The notes are recorded in relation to a fixed tempo, with rhythmic expressivity occurring as deviations from the quantized beats.

Although a substantial quantity of **score** datasets exist, there are significantly fewer in both **performance** categories. As a result, systems such as that proposed in [7], wherein both composition and performance elements are jointly trained and predicted, are limited by this inequality.

A number of recent models have been proposed to exclusively add performance elements, such as RenderingRNN [8] and ScorePerformer [9]. However, they rely on the prediction of *tempo* tokens in alignment with the **Time-Performance** standard. This results in MIDI files that are still rhythmically *quantized*, albeit with varying tempos. We posit that this approach is incompatible with the common production standards of many modern genres that instead rely on fixed tempos.

The framework in Compose & Embellish [10] proposed a system of jointly training Lead-Sheet (score) and Performance models. With a modified REMI [3] tokenization they demonstrated that the lead-sheet model could be pre-trained on a greater quantity of **score** data, and subsequently fine-tuned with the performer, on a smaller **performance** dataset. Similar to prior systems, they quantize rhythms to the nearest 16th position, and instead predict [Tempo] for expressive timing. Furthermore, due to the joint-conditioning training method, the compose model can lose certain capabilities from the fine-tuning process as it fits to the smaller performance dataset.

1.3 Generating Variations

A number of models have been proposed to solve variations-adjacent tasks. ThemeTransformer [11] utilises contrastive representation learning in a sequence-to-sequence framework to generate a melody and accompaniment that recurrently incorporates the original theme. The authors of Music FaderNets [12] instead propose a *style-transfer* task, wherein a number of high-level attributes can be applied to transform a polyphonic sequence. Our work most notably builds off of the model proposed in MuseMorphose [13], which uses a novel *in-attention* mechanism in a transformer-based VAE for generating attribute-controlled variations on symbolic data. However, these models are designed to predict long-form sequences (16+ bars) that do not contain any expressive information.

Overall, our key contributions to the field through this work are two-fold. Firstly, in section 2.2 we introduce

PerTok, a novel MIDI encoding method that captures expressive details with required granularity while maintaining compact sequence lengths and manageable vocabulary sizes. PerTok is implemented with the MidiTok [4] library, released open source and is compatible with any token-based sequential generation model. Secondly, the *Cadenza* framework itself represents a significant leap forward as presented in section 3, integrating the 'Composer' and 'Performer' models into a cohesive architecture that is researched and designed for the domain of AI-assisted music creation. Our framework offers a natural and intuitive way for musicians to create and modify music, which can be tailored to specific stylistic goals. Human evaluations showcase how Cadenza matches other state-of-the-art MIDI models in unconditional score generation quality, creates dynamic variations on input ideas, and sets a new standard for human-like expressive articulations.

2. SYMBOLIC DATA ENCODING

We aim to encode MIDI data in a manner that is both 1) aligned with common audio production use-cases and 2) efficient in the context of transformer-based generative models. While a number of state-of-the-art models such as Anticipatory Music Transformer (AMT) [14], Figaro [15] and Multi-Track Music Machine (MMM) [16] have focused on long-form, multi-track generation, we have observed from our experiences in designing products for contemporary music producers that they more commonly interact with short, single-track files. Furthermore, we observe that a number of tokenization methods such as [9] rely on *tempo* tokens to create a sense of expressive performance, whereas music producers often keep a singular, consistent tempo throughout their composition. Thus, our proposed encoding method is focused on single-track MIDI files in which the expressivity is calculated in relation to a fixed tempo.

2.1 Score & Performance Encoding

To address this, we create separate tokens to model the composition and performance timing elements. More specifically, the macro *timeshift* tokens represent the quantized note locations within a score. Separately, the *microshift* events denote a small adjustment from the quantized location, similar to the subtle timing deviations used by human performers. By separating these events, we are able to maintain reasonable balance between vocabulary size and total sequence length. Furthermore, the differentiation allows for models to be separately trained on the composition and performance tasks respectively. As there is a significant difference in the availability between quantized and performed symbolic datasets, this enables us to feed a far greater quantity of data into the composition-only model.

Initial tests revealed that the common convention of quantizing the composition-level timeshift tokens to 16th notes was leading to a number of musical issues. For example, when quarter- or eighth-note triplets were present

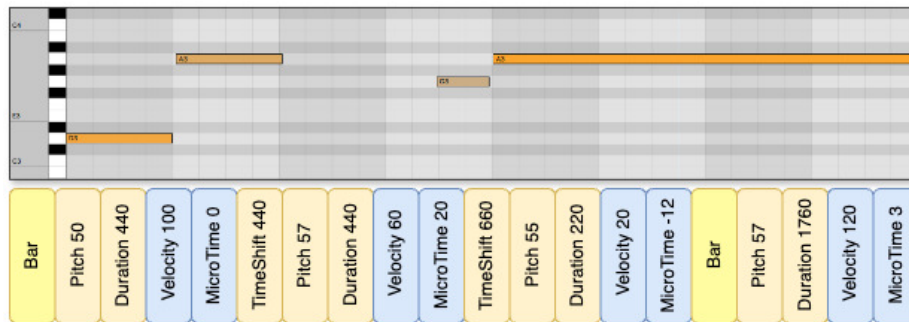


Figure 1. Example of PerTok tokenization on a 2-bar excerpt of a MIDI file. Composition tokens are highlighted in yellow, and performance tokens in blue.

in the input MIDI file, the quantization process was considering them to be 16th notes with large degrees of *microshift*. We addressed this by providing the ability to specify multiple, overlapping quantization grids, such as 16th, quarter-triplets, and 8th-triplets. Thus, the PerTok tokenizer is more adept at capturing the wide variety of rhythmic values that are commonly found in genres such as hip-hop, jazz and salsa (among many others).

2.2 PerTok

Score Tokens: Similar to the MIDI-Like [7] and Structured [5] encoding methods we represent *macro* time changes between notes with **Timeshift** tokens. As MIDI time data is typically expressed as ticks-per-quarter, PerTok allows for multiple overlapping granularities to model a variety of rhythmic values. When encoding the MIDI data, PerTok matches each note’s position to the closest possible timeshift value. **Pitch** is denoted as a MIDI pitch value between 0-127, with the capability to limit this range when musically appropriate. **Duration** tokens are used after each new note, to indicate the length of time before a MIDI note-off message is triggered. Notably, PerTok allows for the removal of duration tokens altogether, which helps further reduce sequence lengths when modeling rhythmic instruments.

Performance Tokens: With the addition of performance tokens, we aim to capture the musical subtleties that transform a written score into an expressive performance. **Velocity** tokens denote the strength of the note’s attack, a property that is typically used in DAWs to augment timbre and volume characteristics. Although MIDI provides a range of 0 - 127 for velocity values, we allow for a bucketing approach to reduce a given velocity into one of n possible values. **Microshift** tokens provide a granular shift from the quantized rhythmic note value. PerTok is provided a maximum microshift value (e.g. 30 MIDI ticks) and a discrete number of possible microshift buckets. For example, *Microshift 15* represents a placement of 15 ticks after the quantized note position, and *Microshift 0* results in the initial quantized value.

In **Table 1** we provide a benchmark of our proposed PerTok encoding against a number of popular MIDI tokenizers. We sampled from 2,000 polyphonic 4- and 8-bar

Tokenizer	Vocab. Size	Seq. Length
REMI	273	195
REMI-p	5505	199
Structured	289	216
Structured-p	7265	216
TSD	288	188
TSD-p	7264	192
PerTok	196	134
PerTok-p	259	243
PerTok no-duration	164	80

Table 1. Vocabulary sizes and average sequence lengths for popular tokenizers and our proposed PerTok encoding.

MIDI files that are used in modern audio production environments. For each tokenizer, we use 32 possible velocity buckets. To demonstrate the tradeoff between composition and performance, we initialize one version with 16th-note quantization (thus removing any performance characteristics), and a second (denoted with a -p) version with 440 timeshifts per quarter note. REMI and Structured methodologies use evenly spaced temporal grids to encode the location of each event, whereas PerTok uses a mixture of macro and micro timeshift tokens leading to a 95% reduction in vocabulary size when encoding expressive rhythmic details. We additionally provide a visualization of our encoding method with a sample 2-bar melody in **Figure 1**.

3. MODEL

With the objective of generating expressive, beat-structured variations on an input MIDI file, we present *Cadenza*, a multi-stage VAE with transformer-based components. The framework is designed upon a principle that the *composition* task requires a significant quantity of data and benefits from auto-regressive generation, whereas the patterns of expressive *performance* tokens can be learned with smaller datasets and predicted in a bi-directional manner.

3.1 The Composer

Given an input sequence of tokens $\{x_1, x_2, \dots, x_t\}$ in which x_t represents a single token x at index t the **composer** model is designed to auto-regressively predict $\{y_1, y_2, \dots, y_t\}$, an output sequence that is musically related yet distinct from the input. We utilise a sequence-to-

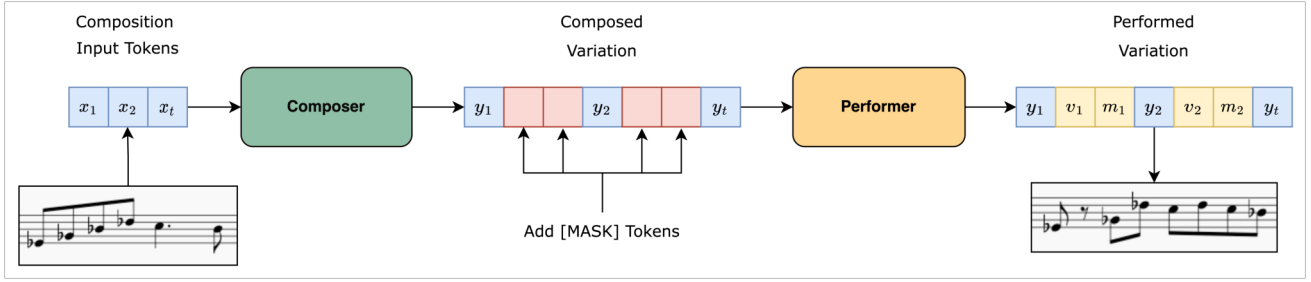


Figure 2. An overview of the multi-stage *Cadenza* architecture.

sequence VAE architecture with in-attention conditioning similar to MuseMorphose [13], enabling the model to learn a compressed, latent representation of short musical ideas within a regularized space. Within each attention mechanism, the query q and key k vectors at timesteps m, n are obtained with RoPE [1] for enhanced positional context:

$$q_m^\top k_n = (R_{\Theta, m}^d W^q x_m)^\top (R_{\Theta, n}^d W^k x_n) \quad (1)$$

Wherein $R_{\Theta, m}^d$ and $R_{\Theta, n}^d$ are the rotary matrices for positions for embedding positional information, W^q and W^k are learnable weight matrices transforming inputs x_m and x_n into the query and key vectors. For additional context we encourage readers to refer the original paper. As music is a deeply temporal phenomenon, the composer benefits from the increased token spatial modelling that is provided by the rotary embeddings.

The encoder is designed to create a latent vector z of the input musical idea which serves as an anchor throughout the decoding process. In alignment with the original transformer [2] we first project the input sequence with a learned embedding space, transforming it into $X \in \mathbb{R}^{d \times t}$ where d is the hidden dimension size. This is then processed through several multi-head self-attention layers. Following a similar approach to [13] we extract the first timestep of the final attention layer output to obtain hidden vector $h \in \mathbb{R}^d$, a contextual representation of the full input sequence.

Following standard VAE methodology [17], the output vector is then processed through two learnable weight matrices $W_\mu \in \mathbb{R}^{d \times d_z}$ and $W_\sigma \in \mathbb{R}^{d \times d_z}$, wherein d_z denotes the size of the latent dimension. This process yields the mean μ and standard deviation σ vectors, encapsulating the latent space distribution parameters. Using the reparameterization trick, we sample ϵ from the Gaussian distribution to obtain $z \in \mathbb{R}^{d_z}$ from our encoder:

$$z = hW_\mu + hW_\sigma \odot \epsilon \quad (2)$$

The encoder’s output distribution $q(z|X)$ is aligned to a Gaussian prior $\mathcal{N}(0, 1)$ by the traditional Kullback-Leibler (KL) divergence loss term:

$$D_{KL}(q(z|X)||p(z)) = -\frac{1}{2} \sum_{k=1}^{d_z} (1 + \log(\sigma_k^2) - \mu_k^2 - \sigma_k^2) \quad (3)$$

We further modify the equation utilising free bits as proposed by [18], allowing the encoder a degree of unpenalized space defined by λ to learn musical attributes without regularization.

$$\mathcal{L}_{KL} = \sum_{k=1}^{d_z} \max(\lambda, D_{KL}(q(z_k|X)||p(z_k))) \quad (4)$$

The decoder is trained to autoregressively predict an output sequence of tokens whilst maintaining a recognizable connection with the input musical idea. Initially, the latent vector z is expanded to the decoder’s hidden dimension d via a learnable matrix $W_{\text{pre}} \in \mathbb{R}^{d_z \times d}$. We separately expand the input tokens v_1, v_2, \dots, v_k with the same embedding layer used by the encoder.

In the context of the autoregressive VAEs it has been noted that *posterior collapse* is a common issue [19–21], in which a sufficiently powerful decoder can simply ignore the encoder’s regularized information, instead relying purely on the previous tokens. We utilise the methodology of Skip-VAE [22] and MuseMorphose [13], integrating the proposed *in-attention* mechanism. Prior to each attention layer, we sum expanded latent vector z_{pre} with every timestep of the previous hidden state, thus reinforcing its information throughout every stage of the decoding process. Therefor hidden state $H \in \mathbb{R}^{t \times d}$ at layer i is calculated as:

$$H_i = \text{SelfAttention}(H_{i-1} + x_{\text{pre}}) \quad (5)$$

The final hidden state is then passed through a feed-forward layer, which has weights tied to the embedding layers as first proposed in [23]. The decoder minimizes the negative log likelihood (NLL) of the output sequence y_t when given prior tokens:

$$\mathcal{L}_{\text{recon}} = \sum_{t=1}^T \log p_\theta(y_t | y_{<t}, z) \quad (6)$$

Thus, the composer is optimized with the NLL reconstruction loss as well as the β -scaled regularization KL loss:

$$\mathcal{L}_{\text{composer}} = \mathcal{L}_{\text{recon}} + \beta \mathcal{L}_{KL} \quad (7)$$

3.2 The Performer

The Performer is separately tasked with computing tokens for **velocity** and **microshift** time values. Thus, it is able to transform a quantized MIDI score into one with expressive characteristics, rendering it more suitable for a variety of music production tasks. Whereas the composition task generally benefits from an autoregressive setup wherein each token is predicted sequentially, it has been demonstrated in [24, 25] that performance attributes can be predicted in a bi-directional manner.

We utilise a framework comparable to the masked token prediction task of BERT [26]. During training and inference, we replace the input performance tokens denoting velocity and microtiming with a single [MASK] token. The tokens related to pitch, timeshift, and duration are left unmodified. The model is a standard transformer encoder as per [2], with sinusoidal positional embeddings, layer normalization and a final feedforward layer that has weights tied to the initial embedding layer.

During training the model is tasked to replace each [MASK] token with an appropriate velocity and microshift value, with cross-entropy loss used exclusively on the masked tokens. We perform the masking operation on 100% of performance tokens. At inference time, we manually mix tokens between the original source and model predictions, thus ensuring the original pitch, timeshift and duration values are maintained.

4. EXPERIMENTAL SETUP

4.1 Composer Ablations

By training several composer models, we aim to understand the relationship between various degrees of KL regularization and the decoded sequence’s similarity to the input. Each model is trained on 4-bar segments of the full Lakh-MIDI dataset [27]. The models all have 12 layers and 8 heads in both the encoder and decoder, with a latent dimensionality d_z of 128 and hidden dimension d of 512. The *Full-KL* model was trained with a KL regularizer $\beta = 1.0$ and free bit $\lambda = 0.15$. The *Balanced-KL* model was trained with $\beta = 0.3$ and $\lambda = 0.25$. In both instances the KL regularization was applied with cosine cyclical annealing [28] every 10,000 steps. We initially keep $\beta = 0.0$ for the first 25,000 steps, and then linearly raise it to the maximum value over the proceeding 25,000 steps. Finally, the *No-KL* model had a $\beta = 0.0$ (no regularization), thus allowing the encoder to exclusively optimize against reconstruction quality.

Objective Evaluations : We generate a single variation for 500 files from the test set for each model, utilising greedy decoding to remove any sampling logic from the evaluation framework. For each sample, we calculate the similarity in **pitch distribution**, **onset locations** and **durations**:

$$\text{similarity}(x^a, x^b) = 100 \frac{\langle x^a, x^b \rangle}{\|x^a\| \|x^b\|} \quad (8)$$

wherein $x \in \mathbb{Z}^t$ is a discrete vector of t attributes; in the case of **pitch** we set $t = 128$ to capture the full MIDI note range, and for both **onset location** and **duration** $t = 64$, representing the nearest 16th value in a 4-bar pattern. Finally, we report **Absolute Similarity**, the percentage of notes that have *identical* characteristics (pitch, onset, duration) between both sequences.

4.2 Performer Fidelity

Two performer models are trained with separate datasets to measure their capacity to model the unique expressive characteristics of a given training set. Both models are trained with 12 layers and heads, a hidden dimensionality of 768, and a dropout of 10%. One model is trained on the classical MusicNet dataset [29], and the other (referred to as *HipHop*) is trained on a proprietary hip-hop dataset. In both cases, we train on approximately 10,000 4-bar excerpts. Each dataset contains polyphonic data with differing expressive patterns of velocities and microtiming.

We randomly extract 2,000 polyphonic 4-bar patterns from the Lakh-MIDI dataset [27] and generate expressive tokens from both models. We subsequently measure the velocity and microtiming distributions of both the generations as well as the two original training datasets. **Velocity** distribution is represented as $v \in \mathbb{Z}^{128}$, a vector representing the number of occurrences of each velocity value. For each note, **microtiming** is calculated as a percentage deviation from the nearest 16th note, with +/-50% denoting the halfway point to/from the adjacent 16th time index. The total distribution of **microtiming** deviations in a given sequence is thus represented with vector $mt \in \mathbb{Z}^{100}$.

For both velocity and microtiming, we compare the distributions of both model’s predictions against the MusicNet and HipHop datasets. In **Table 4** we report the KL divergence, as well as the absolute difference in the mean and standard deviation for these distributions. In each metric, a lower value indicates higher degree of similarity between the model’s predictions and the original dataset’s expressive characteristics.

4.3 User Study

We conduct a thorough user study to achieve a qualitative understanding of our model’s performance, comparing to different external baselines and versions due to hyperparameter settings. All the audio samples that users heard were 4 bar MIDI files voiced through the same Piano VST. In Part A of the user study, 25 human evaluators listened to 5 seed melodies and 4 alternative variations for each of them, coming from the *No-KL*, *Balanced-KL* and *Full-KL* versions of our proposed model, and additionally a *Placebo* melody which was randomly selected to be in the same key and scale as the input but had no relation to it. This was done to confidently ground the musical understanding of our human evaluators. Overall, 44% of our evaluators identified themselves as "Novice : I have little to no experience making music", 32% as "Amateur : I love making music for fun", and 24% as "Professional : I regularly make music in a professional capacity".

Model	Pitch Sim.(%)	Onset Sim.(%)	Duration Sim.(%)	Absolute Sim.(%)	Human Eval(1-5)
Full-KL	71.01	77.99	88.65	9.44	1.94
No-KL	95.64	83.04	98.40	20.05	2.84
Balanced-KL	92.06	80.80	96.95	16.46	2.71
Placebo	-	-	-	-	1.22

Table 2. Objective and human-evaluation results from the ablation studies. Higher values indicate more similarity to the input’s musical characteristics.

In Part B, the same 25 human evaluators also rated 3 unconditional generations from Cadenza, Anticipatory Music Transformer (AMT) [14]¹ and Figaro [15] models, which broadly represent the state-of-the-art in symbolic polyphonic generation. We randomly sampled 3 generations from publicly-available checkpoints of each model, all of which were trained on identical versions of the Lakh MIDI dataset [27], and present the results in **Table 3**. As Cadenza’s composer requires a latent vector, we randomly sample from a Gaussian distribution for unconditional generations. For each sample, the evaluator was asked to rate between how musically appealing it sounded to them with 1 being the lowest and 4 being the highest score. In addition, they were also asked to select, in a binary choice, whether they thought the performance was generated by a human or computer.

Model (Params)	Musical Appeal Score ↑ (1-4)	‘Human-like’ Score ↑ (1-2)
AMT (360M)	3.33	1.57 (57.3%)
Figaro (87M)	2.73	1.57 (57.3%)
Cadenza (142M)	2.91	1.72 (72.0%)

Table 3. Human Evaluation Results for Model Quality. Percentages represent the fraction of modeling outputs that were selected by human evaluators when asked if it could have been created by a human.

5. RESULTS AND DISCUSSION

We discuss our experimental results to answer three high level questions - 1) provided an input MIDI sequence, how *musically related* are the Cadenza variations; 2) can the performer model tangibly improve expressivity; and 3) how appealing are the novel generations. We analyze our proposed scientific approach through quantitative and qualitative measures.

Both human and objective evaluations demonstrate in **Table 2** that training the composer with varying degrees of KL regularization has noticeable impacts on the balance between between recall and variety. Provided a musical idea as input, the *No-KL* model will produce outputs nearly identical to the input melody. Alternatively, the *Balanced-KL* model will produce outputs that are related, yet altered enough to provide new sources of inspiration. In many cases, the *Full-KL* model will produce entirely unrelated outputs, as a result of the encoder’s heavy focus on regularizing the latent vectors. Since a score of 4.0 for a generation would be considered identical we can infer that the

¹ Specifically, the music-medium-800k checkpoint

Model (Metric)	KL		Mean Δ		Std Dev Δ	
	Train	Opposite	Train	Opposite	Train	Opposite
HipHop (Velocity)	1.68	3.63	1.81	16.83	1.74	9.04
HipHop (Microtiming)	0.66	2.64	0.05	0.05	0.00	0.14
MusicNet (Velocity)	3.17	11.57	1.95	13.06	1.20	12.00
MusicNet (Microtiming)	0.07	3.17	0.02	0.13	0.01	0.15

Table 4. Objective results on the Performer fidelity evaluations.

ideas generated by *No-KL* and *Balanced-KL* are roughly 70% related to the input. This aligns with the quantitative results, which consistently show a negative correlation between the KL regularization and input/generation similarity metrics. As such, our framework is demonstrated to consistently generate variations that are perceptually relevant to, yet distinct from, the input musical idea.

In **Table 4**, we report results from the Performer Fidelity quantitative study. In both MusicNet and HipHop models, the distributions of predicted velocities and microtimings are consistently closer to that of their respective training datasets. We can therefore infer that the performer model, in conjunction with our newly proposed PerTok tokenizer, is capable of accurately learning the patterns of expressive characteristics from a comparatively small dataset.

In **Table 3** we compare Cadenza to AMT [14] and Figaro [15] on the task of novel generations. Our model, although comparable to its competitors on musical appeal, comes in second to the AMT. However, Cadenza comprehensively outperforms other models by 14.7% in the *human-like* expressivity ratings. We note that our framework is highly adaptable, in that both composer and performer models could be replaced with any type of sequential network. Theoretically, one could further improve the unconditional generation quality by replacing our VAE-based composer model with a decoder-only model, similar to that of AMT.

6. CONCLUSION

We introduced a multi-stage generative framework which allows for both variation and novel generation tasks, maintaining a competitive quality of composition and setting a new state-of-the-art of expressive characteristics. Our proposed tokenizer is used to create expressive symbolic sequences while effectively reducing vocabulary size and sequence length. We invite readers to our model page ² where we showcase its fidelity in generating outputs for polyphonic, monophonic, bass and drum instruments.

In particular, our performer model in conjunction with the new tokenization method led to a quantifiable increase in listener’s perceptions of the expressivity in the generations. These results were achieved with relatively tiny datasets, paving the way for further collaborations with artistic communities. Future research directions include exploring *controllability*, as well as further improvements in the domain of novel generation.

² Access code and demonstrations on our model page here. PerTok tokenizer is available as part of MidiTok library here.

7. ETHICS STATEMENT

As generative AI technology advances rapidly, it is crucial to address the implications of these developments in the generative domain. Concerns such as perpetuating cultural biases, undermining artists' financial opportunities, and using data without proper consent require urgent attention and dialogue within research communities. When developing new models, we must carefully consider both their intended applications and potential impacts.

Our research involves deep collaboration with artists to understand their motivations and needs, ensuring our efforts benefit the creative communities we serve. For instance, our new framework, designed for the MIDI symbolic domain, focuses on enhancing artists' tools with features that inspire creativity rather than replacing the artists. We also deliberately chose to work with smaller models, which helps minimize data requirements. This strategy promotes fair data agreements and increases the chances of fairly compensating musicians, thus fostering sustainability in creative industries and prioritizing ethical responsibility, especially in creative domains.

8. REFERENCES

- [1] J. Su, M. Ahmed, Y. Lu, S. Pan, W. Bo, and Y. Liu, "Roformer: Enhanced transformer with rotary position embedding," *Neurocomputing*, vol. 568, p. 127063, 2024.
- [2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *CoRR*, vol. abs/1706.03762, 2017. [Online]. Available: <http://arxiv.org/abs/1706.03762>
- [3] Y. Huang and Y. Yang, "Pop music transformer: Beat-based modeling and generation of expressive pop piano compositions," in *MM '20: The 28th ACM International Conference on Multimedia, Virtual Event / Seattle, WA, USA, October 12-16, 2020*, C. W. Chen, R. Cucchiara, X. Hua, G. Qi, E. Ricci, Z. Zhang, and R. Zimmermann, Eds. ACM, 2020, pp. 1180–1188. [Online]. Available: <https://doi.org/10.1145/3394171.3413671>
- [4] N. Fradet, J.-P. Briot, F. Chhel, A. El Fallah Seghrouchni, and N. Gutowski, "MidiTok: A python package for MIDI file tokenization," in *Extended Abstracts for the Late-Breaking Demo Session of the 22nd International Society for Music Information Retrieval Conference, 2021*. [Online]. Available: <https://archives.ismir.net/ismir2021/latebreaking/000005.pdf>
- [5] G. Hadjeres and L. Crestel, "The piano inpainting application," *CoRR*, vol. abs/2107.05944, 2021. [Online]. Available: <https://arxiv.org/abs/2107.05944>
- [6] C. Hawthorne, A. Stasyuk, A. Roberts, I. Simon, C.-Z. A. Huang, S. Dieleman, E. Elsen, J. Engel, and D. Eck, "Enabling factorized piano music modeling and generation with the MAE-STRO dataset," in *International Conference on Learning Representations, 2019*. [Online]. Available: <https://openreview.net/forum?id=r11YRjC9F7>
- [7] S. Oore, I. Simon, S. Dieleman, D. Eck, and K. Simonyan, "This time with feeling: Learning expressive musical performance," *CoRR*, vol. abs/1808.03715, 2018. [Online]. Available: <http://arxiv.org/abs/1808.03715>
- [8] A. Maezawa, K. Yamamoto, and T. Fujishima, "Rendering music performance with interpretation variations using conditional variational rnn." in *ISMIR, 2019*, pp. 855–861.
- [9] I. Borovik and V. Viro, "Scoreperformer: Expressive piano performance rendering with fine-grained control." in *ISMIR, 2023*, pp. 588–596.
- [10] S.-L. Wu and Y.-H. Yang, "Compose & embellish: Well-structured piano performance generation via a

- two-stage approach,” in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.
- [11] Y.-J. Shih, S.-L. Wu, F. Zalkow, M. Muller, and Y.-H. Yang, “Theme transformer: Symbolic music generation with theme-conditioned transformer,” *IEEE Transactions on Multimedia*, 2022.
- [12] H. H. Tan and D. Herremans, “Music fadernets: Controllable music generation based on high-level features via low-level feature modelling,” *arXiv preprint arXiv:2007.15474*, 2020.
- [13] S.-L. Wu and Y.-H. Yang, “Musemorphose: Full-song and fine-grained piano music style transfer with one transformer vae,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 1953–1967, 2023.
- [14] J. Thickstun, D. Hall, C. Donahue, and P. Liang, “Anticipatory music transformer,” *arXiv preprint arXiv:2306.08620*, 2023.
- [15] D. von Rütte, L. Biggio, Y. Kilcher, and T. Hoffman, “FIGARO: generating symbolic music with fine-grained artistic control,” *CoRR*, vol. abs/2201.10936, 2022. [Online]. Available: <https://arxiv.org/abs/2201.10936>
- [16] J. Ens and P. Pasquier, “MMM : Exploring conditional multi-track music generation with the transformer,” *CoRR*, vol. abs/2008.06048, 2020. [Online]. Available: <https://arxiv.org/abs/2008.06048>
- [17] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” *arXiv preprint arXiv:1312.6114*, 2013.
- [18] D. P. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling, “Improved variational inference with inverse autoregressive flow,” *Advances in neural information processing systems*, vol. 29, 2016.
- [19] J. Lucas, G. Tucker, R. Grosse, and M. Norouzi, “Understanding posterior collapse in generative latent variable models,” 2019. [Online]. Available: <https://openreview.net/forum?id=r1xaVLUYuE>
- [20] S. Zhao, J. Song, and S. Ermon, “Infovae: Information maximizing variational autoencoders,” *CoRR*, vol. abs/1706.02262, 2017. [Online]. Available: <http://arxiv.org/abs/1706.02262>
- [21] T. Ucar, “Bridging the ELBO and MMD,” *CoRR*, vol. abs/1910.13181, 2019. [Online]. Available: <http://arxiv.org/abs/1910.13181>
- [22] A. B. Dieng, Y. Kim, A. M. Rush, and D. M. Blei, “Avoiding latent variable collapse with generative skip models,” in *Proceedings of the Twenty-Second International Conference on Learning Representations and Statistics*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and M. Sugiyama, Eds., vol. 89. PMLR, 16–18 Apr 2019, pp. 2397–2405. [Online]. Available: <https://proceedings.mlr.press/v89/dieng19a.html>
- [23] O. Press and L. Wolf, “Using the output embedding to improve language models,” *arXiv preprint arXiv:1608.05859*, 2016.
- [24] B. Haki, M. Nieto, T. Pelinski, and S. Jordà, “Real-Time Drum Accompaniment Using Transformer Architecture,” in *Proceedings of the 3rd International Conference on AI and Musical Creativity*. AIMC, Sep. 2022. [Online]. Available: <https://doi.org/10.5281/zenodo.7088343>
- [25] J. Lenz, “Disentangle and deploy: Generative rhythmic tools for musicians,” Master’s thesis, Pompeu Fabra University, 2023, available at <https://zenodo.org/records/8380515>.
- [26] J. Devlin, M. Chang, K. Lee, and K. Toutanova, “BERT: pre-training of deep bidirectional transformers for language understanding,” *CoRR*, vol. abs/1810.04805, 2018. [Online]. Available: <http://arxiv.org/abs/1810.04805>
- [27] C. Raffel, “Learning-based methods for comparing sequences, with applications to audio-to-midi alignment and matching,” Ph.D. dissertation, Columbia University, 2016.
- [28] H. Fu, C. Li, X. Liu, J. Gao, A. Celikyilmaz, and L. Carin, “Cyclical annealing schedule: A simple approach to mitigating kl vanishing,” *arXiv preprint arXiv:1903.10145*, 2019.
- [29] J. Thickstun, Z. Harchaoui, and S. M. Kakade, “Learning features of music from scratch,” in *International Conference on Learning Representations (ICLR)*, 2017.

LOOKING FOR TACTUS IN ALL THE WRONG PLACES: STATISTICAL INFERENCE OF METRIC ALIGNMENT IN RAP FLOW

Nathaniel Condit-Schultz

natcs@gatech.edu

ABSTRACT

Musical rhythm and meter are characterized by simple proportional relationships between event durations within pieces, making comparison of rhythms between *different* musical pieces a nebulous practice, especially at different tempos. Though the “main tempo,” or tactus, of a piece serves as an important cognitive reference point, it is difficult to identify objectively. In this paper, I investigate how statistical regularities in rhythmic patterns can be used to determine how to compare pieces at different tempos, speculating that these regularities could relate to the perception of tactus. Using a Bayesian statistical approach, I model first-order (two-gram) rhythmic event transitions in a symbolic dataset of rap transcriptions (MCFlow), allowing the model to renotate the rhythmic values of each transcription as needed to optimize fit. The resulting model predicts makes “renotations” which match a priori predictions from the original dataset’s transcriber. I then demonstrate that the model can be used to rhythmically align new data, giving an objective basis for rhythmic annotation decisions.

1. INTRODUCTION

Symbolic representations of music generally encode rhythm using integer-related *note-value* categories—whether expressed as durations or inter-onset-intervals (ioi). Absolute timing is encoded indirectly (if at all) as the *tempo* of a reference note-value, conventionally the quarter-note. The musical and psychological validity of this approach is well established, as the schematic syntax of musical rhythm is primarily determined by proportional relationships, not absolute (clock-time) durations [1].¹ However, this approach also presents a problem: If only proportional relationships *within* a piece are rhythmically relevant, on what basis can relationships or comparisons be made across pieces? Can we be confident that a “quarter-note” in one piece is the same as a “quarter-note” in another? For example, consider three expert transcriptions of

¹ In fact, human perception tends to normalize ioi ratios that are *not* simple ratios to the nearest simple-ratio category [2].

songs by Johnny Cash from the *RS200* dataset [3]: “Folsom Prison Blues” (1955) and “Ring Of Fire” (1963) are transcribed with quarter-notes at 110_{bpm} and 104_{bpm} respectively, while “I Walk The Line” (1956) is transcribed at 210_{bpm} .² These three songs share many idiomatic musical features, including *backbeat* strikes in between bass-notes at a $105\text{--}110_{bpm}$ pulse. Given these similarities, perhaps the quarter-notes in “I Walk the Line” ought to be compared to the eighth-notes in “Ring of Fire.”

The quarter-note is more than a default reference unit for rhythmic encoding: It is also associated with the cognitive phenomenon of the “main beat” or *tactus*, and thus the “true” tempo of metric music [1, 4–6]. Other metric *levels* may be related to the tactus, both in notation and in human perception [1, 5, 6]. Thus, rhythmic comparison (in metric music) might be, essentially, a question of tactus comparison between two or more pieces. Which metric level in, for example, “Ring of Fire” or “I Walk the Line” is the tactus? This is essentially another perspective on the classic issue of “tempo octaves” in tempo-estimation research.

1.1 Background

Listeners must infer metric structure from music as they hear it [7], including the tactus level [5]. Though listeners’ metric interpretations often agree [8], disagreement is also common, especially regarding tactus [4, 5, 8–11]. This suggests that tactus inference is constrained, but not determined, by features of music’s objective organization. Which features constrain our perception of tactus? The obvious feature to consider is absolute (clock) time. Indeed, listeners tend to subdivide slower pulses or group faster pulses into beats in a preferred timing range, approximately corresponding to a tempo octave ($2/1$ ratio) of $160\text{--}80_{bpm}$ [5, 10, 12]. However, empirical measures of optimal tempo ranges have often covered larger ratios—from $2.25/1$ [12] to $2.5/1$ [13]—and a non-trivial number of observations spread across even more extreme tempos [4, 10]. Tempos from $200\text{--}60_{bpm}$ will feel somewhat familiar to most musicians, creating an “apparent contradiction between the narrow range of preferred tempi and the wide range of (absolute) tempi found in real music” [14]. These findings demonstrate that tactus perception is not determined by absolute timing in a trivial manner. Even if a strict tempo-octave were used for comparison, this still requires an arbitrary choice of the cutoff between tempo-octaves [10].

² “Ring of Fire” was transcribed by Temperley, the other two songs by de Clercq.



Relative rhythmic features also contribute to tactus perception [4, 13–15]. In particular, the density and consistency of attacks at particular metric levels—what Martens [4] calls “pulse consistency”—serve as an important cue [6, 10]. Music theorists have also noted specific rhythmic patterns or aspects of the music’s feel³ that relate to tactus. A notable example is the backbeat pattern evident in the Johnny Cash examples above, which is often regarded as tactus defining [5, 17, 18]. However, De Clercq [17] has argued that absolute speed overwhelms the backbeat norm in many cases, and musical features must be balanced with absolute speed when inferring the tactus.

In traditions that rely on notated music, composers’ explicit choice of note values and time signature might be regarded as the “correct” tactus; However, many scholars have noted that classical time signatures leave room for ambiguity regarding the true tactus [4, 10, 17, 19, 20]. Music from vernacular traditions pose an even more acute problem for research, as rhythm values must be chosen by a scribe [17]. If theory, convention, and intuition serve, we might hope that homogeneous collections of scores are coherently aligned. Unfortunately, representing metric alignment across pieces is not necessarily an important goal in traditions of music notation, and there are no clear standards for composers, transcribers, or arrangers to follow.

1.2 Hypothesis

If metric orientation is essential to the syntactic organization of music, then proper metric alignment of pieces is necessary to reveal structural similarities and generalize about rhythmic syntax in a body of music [17]. Conversely, any “misaligned” pieces—like “I Walk the Line,” perhaps—add noise to empirical distributions and hinder musicological analysis. In this paper, I explore a novel statistical approach to aligning and comparing rhythmic patterns across pieces within stylistically homogeneous musical corpora. I hypothesize that regularities in proportionally-encoded rhythmic patterns can serve as consistent cues of metric alignment of pieces, independent of absolute speed. In other words, that specific rhythmic patterns or features (notably, pulse saliency) will be statistically associated with particular metric levels, and that these patterns can then be used as the basis to align and compare pieces. To achieve this, we can systematically rescale note values of transcriptions—either in *augmentation* (longer values) or *diminution* (shorter values)—so as to optimize the fit of statistics related to syntactic rhythm relationships. For example, we could renotate “I Walk the Line” in diminution, and then confirm if the resulting tabulation of the overall RS200 collection is less noisy, “expos[ing] connections that would be otherwise hidden or obscured” [17]. My argument is that these connections, should they be revealed, may relate to listeners’ perceptual

³ Another plausible area where musical organization might influence metric alignment is sub-syntactic *micro-timing*: small discrepancies between actual rhythmic timing and their perceived rational categories. Micro-timing is often related to the “feel” of music, and can be used to emphasize particular beat levels [16].

experience of the tactus, though I cannot directly demonstrate that here.

A central premise of my hypothesis, is that metric alignment can be done based on proportional rhythmic data, without absolute timing information. This does not preclude that absolute timing plays an important role in musical alignment, but if the hypothesis is supported, it would demonstrate that rhythmic syntax is at least partly independent of tempo, and help explain why tempos are used outside a preferred tempo octave.

2. METHODOLOGY

With no ground truth available, I can only attempt to optimize fit to my data in an unsupervised way. My approach is to characterize empirical probability distributions of rhythmic data conditioned on different interpretations of the metric alignment of pieces.

2.1 Data

For this project, I use my own Musical Corpus of Flow (MCFlow) [18], in which I transcribed the rapped part of 124 popular hip-hop songs, all in $\frac{4}{4}$ time. Rap flow is suitable for this task for several reasons: Rap flow is saturated with the rhythmic features of American popular music more broadly, with lots of rhythmic variation within songs. Rap also tends to exhibit a rhythmically dense, fast pace, with few long iois, which makes relatively simple ngram-like analyses (described below) more plausible. I parsed the MCFlow dataset using humdrumR [21]—a R package for analyzing data encoded in the humdrum syntax (as MCFlow is). I restrict my analysis to inter-stress-intervals because rap scholars agree that most useful rhythmic information is in the stressed syllables of rap [18, 22].

MCFlow divides each of its 124 songs into verses. In some cases, different artists perform different verses, occasionally even at different tempos. I thus regard each verse as a separate rhythmic passage to analyze. To isolate only “pure” duple rhythmic data, I remove 155 measures of music, in 44 unique verses, which contain at least one triplet. I then removed 16 verses with fewer than eight measures remaining. This leaves a total of 376 verses, containing 36,553 stressed syllables; the shortest remaining verse has only 21 stressed syllables, with the longest containing 314 and a median length of 98.

In my MCFlow transcriptions, I used the backbeat in the rap’s accompaniment to determine the quarter-note value [18]. However, one of the most important reasons I use MCFlow is because I [23] originally noted thirty-five verses (in eleven⁴ songs) which are clear outliers in tempo annotation (Table 1), and speculated that they might be better notated at a different tempo. This gives us a set of *a priori* predictions about metric alignment in the data. Figure 1 illustrates the distribution of quarter- and eighth-note iois in MCFlow, as notated (above) and incorporating my speculated renotations (below). Interestingly, the

⁴ I also identified two other outlier verses which I exclude because they contained fewer than eight measures of non-triplet bars.

Song	Verse(s)	BPM
Dead and Gone (T.I., 2009)	1–2	68
Niggas in Paris (Jay Z and Kanye West, 2011)	1–3	70
Mercy (Kanye West, at al., 2012)	1,2,4	70
What’s Your Fantasy (Ludacris, 2000)	1–3	70
Holy Grail (Jay Z, 2013)	1–2	72
How Low (Ludacris, 2009)	1–2	72
Woof (Snoop Dogg, 1998)	1–3	83
Pray (M.C. Hammer, 1990)	1–5	122
It’s Tricky (Run-D.M.C., 1987)	1–4	128
You Be Illin’ (Run-D.M.C., 1986)	1–4	128
Fight for Your Right (the Beastie Boys, 1987)	1–3	134
Mercy (Kanye West, at al., 2012)	3	140

Table 1. List of verses in MCFlow which Condit-Schultz [18] identified as tempo outliers. BPM = quarter-notes per minute.

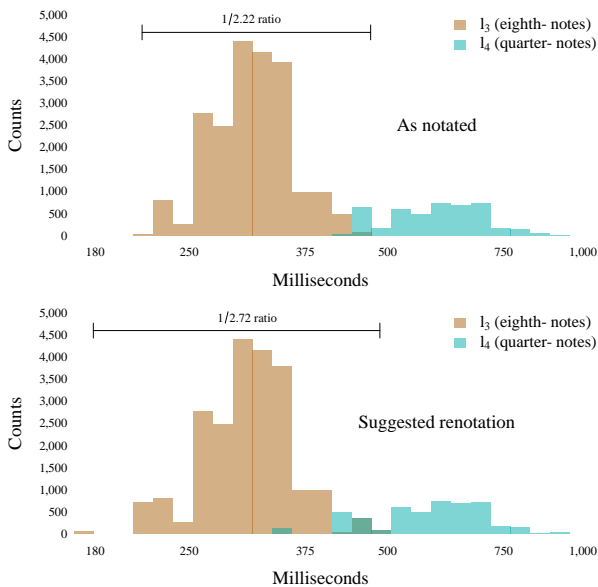


Figure 1. Distribution of notated quarter- and eighth-note inter-stress-intervals in MCFlow, by absolute duration.

raw, backbeat-based notation covers an tempo range only slightly greater than one tempo octave, similar to observed listener preferences [12, 13]. In contrast, following my speculated renotations results in a few verses being moved into more extreme absolute tempos.

2.2 Meter

Meter is an organizational structure in music, wherein multiple phase-aligned beats with integer-related periods form a nested hierarchical pattern [5]. These beats can be sorted from fastest (“lowest”) to slowest (“highest”), each considered one *metric level*, notated here as $[l_1, \dots, l_k]$. The highest metric level (l_k) defines the overall period of the meter, a *measure*; the lowest metric level (l_1) is known as the *tatum*. In a musical passage, each note onset is associated with a tatum pulse [24], and thus a unique *metric position* within each measure—e.g., “beat 4.” Metric positions may also coincide with one or more higher-metric levels, with the highest level defining the “level” of that position. For example, the downbeat of each measure is the unique position at level l_k .

In this paper, I consider only simple duple meter, with each metric level having twice the period of the level below it [1]: essentially a $\frac{4}{4}$ meter with strictly no triplets. The standard $\frac{4}{4}$ generally presumes at least three central levels [5, 20]. However, music often evinces hyper-metric pulses above the measure level [20] and, conversely, faster levels well below the ostensible tatum (e.g., 16th- and 32nd-notes). Thus, I proceed with a slightly expansive $k = 6$: six metric levels with 32 metric positions. This could be interpreted as one measure of 32nd-notes, two measures of sixteenth-notes, or four measures of eighth-notes. Throughout this paper, I will take l_1 as 32nd-notes, putting quarter-notes in l_4 .

Regardless of notation, the fastest metric level (tatum) can always be identified in any transcription. However, some musical passages may have *implicit* subdivisions, that would be felt by a listener, but are never articulated in the music. Thus, the *true* tatum l_1 may be different than the observed tatum l_1 . For any given musical transcription, we can postulate one or more implicit subdivisions, effectively “shifting” the observed metric positions up one level—equivalent to renotating the music in augmentation.

2.2.1 Modeling Meter

To characterize the rhythms of music in metric terms, I use a first-order (two-gram) model, considering the joint probability of the metric positions of sequential pairs (antecedent-consequent) of note events. Given 32 positions, a full transition matrix would require 1,024 parameters, many of which would be close to zero or simply redundant, as rhythmic patterns in different parts of the measure can be closely related. To work with less sparse and more interpretable parameters, I explored ways of reducing the full 32x32 parameters space to a smaller number of parameters while maintaining predictive power. My final approach is to bin each antecedent note according to its metric level l_k and each consequent into one of nine categories defined relative to the antecedent position. My nine *metric consequent types*, illustrated in Figure 2, are able to differentiate between shorter and longer iois, weak-to-strong versus strong-to-weak beat transitions, and different sorts of syncopations. These forty-seven parameters are represented in a vector \mathbf{p} , with components $p_{l,m}$ corresponding to the probability of each metric transition (Table 2). In the raw MCFlow data, the full 32X32 metric transition matrix has a joint entropy of 6.19 bits (10 being the maximum theoretical value). My antecedent-consequent parameterization, with only 47 parameters, achieves a cross-entropy with the same data of 7.44 bits, gaining only 1.25 bits⁵ by removing 977 parameters. Figure 3 illustrates the distribution of my antecedent-consequent parameters, using the raw MCFlow note-values.

2.3 Statistical Model

The statistical model I employ mirrors the thought process we explored with Johnny Cash songs above, “renotating”

⁵ This difference in bits is equivalent to the Kullback–Leibler divergence.

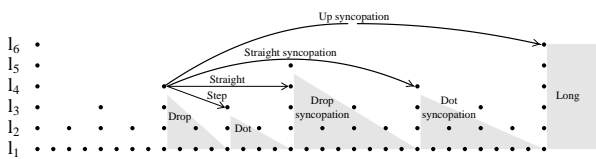


Figure 2. Illustration of nine metric consequent types at the l_4 level. Arrows point to exact points; shaded areas indicate points binned together.

	Drop	Step	Dot	Straight	Drop sync.	Straight sync.	Dot sync.	Up sync.	Long
l_6	$p_{6.1}$	$p_{6.2}$	$p_{6.3}$	$p_{6.4}$	$p_{6.5}$		$p_{6.7}$		$p_{6.9}$
l_5	$p_{5.1}$	$p_{5.2}$	$p_{5.3}$	$p_{5.4}$	$p_{5.5}$	$p_{5.6}$	$p_{5.7}$	$p_{5.8}$	$p_{5.9}$
l_4	$p_{4.1}$	$p_{4.2}$	$p_{4.3}$	$p_{4.4}$	$p_{4.5}$	$p_{4.6}$	$p_{4.7}$	$p_{4.8}$	$p_{4.9}$
l_3	$p_{3.1}$	$p_{3.2}$	$p_{3.3}$	$p_{3.4}$	$p_{3.5}$	$p_{3.6}$	$p_{3.7}$	$p_{3.8}$	$p_{3.9}$
l_2	$p_{2.1}$	$p_{2.2}$	$p_{2.3}$	$p_{2.4}$	$p_{2.5}$	$p_{2.6}$	$p_{2.7}$	$p_{2.8}$	$p_{2.9}$
l_1				$p_{1.4}$		$p_{1.6}$		$p_{1.8}$	$p_{1.9}$

Table 2. Forty-Seven metric coefficients (\mathbf{p}). Empty slots are logically impossible given the definitions. sync. = syncopation.

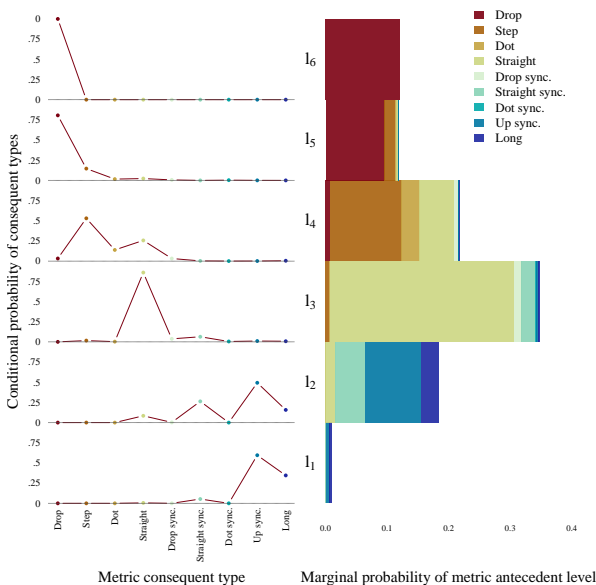


Figure 3. Raw empirical estimates for $p_{l,m}$. Both sides of the figure show the same information in two different formats: the left side shows the conditional probability of each metric consequent, given the metric level of an antecedent syllable; the right side shows the joint probability of the same antecedent-consequent pairings.

verses in the MCFlow corpus to find a good fit. With each iteration of the model’s Monte Carlo algorithm, the model finds estimates of the metric coefficients (explained below) using the dataset as currently encoded. The model then estimates parameters which represent the “scaling” of each individual verse, by retabulating the music assuming one or two unobserved sub-divisions—equivalent to renotating the music in augmentation. The process repeats, reestimating the meter parameters using the new scaling parameters, etc., until a complete picture of the posterior distribution emerges, as guaranteed by the Metropolis-Hasting algorithm [25]. Ultimately, I find the scalings of each verse that result in the best fit to the overall metric distribution.

In each verse, I count instances of 47 metric transition bins, indexed $l.m$ as defined above for \mathbf{p} . Let the counts in the n th verse be labeled $C^n = [c_{l,m}^n, \dots]$. I then model each set of counts as an independent draw from a multinomial distribution $C^n \sim \mathcal{M}(\sum C^n, \mathbf{p})$. The core purpose of the project, however, is to estimate a set of indicator, “shift,” parameters, one for each verse: $\mathbf{s} = [s^1, \dots, s^n]$, where $s^n \in \{0, 1, 2\}$. There are thus actually three different counts ($C^{n(s \in \{0,1,2\})}$) for each verse, one for each possible shift parameter: When $C^{n(s=0)}$, the metric parameters are counted assuming the observed tatum is the true tatum $l_{\bar{1}} = l_1$. When $C^{n(s=1)}$, count assuming that there is one implicit level of duple subdivision in the meter, $l_{\bar{1}} = l_2$, “shifting” the metric parameters up one level. When $C^{n(s=2)}$, counted assuming two subdivisions, $l_{bar1} = l_3$. I assume also that the values of $\mathbf{s} \sim \mathcal{M}(n, \mathbf{S})$, where $\mathbf{S} = [S_0, S_1, S_2]$ is another discrete probability distribution (though this ultimately had little impact on my results).

2.3.1 Model Estimation

Given the assumed distributions above, I use a Bayesian Markov Chain Monte Carlo (MCMC) algorithm to calculate posterior distributions for \mathbf{p} , \mathbf{s} , and \mathbf{S} . Since objective estimates for the \mathbf{s} shifting parameters are my main goal, I specify no prior distribution for \mathbf{s} , letting the model believe (initially) that all values of \mathbf{s} are equally probable. For \mathbf{p} and \mathbf{S} , I specify minimally informed Dirichlet prior distributions: $prior(\mathbf{p}) \sim Dir(\alpha_{l,m} = 5)$ and $prior(\mathbf{S}) \sim Dir(\alpha_S = 5)$. These minimal priors—equivalent to observing 235 prior note transitions and 15 prior verse shifts respectively—mainly serve to (weakly) discourage the model from assigning values close to zero. Note that the Bayesian approach here does not only find the optimal point-estimate for each parameter, but a complete prior distribution of belief regarding each parameter. This will allow the model to express degrees of certainty about each s^n , rather than finding only one optimal choice.

I estimate the posterior distribution using a custom MCMC implementation in R, with three Gibbs-sampler steps (for \mathbf{s} , \mathbf{S} , and \mathbf{p}) in each iteration, i . In each Gibbs step, I sample new parameter estimates for one parameter from the conditional distribution of that parameter given the current values of the other parameters. The result is a sequence of estimates for each parameter, forming a

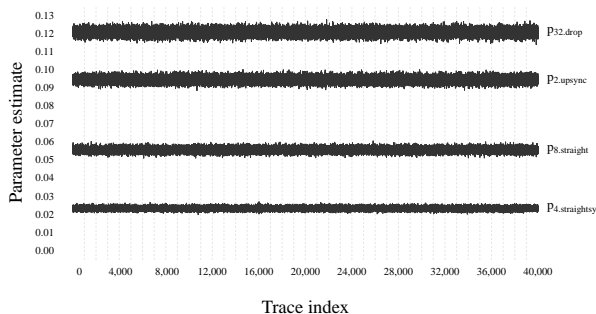


Figure 4. MCMC trace for four selected parameters. Dashed lines indicate boundaries between independent chains.

Markov chain which converges on the true posterior distribution.

For the \mathbf{s} scaling parameters, new estimates for all $[s^1 \dots s^n]$ parameters are sampled in a single step. For each verse, the probability of observing $C^{n(s)}$ for all three values of s , conditioned on \mathbf{p}_i and \mathbf{S}_i , is computed.

$$s_{i+1}^n \sim P(C^{n(s)} | \mathcal{M}(\sum_1^n C_i^{n(s)}, \mathbf{p}_i)) * \mathbf{S}_i$$

For the \mathbf{p} metric coefficient parameters, new estimates for all parameters were sampled in a single step, conditional only on \mathbf{s}_i . Taking advantage of the conjugate relationship between the multinomial and Dirichlet distributions, I can sample from the conditional distribution of \mathbf{p} directly using the Dirichlet distribution:

$$\mathbf{p}_{i+1} \sim Dir(\alpha = \sum_{n=1}^N C^{n(s_i)} + prior_\alpha(\mathbf{p}))$$

Updates for \mathbf{S} are similar but even simpler, using only the current (estimated) counts of \mathbf{s} : $\mathbf{S}_{i+1} \sim Dir(\alpha = counts(\mathbf{s}_i) + prior_\alpha(\mathbf{S}))$.

To minimize the effect of initial values, I initialized forty independent markov chains on different random draws from the prior distributions of \mathbf{p} and \mathbf{S} , and a uniform random sample of \mathbf{s} parameters. Each chain ran for 11,000 samples, with an initial “burn in” of 1,000 iterations removed from each chain, though each chain appeared to reach its stationary distribution well before the 1,000th iteration. All forty chains converged on the same final distributions for all parameters (see Figure 4 for a few examples). As is usually the case with MCMC models, several parameter chains evinced moderate autocorrelation values (the largest being 0.287), so I thinned the chain by taking every tenth sample, cutting the absolute autocorrelation values down to $r \leq 0.115$. The result is a chain of 40,000 samples for each parameter. Figure 4 shows the MCMC traces for four of the \mathbf{p} parameters; the other parameter traces look essentially identical.

3. RESULTS

The main parameters I am interested in are the estimates of \mathbf{s} , the “shift” parameters for each verse. Despite the fairly long MCMC trace (40,000 samples), in 372 of 376 verses the model selected the same shift parameter in every sample; evidently, most verses fit in one, and only one, interpretation. In only three verses—none of which were *a priori* tempo outliers—did the model find significant uncertainty, with the non-modal choice sampled between 12.6% and 43.9% of the time (these appear midway between shift levels in Figure 5). The important question is whether these highly confident shift parameters match my *a priori* expectations. If we take the posterior modal value for each s^n , we observe 37 shifts of 0, 318 of 1, and 21 of 2. Figure 5 shows these average posterior \mathbf{s} values normalized relative to the original empirical l_1 of each verse, such that 0 indicates the original notated quarter-note. The model correctly identifies the predicted renotation for 27 of the 35 *a priori* outliers. The model also identifies four unanticipated verses that need shifting, and fails to shift eight verses—if we view this as a binary classification task, the model achieves an F-score of .818. Note that the model was not provided any information about absolute timing, so this accuracy is achieved purely by looking at metric transitions. Close investigation of the false positives reveals that, though I didn’t originally identify these verses as outliers [18], each features flow that could make sense renoted. The false negatives are not as easy to interpret; However, in no case did the model falsely reject all outlier verses in a song: for example, the model correctly shifts four of the five verses in MC Hammer’s “Pray,” but fails to shift the fourth verse (for no obvious reason).

To visualize the posterior distribution of \mathbf{p} , the metric coefficients, I show the average posterior value for each $p_{l.m}$ parameter in Figure 6. If we compare this to Figure 3, there are no dramatic differences, except at l_1 , where the model places considerably less probability mass. The average entropy of the \mathbf{p} posterior is 3.339 bits, slightly lower than the joint distribution of the raw-notation counts at 3.409 bits, demonstrating that model has improved the overall fit of the data. Finally, I can also use the posterior parameters to evaluate unseen data. For example, if I apply the posterior \mathbf{p} to our three Johnny Cash songs, we find that the model shows (with total confidence) that the three songs *should* be aligned at the same backbeat level, as I speculated at the outset.

4. DISCUSSION AND CONCLUSION

Though it appears that my statistical approach both improves fit and matches (my) expert judgments [18], this initial foray is not a decisive demonstration that this approach can help us generalize about metric syntax. It appears that my model is learning, or at least observing, *something* about the organization of metric syntax across metric levels, but future work is needed to elucidate what is going on, and determine how robust this methodology can be.

It may seem that my full Bayesian treatment is overkill:

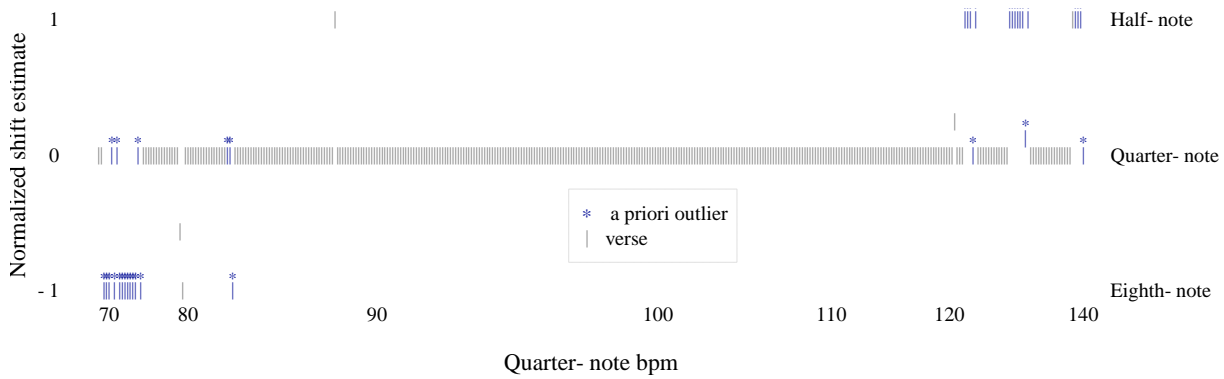


Figure 5. Average posterior s estimates for each verse, sorted by raw quarter-note bpm. Blue marks indicate the predicted tempo outliers. The shifts have been "normalized" such that 0 indicates the original notated quarter-note, rather than the fastest metric level.

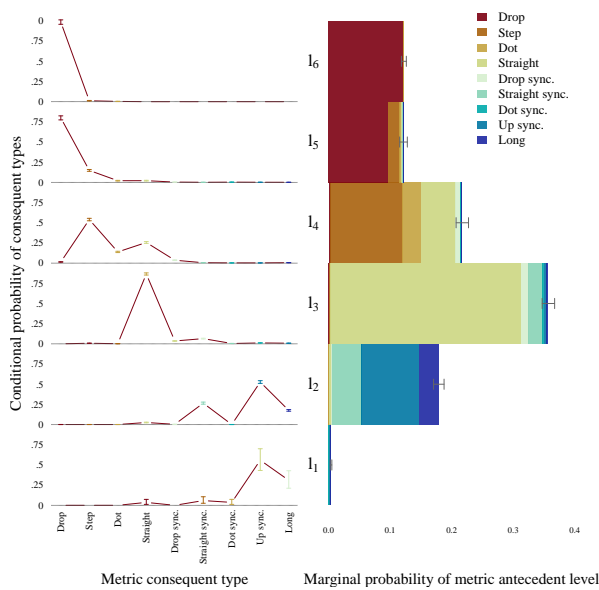


Figure 6. Mean posterior parameter estimates for $p_{l,m}$. Both sides of the figure show the same information in two different formats: the left side shows the conditional probability of each metric consequent, given the metric level of an antecedent syllable; the right side shows the joint probability of the same antecedent-consequent pairings. Bars indicate the Bayesian 95% credible interval for each parameter.

My parameter estimates are tightly packed around their mean and not dramatically different than the simple counts derived from the raw data; my posterior estimates of s also show little variability. This suggests that a simpler approach could probably achieve similar results on this dataset. My results are also strongly fitted to this particular dataset—for this initial attempt I specified uninformative priors on all parameters, allowing the model to fit the data at hand very closely. However, I believe this full Bayesian approach will prove robust if extended to other datasets which might be less rhythmically uniform than MCFlow, and the results here could be used as the basis for more informative priors for future work.

Finally, though I have argued that this task is theoretically connected to perceptual and musicological ideas of tactus and tempo, future work with human participants will be necessary to establish direct connections between my findings and human perception. For example, my p estimates could be used to generate rhythmic stimuli with different (predicted) tactus interpretations. For course, as discussed above, there is plenty of evidence that tactus is never fully determined by musical features [4, 5, 8–11]. Listeners’ perception might be shaped previous context, personal experience, their own personal state, or by conscious effort. My analysis of “syntactic regularities,” even if valid, isn’t necessarily connected to tactus at all: indeed, at least one prominent psychomusicological theory of rhythm, London’s [20] (p. 95) *tempo-metrical types*, “is [explicitly not] defined in terms of the level heard as the tactus.” It’s possible that the statistical regularities found by my model represent tempo-metrical types, or other rhythmic structural principles, but *not* tactus.

Basing psychological conclusions on statistical evidence requires a match between the musical corpora and the listening experience of people. Different musical exposure (and thus statistical experience) might explain disagreements about tactus. Stepping back further, it is possible that syntactic relationships in music involve relationships between various rhythms and beats *without* assuming any privileged reference level at all. My results here make this final possibility appear unlikely, but much work remains to be done.

5. REFERENCES

- [1] M. R. Jones and M. G. Boltz, "Dynamic attending and responses to time," *Psychological Review*, vol. 96, no. 3, p. 459–491, 1989.
- [2] N. Jacoby and J. H. McDermott, "Integer ratio priors on musical rhythm revealed cross-culturally by iterated reproduction," *Current Biology*, vol. 27, no. 3, p. 359–370, 2017.
- [3] D. Temperley and D. Tan, "Emotional connotations of diatonic modes," *Music Perception*, vol. 30, no. 3, p. 237–257, February 2013.
- [4] P. A. Martens, "The ambiguous tactus: Tempo, subdivision benefit, and three listener strategies," *Music Perception*, vol. 28, no. 5, p. 433–448, June 2011.
- [5] P. A. Martens and F. Benadon, "Musical structure: Time and rhythm," in *The Routledge Companion to Music Cognition*, R. Ashley and R. Timmers, Eds. Taylor & Francis, 2017, p. 115–127. [Online]. Available: <https://doi.org/10.4324/9781315194738>
- [6] R. Parncutt, "A perceptual model of pulse salience and metrical accent in musical rhythms," *Music Perception*, vol. 11, no. 4, p. 409–464, July 1994.
- [7] H. C. Longuet-Higgins and C. S. Lee, "The rhythmic interpretation of monophonic music," *Music Perception*, vol. 1, no. 4, p. 424–441, July 1984.
- [8] C. K. Madsen, R. A. Duke, and J. M. Geringer, "The effect of speed alterations on tempo note selection," *Journal of Research in Music Education*, vol. 34, no. 2, p. 101–110, 1986. [Online]. Available: <http://www.jstor.org/stable/3344738>
- [9] D. Hammerschmidt and C. Wöllner, "Sensorimotor synchronization with higher metrical levels in music shortens perceived time," *Music Perception*, vol. 37, no. 4, p. 263–277, March 2020.
- [10] M. F. McKinney and D. Moelants, "Ambiguity in tempo perception: What draws listeners to different metrical levels?" *Music Perception*, vol. 24, no. 2, p. 155–166, December 2006.
- [11] P. Toiviainen and J. S. Snyder, "Tapping to bach: Resonance-based modeling of pulse," *Music Perception*, vol. 21, no. 1, p. 43–80, September 2003.
- [12] L. van Noorden and D. Moelants, "Resonance in the perception of musical pulse," *the Journal of New Music Research*, vol. 28, no. 1, p. 43–66, 1999.
- [13] S. Quinn and R. Watt, "The perception of tempo in music," *Perception*, vol. 35, no. 2, p. 267–280, 2006, PMID: 16583770. [Online]. Available: <https://doi.org/10.1068/p5353>
- [14] G. Madison and J. Paulin, "Ratings of speed in real music as a function of both original and manipulated beat tempo." *the Journal of the Acoustical Society of America*, vol. 128, no. 5, p. 3032–3040, 2010. [Online]. Available: <http://search.proquest.com/docview/815547411/>
- [15] J. London, "Tactus \neq tempo: Some dissociations between attentional focus, motor behavior, and tempo judgment," *Empirical Musicology Review*, vol. 6, no. 1, p. 43–55, 2011.
- [16] C. Drake, A. Penel, and E. Bigand, "Tapping in time with mechanically and expressively performed music," *Music Perception*, vol. 18, no. 1, p. 1–23, October 2000.
- [17] T. de Clercq, "Measuring a measure: Absolute time as a factor for determining bar lengths and meter in pop/rock music," *Music Theory Online*, vol. 22, no. 3, 2016. [Online]. Available: <https://doi.org/10.30535/MTO.22.3.3>
- [18] N. Condit-Schultz, "MCFlow: A Digital Corpus of Rap Transcriptions," *Empirical Musicology Review*, vol. 11, no. 2, p. 124–147, 2016.
- [19] M. R. Jones, R. R. Fay, and A. N. Popper, Eds., *Music Perception*, ser. Springer Handbook of Auditory Research. Springer, 2010, vol. 26.
- [20] J. London, *Hearing in Time: Psychological aspects of musical meter*, 2nd ed. Oxford: Oxford University Press, 2012.
- [21] N. Condit-Schultz and C. Arthur, "humdrumr: a new take on an old approach to computational musicology," in *Proceedings of the International Society for Music Information Retrieval*, November 2019, p. 715–722.
- [22] M. Ohriner, *Flow: The rhythmic voice in rap music*. Oxford Studies in Music Theory, 2019.
- [23] N. Condit-Schultz, "MCFlow: A Digital Corpus of Rap Flow," Ph.D. dissertation, Ohio State University, 2016.
- [24] F. Lerdahl and R. Jackendoff, *A Generative Theory of Tonal Music*. Cambridge, Massachusetts: The MIT Press, 1983.
- [25] S. Chib and E. Greenberg, "Understanding the metropolis-hastings algorithm," *The American Statistician*, vol. 49, no. 4, pp. 327–335, 1995. [Online]. Available: <http://www.jstor.org/stable/2684568>

EXPLORING GPT’S ABILITY AS A JUDGE IN MUSIC UNDERSTANDING

Kun Fang^{1,2}

Ziyu Wang^{3,2}

Gus Xia^{2,3}

Ichiro Fujinaga¹

¹ Schulich School of Music, McGill University

² Machine Learning Department, MBZUAI

³ Computer Science Department, NYU Shanghai

kun.fang@mail.mcgill.ca, ziyu.wang@nyu.edu,
gus.xia@mbzuai.ac.ae, ichiro.fujinaga@mcgill.ca

ABSTRACT

Recent progress in text-based Large Language Models (LLMs) and their extended ability to process multi-modal sensory data have led us to explore their applicability in addressing music information retrieval (MIR) challenges. In this paper, we use a systematic *prompt engineering* approach for LLMs to solve MIR problems. We convert the music data to symbolic inputs and evaluate LLMs’ ability in detecting annotation errors in three key MIR tasks: beat tracking, chord extraction, and key estimation. A *concept augmentation* method is proposed to evaluate LLMs’ music reasoning consistency with the provided music concepts in the prompts. Our experiments tested the MIR capabilities of Generative Pre-trained Transformers (GPT). Results show that GPT has an error detection accuracy of 65.20%, 64.80%, and 59.72% in beat tracking, chord extraction, and key estimation tasks, respectively, all exceeding the random baseline. Moreover, we observe a positive correlation between GPT’s error finding accuracy and the amount of concept information provided. The current findings based on symbolic music input provide a solid ground for future LLM-based MIR research.¹

1. INTRODUCTION

Recent advancements in text-based Large Language Models (LLMs) have showcased their significant reasoning and knowledge retrieval capabilities across various domains, including music understanding. For instance, the standard GPT-4 model performs better than random on music theory questions [1]. This success raises the question of whether such text-based reasoning abilities could enhance Music Information Retrieval (MIR) tasks. From a psychological perspective, we are interested in how a *cognition module*, typically represented by a text-based LLM, can possibly

interplay with a *perception module*, typically represented by an MIR network, to improve music understanding.

A key challenge in achieving this goal is the inherent difference between music and text modality, which typically requires aligning data in other modalities to text. Common strategies include either transforming all inputs into a unified modality [1, 2], or developing adapters tailored to other domains, such as MiniGPT-5 [3] and NextGPT [4]. Given the substantial data requirements and training costs involved in addressing cross-modality issues, we believe a practical initial step for LLM-based MIR research is to translate sensory inputs into symbolic representations and investigate the performance of text-based LLMs in a training-free way (e.g., prompt engineering [5]). This methodology allows us to assess how much cognition alone, without additional auditory perception, can enhance MIR tasks.

To this end, we propose a systematic prompt engineering method to assess the music understanding capabilities of text-based LLMs, focusing specifically on their ability to *detect errors* in MIR annotations. Each task input includes: 1) a music segment converted into MIDI or higher-level musical features, 2) a corresponding MIR annotation with deliberately inserted errors, and 3) a text prompt that introduces the MIR problem and outlines relevant musical concepts. The LLM’s role is to pinpoint errors within the musical annotations, effectively acting as an MIR judge. In all the tasks, annotation errors are randomly applied at controlled rates, and prompts are crafted using common prompt engineering techniques. Additionally, we propose a *concept augmentation* strategy to evaluate the LLM’s behavioral consistency in response to the musical concepts provided. This involves adjusting the occurrence of certain musical concepts in the prompt, such as replacing a musical term (e.g., pitch sequence) with a more general term (e.g., time series) to obscure a concept, or vice versa, to explore whether these changes influence the LLM’s performance in predictable ways.

We carried out experiments using the GPT-3.5 model (hereafter, GPT), targeting three MIR tasks: beat tracking, chord extraction, and key estimation. The experiment results indicate that the error detection rates are higher than random, achieving scores of 65.20%, 64.80%, and 59.72%, respectively. Furthermore, the concept augmentation experiments show that GPT’s performance broadly

¹ Code repository: <https://github.com/kunfang98927/gpt-eval-mir>



correlates with the amount of musical concepts introduced in the prompts. These findings suggest that GPT exhibits measurable music understanding capability, which sets a foundational baseline for future LLM-based MIR research. In sum, the contributions of the paper are as follows:

1. **We pioneer the integration of MIR problems with text-based LLMs.** Our approach utilizes prompt-engineering techniques for MIR error detection and adopts the symbolic music format to unify music and text modality, which does not require additional training.
2. **We perform a systematic study on GPT’s abilities as a judge** in beat tracking, chord extraction, and key estimation tasks, demonstrating GPT’s capability in solving MIR problems.
3. **We provide a solid ground for LLM-based MIR research.** The proposed methodology sets a baseline for future studies.

2. RELATED WORK

Recently, the advancements of text-based LLMs [6–8] have expanded beyond textual data, incorporating capabilities to interpret information from various other modalities. In the computer music domain, the research to combine text and audio LLMs is also popular. For example, Chat-Musician is a text-based LLM, which focuses mainly on generating symbolic music in ABC notation [1]; Music-Gen [9] and Coco-Mulla [10] are audio-based LLMs allowing text and symbolic music control; and MU-LLaMA is an audio-to-text model for caption generation [11]. Despite all these achievements, the current cross-modal research of text-based LLMs is restricted to generative tasks; and their ability to reason about cross-domain data is still under-researched. The focus of this paper is to evaluate whether LLMs can be used for music understanding and solving MIR problems.

In most cross-modal LLM studies, extensive training is required to align cross-modal information. These approaches involve training separate adapters to align the pre-trained model with other-domain data [3,4,12], fine-tuning an LLM on symbolic cross-domain data [1], or learning a trainable autoencoder to convert other-domain data to text tokens [2]. In the music domain, since music can be naturally represented as readable symbolic representations, we propose using prompt engineering methods to connect the music and text domains to avoid extra training.

The cross-domain prompt engineering methods used in this paper originate from the text domain. These strategies involve chain-of-thought [5], few-shot prompting [13], least-to-most prompting [14], and many others [15–17]. These methods show that the more organized the prompt is, the better the LLM will be able to reason. To the best of our knowledge, we present the first attempt of using prompt engineering to teach LLMs to reason about music. We aim to explore to what extent music reasoning alone can help MIR.

3. METHODOLOGY

In this study, we use prompt engineering to evaluate the capabilities of text-based LLMs through three MIR error detection tasks: beat tracking, chord extraction, and key estimation (as shown in Figure 1). In Section 3.1, we introduce the task definition and data representations for each task. In Section 3.2, we discuss the structure and main components of the prompts. Finally, Section 3.3 introduces the proposed concept augmentation methods to test the LLMs’ music reasoning ability with respect to the music concepts included in prompts.

3.1 Task Definition and Data Representation

For symbolic MIR tasks, beat tracking determines the precise timing of beats in a MIDI-like music representation, chord extraction assigns a chord label to each segment, and key estimation identifies the musical key of each segment. Building on these tasks, we introduce a novel task: MIR error detection. This task involves identifying errors specific to each of the three traditional MIR tasks. The following subsections define the error detection tasks for beat tracking, chord extraction, and key estimation.

3.1.1 Beat Tracking Error Detection

We deliberately introduce a certain proportion of errors to the ground-truth beat annotations and ask the LLM to output the *beat index range* containing beat errors based on the music performance data in the symbolic music format. We introduce three types of error on beat annotations: 1) insert an extra beat between adjacent beats; 2) delete a beat; and 3) offset the timing of one beat, where the offset should be greater than a 70ms tolerance [18]. In beat tracking tasks, error detection is not a binary classification problem per detected beat, because there are false negative predictions (i.e., missed beats error). Therefore, it is crucial to return the beat index range so that both false positive beats and missed beats can be captured.

As shown in Figure 1 (left), the music segment and the beat annotations with errors are provided in the JSON format. The notes and beats are sorted by the temporal positions (i.e., onsets or beat locations).

3.1.2 Chord Extraction Error Detection

We deliberately introduce chord annotation errors and ask the LLM to output the *indices of incorrect chords* based on the music performance data in the symbolic music format. The chord errors are applied to the root, chord quality, and chord inversion attributes independently at a controlled rate.

As shown in Figure 1 (middle), we use the JSON format to represent the music segment and the chord annotations. The notes and chords are sorted by their temporal positions, and chords are notated as chord symbols in the conventional format [19].

3.1.3 Key Estimation Error Detection

We deliberately introduce key annotation errors and ask the LLM to output “correct” or “incorrect” based on the

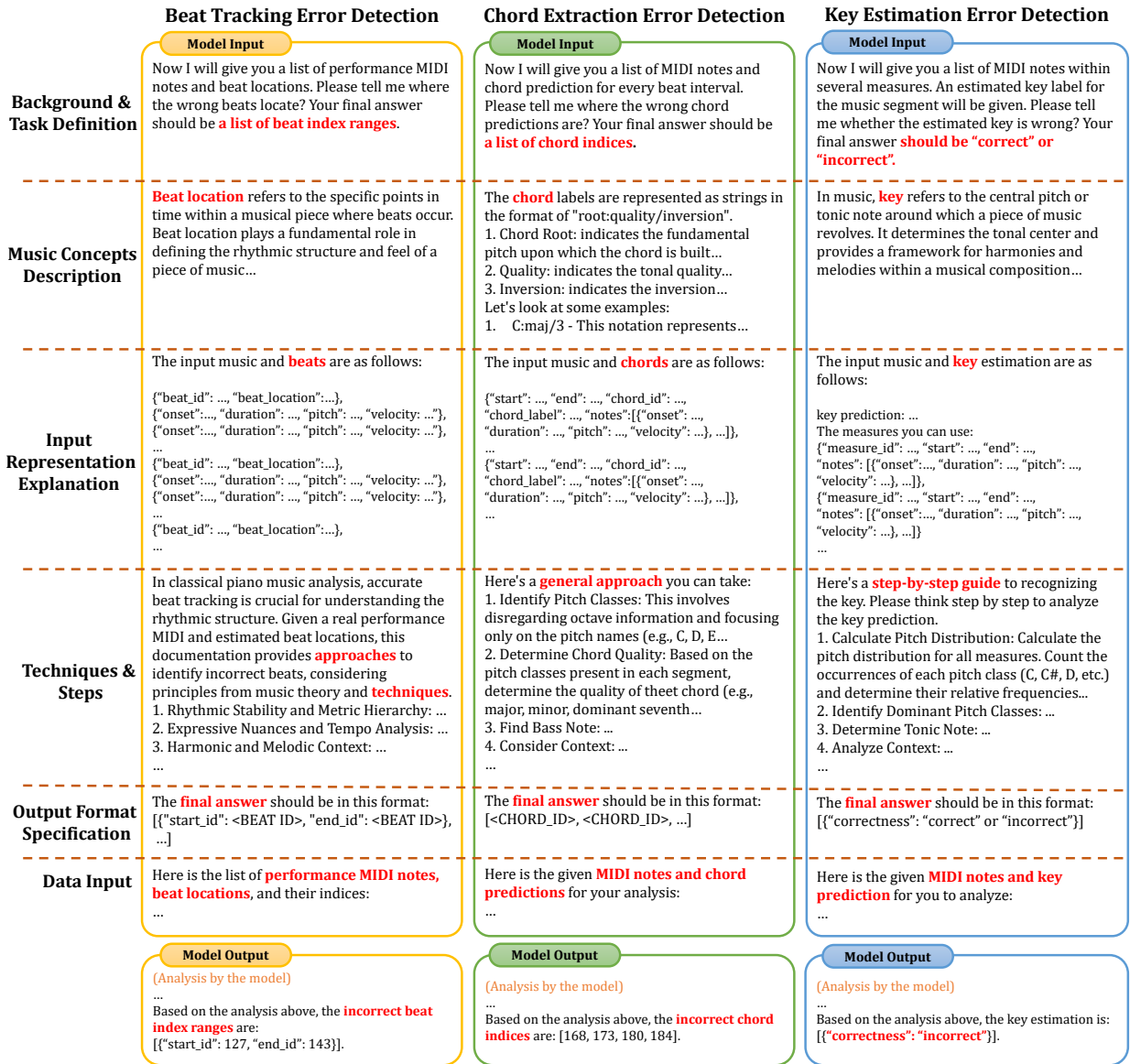


Figure 1: The example prompts and model outputs for the three error detection MIR tasks: beat tracking, chord extraction, and key estimation. Some keywords are highlighted in red in this figure for better readability. Orange texts indicate omitted content. The prompt structure is shown on the left.

music performance data in symbolic format. The errors are introduced by selecting an incorrect key out of the other 23 major and minor keys randomly at a controlled rate.

As shown in Figure 1 (right), we use the JSON format to represent the music segment and key annotations, where the key annotation is given at the beginning. The predicted key is represented by a formatted string of tonic and mode (e.g., "A:min").

3.2 Prompt Structure

Our investigation of prompt engineering methods indicates that a well-organized prompt structure is essential for successful MIR error detection. As shown in Figure 1, the prompt of the three MIR tasks all consists of six components as follows:

- **Background and Task Definition** introduces the MIR task and music domain background, and specifies the

role of the LLM as a judge in assessing the correctness of MIR results.

- **Music Concepts Description** introduces relevant music concepts about beat, chord, or key, together with examples of those concepts. For example, we show examples of chord root, quality, and inversion for chord extraction, to guide the LLM to better parse the chord labels such as C:maj/3.
- **Input Representation Explanation** specifies the data structure and format of the input music data.
- **Techniques and Steps** provides reference steps to encourage the LLM to apply "chain-of-thought" in the error detection process. For example, we provide clear steps in the chord extraction task: 1) Identify Pitch Classes; 2) Determine Chord Quality; 3) Find Bass Note; 4) Consider Music Context; and etc.
- **Output Format Specification** defines the JSON-like output format, ensuring consistency for post-processing.

- **Data Input** provides the subject music piece and the MIR results to be judged, following the format defined in input representation explanation section.

3.3 Concept Augmentation

The prompts defined in Section 3.2 contain extensive music concepts for each of the three tasks, which we regard as *Basic Concepts*. Based on these, we apply *concept augmentation* by either introducing new concepts or masking basic concepts in order to compare the LLM performance under varying amounts of music knowledge provided.

In *Concept Introduction*, we add new concepts that are supposedly helpful for doing MIR tasks. For example, for beat tracking, we introduce “rhythm” to the LLM: we provide a brief description of on-beat notes and off-beat notes, and how to compute their density percentages. We explain how such concepts contribute to better judgments.

Conversely, we also define the *Concept Masking* operation, which eliminates or blurs music concepts at different levels. Such operations are used to examine the *innate* reasoning ability of LLMs as a reference:

- **Music Attribute Masking:** removes explanations about music concepts pertaining to the musical objects under operations. For example, in chord extraction, “root note”, “chord quality”, and “inversion” are replaced by an abstract generic expression, “a chord feature”.
- **Task Masking:** on top of Music Attribute Masking, removes explanations about all concepts related to the MIR task, so that the LLM is required to reason about the correctness for a novel abstract task. For example, for beat tracking, the task becomes “Please read a sequence of MIDI notes and music labels to determine the correctness of each label.” All expressions that imply beat tracking will be deleted, including “tempo”, “fast”, “slow”, etc., to ensure that the task information is not implied in any form.
- **Domain Masking:** on top of Task Masking, eliminates explanations about all concepts related to the *music* domain to the greatest extent, leaving the LLM with an abstract logic-domain reasoning problem. For example, the LLM is told: “You will be given some labels and the corresponding raw data. Your task is to tell me where the wrong labels are located?”

4. EXPERIMENTS

We conduct our LLM-based MIR tasks with GPT-3.5. We introduce the datasets in Section 4.1 and the evaluation metrics in Section 4.2. The evaluation results are provided in Section 4.3.

4.1 Datasets

We use symbolic performance MIDI dataset for the three proposed tasks. For beat tracking error detection, we use classical piano recordings from the MAPS database [20], specifically from the “ENSTDkCI” subset (29 pieces) which has been commonly used as a beat tracking test

set. We also use the corresponding metrical annotations from [21]. We randomly create beat errors by inserting (9%), deleting (12%), or offsetting (9%) beats, resulting in an emulated MIR prediction with an F-score of 0.8370.

For both chord extraction and key estimation error detection, we use MIDI for Chinese pop songs on a subset of the POP909 dataset [22]. For chord extraction, we randomly introduce errors in root, quality, or inversion with a ratio of 30%, resulting in an “MIR” accuracy of 0.7327. We choose 50 songs and divide each song into segments with 32 chord labels. For key estimation, we test on 757 songs in the dataset whose ground-truth key is unchanged throughout the piece. We randomly select three four-measure segments for each song and modify 30% of the key labels at random. A summary of the data statistics is shown in Table 1.

	Beat Tracking	Chord Extraction	Key Estimation
#Notes	70,607	48,919	177,535
#Labels	14,194	9,200	2,271
#Tokens (per call)	6065.31	9256.80	3214.63

Table 1: Statistics of the music data used for evaluation. The row #Notes represents the total number of MIDI notes processed for each task. The row #Labels indicates the number of labels used in the evaluation of each task. The row #Tokens (per call) shows the average number of tokens per call fed into the GPT-3.5 model for each task.

4.2 Evaluation Metrics

We design metrics to evaluate the performance of LLMs in identifying errors in MIR annotations. Since our approach does not directly predict MIR annotations, our metrics differ from existing MIR evaluation metrics. For chord extraction and key estimation tasks, we regard error detection as a binary classification task in which each chord or key label is classified as correct or incorrect. We use weighted precision, recall, and F1-score to evaluate GPT’s performance on both correct and incorrect classes [23].

In beat tracking error detection, the beat sequence with potential errors is typically not one-to-one aligned with the ground truth beats. We consider three types of beat locations: 1) correctly identified beats, 2) additional beats, and 3) missing beats, which are also referred to as true positives, false positives, and false negatives, respectively, in conventional beat tracking tasks [18]. We use TP, FP, and FN to denote these sets of beat positions and I to denote the union of time intervals predicted by an LLM error detector. We consider the following metrics:

- **CPR (Correct Pass Rate on TP)** is defined as $\frac{|TP-I|}{|TP|}$, which measures the proportion of true positives that are correctly identified (by GPT) as “correct beats”.
- **EDR_P (Error Detection Rate on FP)** is defined as $\frac{|I \cap FP|}{|FP|}$, which evaluates the proportion of false positives that are correctly identified (by GPT) as “incorrect beats”.
- **EDR_N (Error Detection Rate on FN)** is defined as: $\frac{|I \cap FN|}{|FN|}$, which evaluates the proportion of false negatives that are correctly identified (by GPT) as “incorrect beats”.

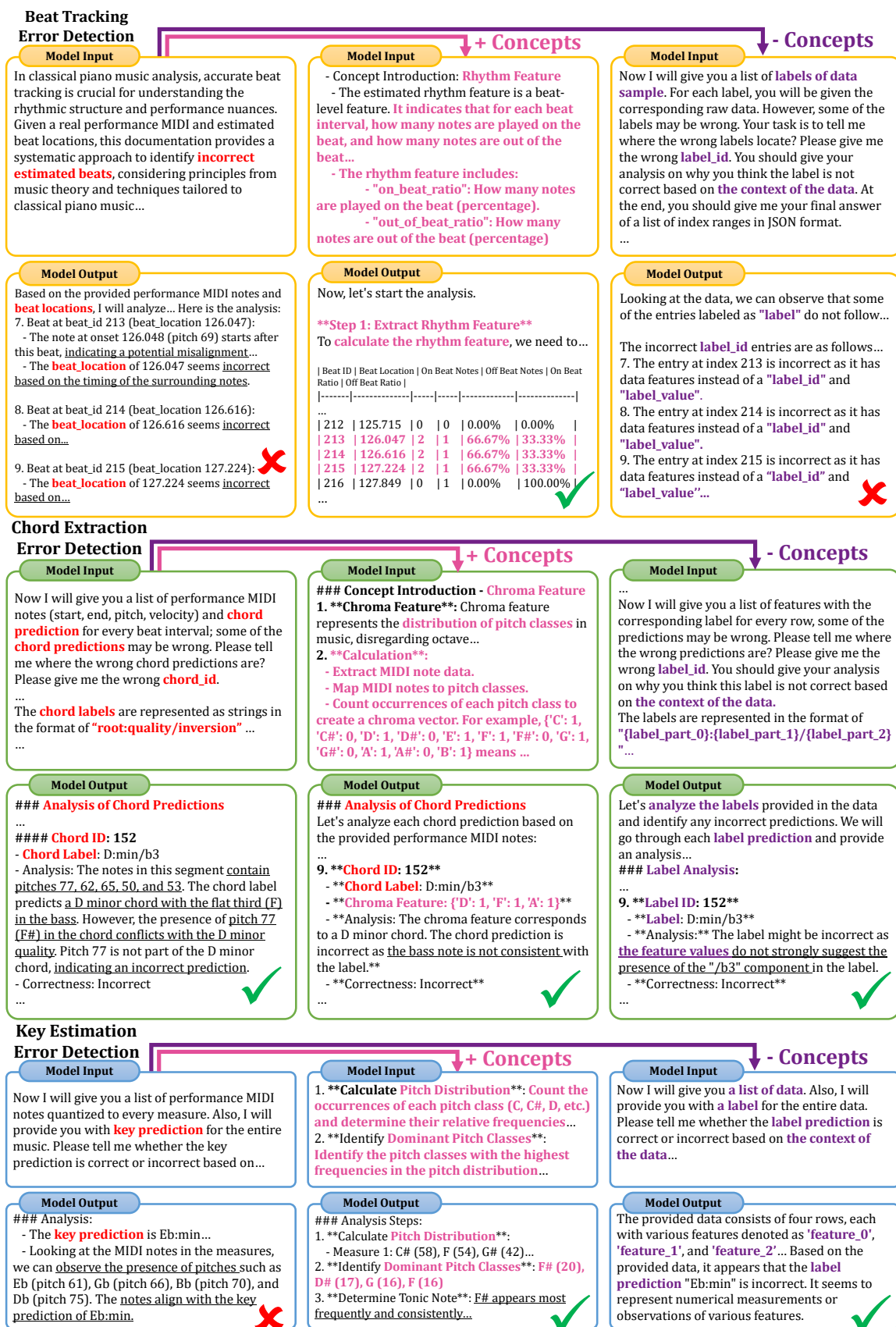


Figure 2: The impact of concept augmentation on GPT's behavior in three MIR error detection tasks: 1) *Basic Concepts* (left), 2) *Concept Introduction* (middle), and 3) *Concept Masking*: all music domain concepts removed (right). Red color indicates the basic concepts. Pink color indicates the introduced concepts. Purple color represents the expression after masking all music-related concepts. Underlines denote reasoning process. The checkmark indicates a correct judgment made by GPT, while the cross indicates an incorrect judgment by GPT.

Concept Augmentation	CPR \uparrow	EDR $_P$ \uparrow	EDR $_N$ \uparrow	WS \uparrow
Basic Concepts	0.6681	0.3728	0.1794	0.5607
+ "Rhythm"	0.8533	0.1496	0.0968	0.6520
- "Beat Location"(Music Attribute Masking)	0.6898	0.3720	0.2008	0.5792
- "Beat Tracking"(Task Masking)	0.5998	0.4010	0.2862	0.5296
- "Music"(Domain Masking)	0.2418	0.7657	0.7061	0.3785
Random	0.513 \pm 0.0586	0.4891 \pm 0.0564	0.3238 \pm 0.0608	0.4843 \pm 0.0274

(a) Evaluation results on beat tracking error detection

Concept Augmentation	p \uparrow	r \uparrow	f \uparrow
Basic Concepts	0.6345	0.6948	0.6207
+ "Chroma"	0.6996	0.7174	0.6290
- "Root"; "Quality"; "Inversion"(Music Attribute Masking)	0.6503	0.6992	0.6376
- "Chord Extraction"(Task Masking)	0.6497	0.6947	0.6362
- "Music"(Domain Masking)	0.6848	0.7144	0.6480
Random	0.5812 \pm 0.0032	0.5003 \pm 0.0034	0.5213 \pm 0.0033

(b) Evaluation results on chord extraction error detection

Concept Augmentation	p \uparrow	r \uparrow	f \uparrow
Basic Concepts	0.5789	0.6513	0.5965
+ "Scale"	0.5847	0.6169	0.5972
- "Tonic"; "Mode"(Music Attribute Masking)	0.5754	0.5812	0.5782
- "Key Estimation"(Task Masking)	0.5840	0.6143	0.5960
- "Music"(Domain Masking)	0.5927	0.4161	0.4085
Random	0.5779 \pm 0.0086	0.4977 \pm 0.0093	0.5186 \pm 0.0089

(c) Evaluation results on key estimation error detection

Table 2: The evaluation results of GPT on three MIR error detection tasks: beat tracking, chord extraction, and key estimation. Each task is assessed under different concept augmentation. "+" denotes *Concept Introduction*. "-" denotes *Concept Masking*. \uparrow indicates that higher values are better. p , r , and f stand for precision, recall, and F-score, respectively.

Finally, we compute a weighted average of these metrics, denoted by WS:

$$WS = \frac{CPR \times |TP| + EDR_P \times |FP| + EDR_N \times |FN|}{|TP| + |FP| + |FN|} \tag{1}$$

4.3 Evaluation Results

We evaluate the performance of GPT on three MIR error detection tasks. We first use the prompt with Basic Concepts and compare it with a random baseline, as well as prompts under different concept augmentation methods (see Section 3.3). The results are summarized in Table 2.

The results of beat tracking error detection task are shown in Table 2a. The random baseline is implemented by first randomly selecting k beat labels and joining consecutively selected beats into time intervals serving as detected error ranges. In Concept Introduction, we guide the GPT to compute the number of on-beat and off-beat note percentages, and in Concept Masking, we apply music attribute, task, and domain masking incrementally. Results show the basic prompt outperforms the random baseline in all prompt settings. Moreover, as the number of concepts decreases, the performance of GPT in judging the correctness of beat labels shows an overall downward trend.

The results of chord extraction error detection task are shown in Table 2b. The random baseline detects incorrectness with a probability of 50%. In Concept Introduction, we show GPT the chord chroma concept and encourage GPT to deduce the pitch distribution from input music. Results show that all GPT settings far exceed the random baseline. There remains a downward trend as the number concepts decreases except in the Domain Masking setting.

The results of key estimation error detection task are shown in Table 2c. The random baseline and concept aug-

mentation are implemented similarly to those of chord extraction. In Concept Introduction, we show GPT the scale concept. Results show that GPT performs slightly better than the random baseline in F-score and recall, and similar to the baseline in precision. The downward trend of concept augmentation is less salient.

Finally, we provide a case study (Figure 2) to illustrate GPT’s behavior under different settings of concept augmentation. In all tasks, GPT exhibits general time series analysis abilities even when music concepts are all masked, and the introduced music concepts help GPT to reason in a more musical fashion, particularly in beat tracking. However, we also observe limitations, including high randomness in output, sensitivity to prompts, and hallucination [24]. These issues make it challenging to empirically summarize or conjecture GPT’s reasoning abilities in solving MIR problems in general.

5. CONCLUSION AND FUTURE WORK

In conclusion, we have proposed a methodology to solve MIR problems with text-based LLMs with prompt engineering. We evaluate the performance of GPT-3.5 in error detection across three MIR tasks and find out that GPT’s music reasoning ability in MIR tasks can be enhanced when provided with well-structured prompts with music concepts. Across all three MIR error detection tasks, GPT consistently outperforms random baseline methods and demonstrates improved performance when prompted with additional music knowledge. In this study, we establish a baseline for assessing LLMs’ ability to understand music solely through reasoning, paving the way for future LLM-based MIR research. In the future, we will consider evaluating LLMs’ judging ability on real MIR errors instead of synthetic ones and using fine-tuning techniques to better explore LLM-based MIR study.

6. ACKNOWLEDGMENTS

This research has been supported by the Social Sciences and Humanities Research Council of Canada (SSHRC 895-2022-1004) and the China Scholarship Council.

7. REFERENCES

- [1] R. Yuan, H. Lin, Y. Wang, Z. Tian, S. Wu, T. Shen, G. Zhang, Y. Wu, C. Liu, Z. Zhou *et al.*, “Chatmusician: Understanding and generating music intrinsically with llm,” *arXiv preprint arXiv:2402.16153*, 2024.
- [2] L. Yu, Y. Cheng, Z. Wang, V. Kumar, W. Macherey, Y. Huang, D. Ross, I. Essa, Y. Bisk, M.-H. Yang *et al.*, “Spae: Semantic pyramid autoencoder for multimodal generation with frozen llms,” *Advances in Neural Information Processing Systems*, vol. 36, 2024.
- [3] D. Zhu, J. Chen, X. Shen, X. Li, and M. Elhoseiny, “Minigpt-4: Enhancing vision-language understanding with advanced large language models,” *arXiv preprint arXiv:2304.10592*, 2023.
- [4] S. Wu, H. Fei, L. Qu, W. Ji, and T.-S. Chua, “Nextgpt: Any-to-any multimodal llm,” *arXiv preprint arXiv:2309.05519*, 2023.
- [5] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou *et al.*, “Chain-of-thought prompting elicits reasoning in large language models,” *Advances in neural information processing systems*, vol. 35, pp. 24 824–24 837, 2022.
- [6] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altschmidt, S. Altman, S. Anadkat *et al.*, “Gpt-4 technical report,” *arXiv preprint arXiv:2303.08774*, 2023.
- [7] W.-L. Chiang, Z. Li, Z. Lin, Y. Sheng, Z. Wu, H. Zhang, L. Zheng, S. Zhuang, Y. Zhuang, J. E. Gonzalez *et al.*, “Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality,” See <https://vicuna.lmsys.org> (accessed 14 April 2023), vol. 2, no. 3, p. 6, 2023.
- [8] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar *et al.*, “Llama: Open and efficient foundation language models,” *arXiv preprint arXiv:2302.13971*, 2023.
- [9] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Défossez, “Simple and controllable music generation,” in *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., 2023.
- [10] L. Lin, G. Xia, J. Jiang, and Y. Zhang, “Content-based controls for music large language modeling,” *arXiv preprint arXiv:2310.17162*, 2023.
- [11] S. Liu, A. S. Hussain, C. Sun, and Y. Shan, “Music understanding llama: Advancing text-to-music generation with question answering and captioning,” *CoRR*, vol. abs/2308.11276, 2023. [Online]. Available: <https://doi.org/10.48550/arXiv.2308.11276>
- [12] H. Zhang, X. Li, and L. Bing, “Video-llama: An instruction-tuned audio-visual language model for video understanding,” *arXiv preprint arXiv:2306.02858*, 2023.
- [13] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, “Language models are few-shot learners,” *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.
- [14] D. Zhou, N. Schärli, L. Hou, J. Wei, N. Scales, X. Wang, D. Schuurmans, C. Cui, O. Bousquet, Q. Le *et al.*, “Least-to-most prompting enables complex reasoning in large language models,” *arXiv preprint arXiv:2205.10625*, 2022.
- [15] X. Wang, J. Wei, D. Schuurmans, Q. Le, E. Chi, S. Narang, A. Chowdhery, and D. Zhou, “Self-consistency improves chain of thought reasoning in language models,” *arXiv preprint arXiv:2203.11171*, 2022.
- [16] Y. Zhou, A. I. Muresanu, Z. Han, K. Paster, S. Pitis, H. Chan, and J. Ba, “Large language models are human-level prompt engineers,” *arXiv preprint arXiv:2211.01910*, 2022.
- [17] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao, “React: Synergizing reasoning and acting in language models,” *arXiv preprint arXiv:2210.03629*, 2022.
- [18] M. E. Davies, N. Degara, and M. D. Plumbley, “Evaluation methods for musical audio beat tracking algorithms,” *Queen Mary University of London, Centre for Digital Music, Tech. Rep. C4DM-TR-09-06*, 2009.
- [19] C. Harte, M. B. Sandler, S. A. Abdallah, and E. Gómez, “Symbolic representation of musical chords: A proposed syntax for text annotations.” in *ISMIR*, vol. 5, 2005, pp. 66–71.
- [20] V. Emiya, R. Badeau, and B. David, “Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 6, pp. 1643–1654, 2009.
- [21] A. Ycart, E. Benetos *et al.*, “A-maps: Augmented maps dataset with rhythm and key annotations,” 2018.

- [22] Z. Wang, K. Chen, J. Jiang, Y. Zhang, M. Xu, S. Dai, X. Gu, and G. Xia, “Pop909: A pop-song dataset for music arrangement generation,” *arXiv preprint arXiv:2008.07142*, 2020.
- [23] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and Édouard Duchesnay, “Scikit-learn: Machine learning in python,” *Journal of Machine Learning Research*, vol. 12, no. 85, pp. 2825–2830, 2011. [Online]. Available: <http://jmlr.org/papers/v12/pedregosa11a.html>
- [24] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. J. Bang, A. Madotto, and P. Fung, “Survey of hallucination in natural language generation,” *ACM Comput. Surv.*, vol. 55, no. 12, mar 2023. [Online]. Available: <https://doi.org/10.1145/3571730>

TOWARDS ASSESSING DATA REPLICATION IN MUSIC GENERATION WITH MUSIC SIMILARITY METRICS ON RAW AUDIO

Roser Batlle-Roca¹ Wei-Hsiang Liao² Xavier Serra¹
Yuki Mitsufuji³ Emilia Gómez^{1,4}

¹ Music Technology Group, Universitat Pompeu Fabra, Spain ² Sony AI, Japan ³ Sony AI, USA
⁴ Joint Research Centre, European Commission, Spain

roser.batlle@upf.edu


ABSTRACT

Recent advancements in music generation are raising multiple concerns about the implications of AI in creative music processes, current business models and impacts related to intellectual property management. A relevant discussion and related technical challenge is the potential replication and plagiarism of the training set in AI-generated music, which could lead to misuse of data and intellectual property rights violations. To tackle this issue, we present the Music Replication Assessment (MiRA) tool: a model-independent open evaluation method based on diverse audio music similarity metrics to assess data replication. We evaluate the ability of five metrics to identify exact replication by conducting a controlled replication experiment in different music genres using synthetic samples. Our results show that the proposed methodology can estimate exact data replication with a proportion higher than 10%. By introducing the MiRA tool, we intend to encourage the open evaluation of music-generative models by researchers, developers, and users concerning data replication, highlighting the importance of the ethical, social, legal, and economic consequences. Code and examples are available for reproducibility purposes.¹

1. INTRODUCTION

Significant advancements in generative algorithms for digital art creation are challenging the role of artificial intelligence (AI) in artistic practices. Regarding generative AI in the music domain, there is an increasing discussion related to the use of computational tools in music creative processes [1], the effects on artists' work, existing listening experiences and business models, and the impacts on intellectual property (IP) management [2,3]. A key point is the potential replication and plagiarism of the training set in AI-generated music [3,4], which can lead to data misuse and IP violations.

¹ <https://github.com/roserbatlleroca/mira>

 © R. Batlle-Roca, W. Liao, X. Serra, Y. Mitsufuji and E. Gómez. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** R. Batlle-Roca, W. Liao, X. Serra, Y. Mitsufuji and E. Gómez, "Towards Assessing Data Replication in Music Generation with Music Similarity Metrics on Raw Audio", in *Proc. of the 25th Int. Society for Music Information Retrieval Conf.*, San Francisco, United States, 2024.

The inherent opaque nature of music generation models complicates tracing replications or references to training set samples in AI-generated music, limiting the interpretation of whether generated samples contain replicated fragments. In addition, diffusion models, one of the most popular generative AI architectures, tend to memorise and replicate training data [5–7]. Understanding the behaviour of these models has become critical to address legal issues [8], especially when dealing with data protected by IP rights. This is significant in the music domain as the vast majority of music is protected by authorship and copyright.

Despite multiple claims emphasising the importance of assessing music-generative algorithms, there is a lack of evaluation tools directly focused on detecting data replication based on raw audio. Considering this research gap, the present investigation is motivated by two main questions:

- Are audio-based music similarity metrics suitable to assess data replication in AI-generated music?
- Can we propose an open model-agnostic evaluation method and tool found on diverse audio-based music similarity metrics?

Thus, this work proposes assessing the effectiveness of five music similarity metrics² (four state-of-the-art widely-used and a novel one) in estimating exact data replication in music. We review the implications of potential data replication in AI-generated music (Section 2) and present our experimental setup, including the selected music similarity metrics and specific methodology to control and estimate exact data replication (Section 3). We analyse metrics' behaviour in different music materials (Section 4.1), aiming to assess later their data replication detection sensitivity (Section 4.2). The proposed methodology is implemented in tool MiRA (Music Replication Assessment), which computes music similarity between reference and target samples to obtain global and per-pair distances (Section 5). Finally, we discuss our research's insights, limitations and future perspectives (Section 6).

By introducing MiRA tool, we advance towards the assessment of data replication in AI-generated music using similarity metrics, contributing to open evaluation methods for accessibility for researchers, developers and users. We strive to raise awareness, detect and prevent misappropriation of training sets, and hope to motivate research on these issues.

² Hereafter, music similarity metrics refer to audio-based metrics.

2. BACKGROUND AND RELATED WORK

2.1 Implications of potential data replication in AI-generated music

Music-generative AI is advancing rapidly with novel high-quality models driven by a strong push from the industry, which is encouraging a suitable environment for real-world deployment. Yet, music generation algorithms bring significant concerns regarding their ethical, social, legal and economic implications. A key challenge is the potential data replication in AI-generated music—inquiring whether a generative model extracts and copies fragments from the training data and whether AI-generated music can be considered novel and original [3, 4]. This issue is further complicated by the implications derived concerning data misuse and IP violations such as copyright infringement. Moreover, diffusion models, one of the most popular architectures for generative AI, present high risks of data replication as they have shown a tendency to memorise their training data [5–7]. In the image generation domain, Somepalli et al. [9] demonstrate instances where generated images with diffusion models contain object-level copies of their training data. Based on image retrieval frameworks, they compare generated images with training samples and detect when content has been replicated. Similarly, Carlini et al. [5] demonstrate that diffusion models memorize and reproduce images from their training data.

Memorising training data and potential IP violations is highly under-discussed in music generative models literature, despite being one of generative AI’s main negative ethical implications in the music domain [10]. However, the recently proposed music generative model *MusicLM* [11] has been refrained from releasing due to the ethical risks and potential work replication. In addition, *MusicLDM* [12] acknowledges potential issues linked to data replication and plagiarism and, to address them, proposes two beat-synchronous mix-up strategies for data augmentation. The exemplified initiatives underscore the relevance of considering and addressing the ethical implications of these algorithms.

2.2 Evaluation methodologies in music generation

Xiong et al. [13] present a survey on music generation evaluation methodologies divided into objective, subjective and combined approaches. They highlight a current claim in finding a standardised proper method that aligns with all stakeholders, from developers to musicians and music listeners. However, even if multiple evaluation methodologies exist for music generation models, the literature highlights a lack of evaluation methodologies focused on assessing data replication and the originality of AI-generated music [4, 14]. In the symbolic domain, Yin et al. [4] introduce the *originality score* to measure the extent to which an algorithm might be copying from the training set. Nonetheless, there is a growing interest in models outputting directly audio music instead of symbolic representations. Thus, a research gap exists in detecting data replication in AI-generated music based on raw audio.

A recent work by Barnett et al. [15] proposes a framework based on two music audio embeddings to assess the similarity between the training data and AI-generated samples for understanding training data attribution. Their approach, based on VampNet [16], computes cosine distance on embeddings obtained from CLMR (Contrastive Learning of Musical Representations) [17] and CLAP (Contrastive Language-Audio Pretraining) [18].

Our perspective is that combining metrics based on audio embeddings, acoustic qualities, and features capturing music characteristics, such as chord progression or tonal similarity, provides a comprehensive assessment of potential data replication in AI-generated music. In this study, we aim to validate the effectiveness of five music similarity metrics and build an open tool to assess exact data replication in AI-generated music using these metrics.

3. FORCED-REPLICATION EXPERIMENT

3.1 Audio Music Similarity Metrics

For this study, we consider five music similarity metrics: four state-of-the-art approaches and a novel one, covering a diversity of characteristics. We here describe the metrics (summarised in Table 1) and methods used to implement them.³

Cover Song Identification (CoverID) [19–21]: Cover song identification is a task aiming to detect whether two music recordings are based on the same composition, accounting for variations in tempo, structure, and instrumentation while keeping a similar melodic or harmonic line. CoverID relies on pitch-content features and local alignment. To obtain CoverID distance, we use the implementation available in Essentia.⁴ A low CoverID value suggests substantial composition similarity between the two analysed music samples.

Kullback-Leibler (KL) divergence: This metric provides a non-symmetric statistical measurement between reference and target probability distributions relative to their entropy. KL divergence has been employed in the literature to estimate similarity in music (e.g. [22, 23]), and more recently, to assess automatic music generation prompt adherence (e.g. [24]). We aim to explore its capabilities to estimate data replication in music samples. To obtain probability distributions, we use the PaSST audio classifier proposed in Koutini et al. [25], trained on Audioset. This methodology aligns with common practice in the literature, such as in *AudioGen* [26] and *MusicGen* [27] to obtain the probabilities of the labels in their audio and music samples. To avoid the non-symmetry of KL divergence, we compute reference to target and target to reference KL divergence and, subsequently, average both results to obtain symmetric KL divergence. Low KL divergence indicates a closer similarity between distributions.

³ Two of the metrics rely on Essentia implementation. Essentia is an open-source library and tools for audio and music analysis, description and synthesis, developed in the Music Technology Group at Universitat Pompeu Fabra: <https://essentia.upf.edu>.

⁴ https://essentia.upf.edu/reference/std_CoverSongSimilarity.html

Table 1: Summary of the considered music similarity metrics, indicating whether values correspond to higher or lower similarity (↓/↑).

Metric	Description
CoverID (↓)	Musical composition similarity based on music-specific characteristics.
KL divergence (↓)	Differences in distributions from an audio classifier.
CLAP (↑)	Distance between embeddings from a music pre-trained model.
DEfNet (↑)	Novel metric based on distance between embeddings from a contrastive learning model for music similarity.
FAD (↓)	Distance between embeddings based on CLAP music model.

Contrastive Language-Audio Pretraining (CLAP) score [18]: CLAP embeddings⁵ allow to obtain latent representations of audio or text by conditioning information. For instance, *MusicLDM* [12] uses this metric to assess the novelty in text-to-music generations. To compute the CLAP score between two music samples, we extract the audio embeddings from the pre-trained music model⁶ for each one and compute the cosine distance between them. A high CLAP score indicates a high similarity between the two music samples.

Discogs-EffNet (DEfNet) score: In addition to state-of-the-art distances between audio embeddings, we incorporate a novel approach based on Essentia models [28]. Essentia’s Discogs-EffNet model⁷ provides music audio embeddings trained on Discogs metadata with contrastive learning purposes for music similarity. We consider DEfNet score to observe the effectiveness of embeddings of a model trained for a music-related task on estimating data replication. Embeddings are extracted based on track self-supervised annotations⁸ and compute the cosine distance between reference and target samples. A high DEfNet score reveals high track similarity.

Fréchet Audio Distance (FAD) [29, 30]: FAD is an adaptation of Fréchet Inception Distance (FID) for music, comparing embedding distributions of a reference and a target set, based on the ViGGish model [31]. Nonetheless, a recent study by Gui et al. [30] questions whether ViGGish is the optimal model for FAD computation for music generation evaluation. They propose a tool kit⁹ with multiple models to obtain more accurate embeddings to assess AI-generated music when calculating FAD. Consequently, we implement the adapted version of FAD using the CLAP audio music pre-trained model. A low FAD score indicates a high resemblance between the compared music samples.

⁵ <https://github.com/LAION-AI/CLAP>

⁶ Checkpoints: music_audioset_epoch_15_esc_90.14.pt.

⁷ <https://essentia.upf.edu/models.html#discogs-effnet>

⁸ Embeddings extracted with weights `discogs_track_embeddings-effnet-bs64-1.pb`.

⁹ <https://github.com/microsoft/fadtk>

3.2 Experimental Approach

To validate the effectiveness of the selected music similarity metrics in detecting exact data replication, we carried out a controlled forced-replication experiment with synthetic data, i.e. replicating music excerpts into another song under controlled conditions. Synthetic data guaranteed that the analysed music samples contained copied instances, limiting our scope to exact data replication.

For this experiment, we use an in-house dataset of 30-second audio previews from the Spotify API¹⁰, composed of over 18,000 samples and 24 music genre classes. We focus on six music genre classes defined by Spotify API internal class labels: *heavy metal*, *afrobeats*, *techno*, *dub*, *cumbia* and *bolero*. These genres were chosen for their diverse musical compositions and elements, allowing us to examine the metrics across multiple scenarios. This selection was supported using ChatGPT, which affirmed that these genres have distinct musical characteristics.

We divide data into three groups: (1) **reference set:** acting as training data, (2) **target set:** composed of synthetic data, representing AI-generated music, and (3) **mixture set:** containing different songs from the reference set but from the same music genre to build synthetic data. Synthetic data with replication contains a controlled percentage of copy from a song in our reference set: 5% (1.5s), 10% (3s), 15% (4.5s), 25% (7.5s) and 50% (15s). A synthetic sample is created by introducing the copied proportion at a random point of a music sample in the mixture set. We create 10 samples with a proportion of replication per song in the reference set. Figure 1 illustrates the procedure to build synthetic data with 5% of replication. For each music genre, the reference and mixture sets are composed of 400 songs each. Thus, the target set comprises 4,000 (400 x 10) songs per percentage of replication for each genre. Music samples are 30 seconds long as currently it is the common length in full song composition music generative models.

We assess each metric for all the songs within the reference set against themselves to establish a baseline (400 x 400 = 160,000 per-pair evaluations). Then, we compute them for each reference song and its copied instances to only consider cases with exact data replication (4,000 per-pair evaluations). Our experiment considers 120,000 samples of synthetic data (approximately 167h of music with a proportion of data replication).

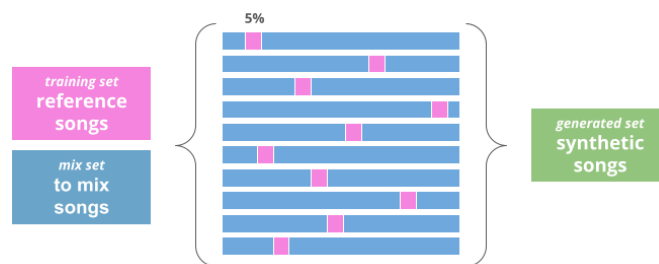


Figure 1: Synthetic data procedure with 5% of replication.

¹⁰ <https://developer.spotify.com/documentation/web-api>

4. RESULTS

4.1 Analysing metric behaviour

Figures 2, 3, 4, 5 and 6 depict the average μ and standard deviations σ of the different metrics per degree of replication and music genre. We observe a steady and similar behaviour by three metrics (CoverID, CLAP and DEfNet) through all studied music genres, showing higher similarity values for cases with higher replication levels (i.e., 50%). Standard deviation decreases with increasing replication level, which suggests less disparity within the analysed pairs. These three metrics show the sensitivity¹¹ to estimate data replication.

Instead, KL divergence presents a different behaviour with very similar values of μ and σ for all degrees of replication. Some sensitivity is observed in all music genres, except for *dub*, where the baseline mean μ_b is smaller than in replication cases μ_r , despite the standard deviation being higher ($\mu_b=0.757$, $\sigma_b=0.511$; $\mu_r=0.862$, $\sigma_r=0.462$). Thus, KL divergence demonstrates the capability of detecting replication but is ineffective in distinguishing between degrees of replication.

Contrasting with the other metrics, FAD based on CLAP music embeddings completely differs from them. On the one side, its behaviour is inconsistent as it exhibits fluctuating trends for the different examined cases. On the other side, it fails to detect data replication. A higher similarity value (low FAD) is always obtained for the baseline. Instead, for the different degrees of replication, higher FAD is achieved. Consequently, FAD based on CLAP music embeddings does not appear to be a suitable metric to assess exact data replication in music samples.

By analysing the metrics' behaviour, we could directly conclude that CoverID, KL divergence, CLAP and DEfNet are suitable for our posed research aim. However, further exploration is required before determining their ability to detect replication and degree of replication. We delve into this analysis in the next subsection.

4.2 Assessing data replication detection sensitivity

In this section, we complement the previous analysis with an assessment of statistical differences. Because our data is not normally distributed and variance is heterogeneous, the Kruskal-Wallis test [32] is the most adequate statistical analysis to examine our results, as is non-parametric, does not rely on normality and handles unequal sample sizes. We perform the Kruskal-Wallis test on CoverID, KL divergence, CLAP and DEfNet. Significant statistical differences ($p < 0.05$) are observed across all music genres and degrees of replication, consistent with our earlier findings.

Nonetheless, the insight of this analysis relies on the pairwise comparisons between the baseline and different degrees of replication. CoverID pairwise comparison reveals a statistically significant difference between the baseline and the 5% replication degree for *afrobeat*, *cumbia* and *techno*. For the three other music genres, this happens

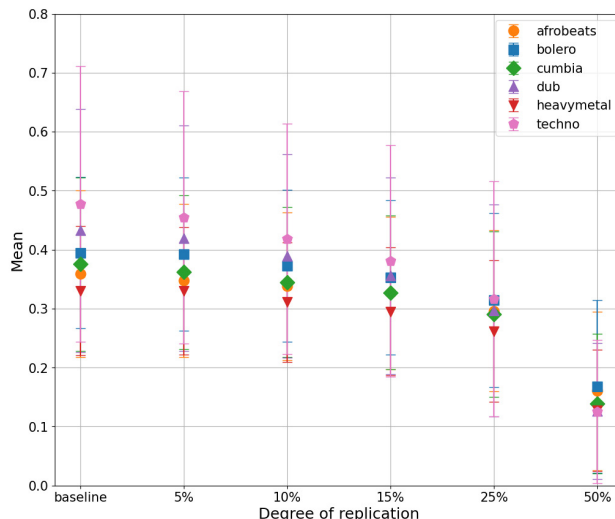


Figure 2: CoverID (↓)

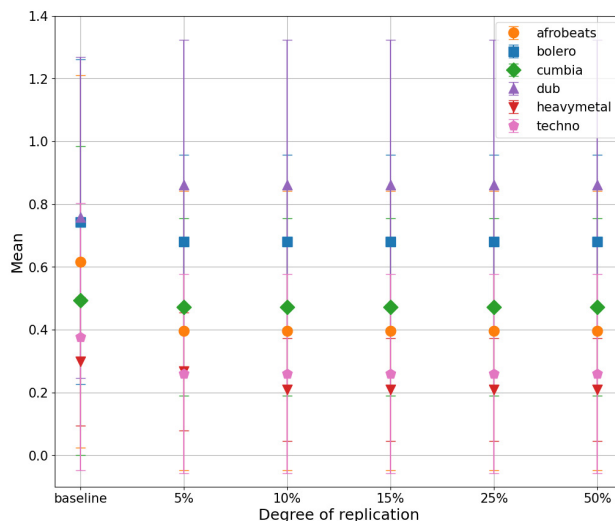


Figure 3: KL divergence (↓)

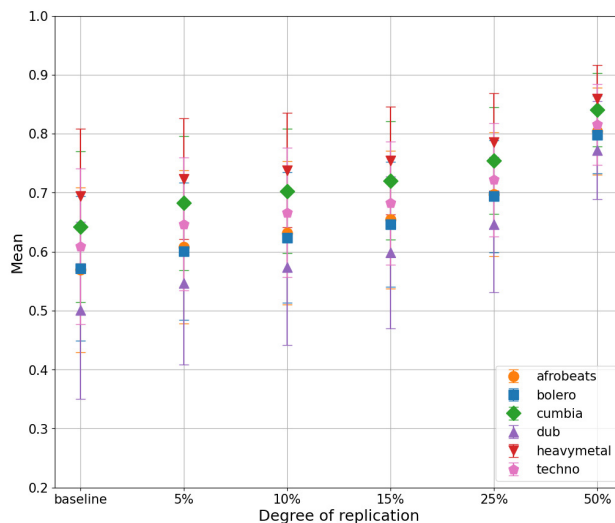


Figure 4: CLAP (↑)

¹¹ *Sensitivity* is understood as the capability to differentiate between degrees of replication.

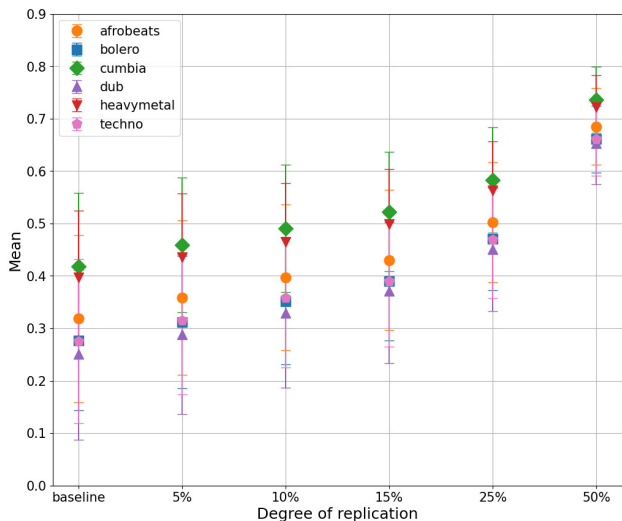


Figure 5: DEfNet (↑)

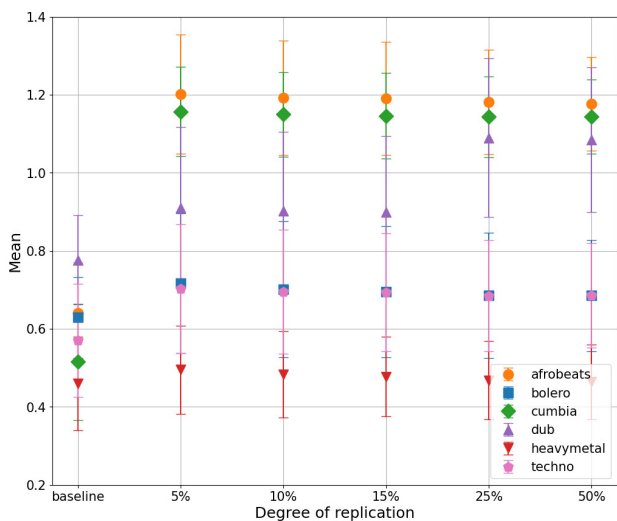


Figure 6: FAD (↓)

for a 10% replication degree. Then, statistical significance also appears in pairwise comparisons of different degrees of replication. We can derive that CoverID is sensible for 10% of replication, and in some cases at 5%. When considering KL divergence, pairwise comparison depicts a statistically significant difference between the baseline and the 5% replication degree. Between degrees of replication, no statistical significance is revealed for any pairwise comparison, except for *heavy metal* between 5% and the other replication degrees. Regarding the CLAP and DEfNet, a significant difference already appears when comparing the baseline against the samples with 5% replication, indicating that these metrics are sensitive to 1.5 seconds of replication. In all cases, a notable difference emerges among the levels of replication, enhancing the sensitivity of these metrics’ detection capabilities. They demonstrate sensitivity to varying replication degrees.

Withal, this statistical analysis sustains the validity of these four metrics to assess exact data replication in the training set and determines their degree of sensitivity.

5. MUSIC REPLICATION ASSESSMENT TOOL

Derived from the presented experiment, we implement the proposed methodology into an evaluation tool. We introduce the Music Replication Assessment (MiRA) tool: an open evaluation method based on four diverse raw audio music similarity metrics.

MiRA computes music similarity between reference and target samples to obtain global and per-pair distances, based on CoverID, KL divergence, CLAP and DEfNet. It can estimate data replication with a proportion higher than 10% (3 seconds), but in most of the examined scenarios, it is sensible to 5% of replication. Per-pair distances are highly beneficial for detecting close pairs, outliers and suspicious cases with potential data replication. Considering that replication detection requirements may vary depending on the evaluation, users are left to set their replication threshold. In addition, MiRA is model-independent as no information about the model architecture or its characteristics is necessary. The evaluation is conducted directly with the training (**reference**) and generated samples (**target**) of the analysed generative model.

However, designating a baseline value is encouraged to accurately interpret the music similarity between the reference and target samples. We propose a third comparison group of samples (**control**) based on songs related to the reference songs but unseen by the model (e.g. shared music genre). Again, this is a decision for the users conditioned to their evaluation scope. Note that using a control group allows us to understand and interpret the results obtained by acting as the baseline similarity level of independent songs with a shared characteristic.

The complete structure of the implemented system is depicted in Figure 7. We release MiRA as an open-source tool, built into a PyPI package¹². Together with the code, we provide examples and best practice recommendations for using this methodology. With the release of MiRA, we hope to enhance transparency in music generation models and data replication assessment.

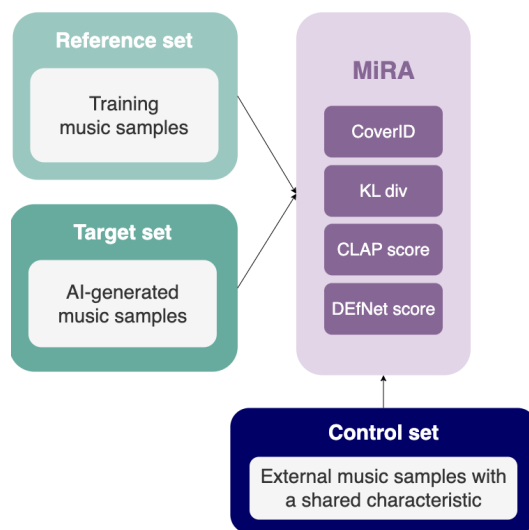


Figure 7: MiRA’s structure scheme.

¹² <https://pypi.org/project/mira-sim/>

6. DISCUSSION AND CONCLUSIONS

This work focused on validating the use of music similarity metrics for assessing data replication in AI-generated music. We hypothesise that similarity metrics are effective in estimating data replication. Therefore, we framed the scope of our study to exact data replication in music samples, while conducting a controlled forced-replication experiment with synthetic data.

We examined five diverse audio-based metrics: four standard metrics (CoverID, KL divergence, CLAP and FAD) and a novel one (DEfNet). Our results indicate that four of the five studied metrics can detect data replication to a certain extent. Instead, FAD based on CLAP music embeddings presented an opposite behaviour compared to the other metrics. Higher similarity is obtained for the baseline group and FAD shows unstable trends throughout the diverse music genres. Thus, we do not find it suitable for our case study. However, it must be acknowledged that the recent publication by Gui et al. [30] offered multiple classifiers to compute FAD. There is the possibility that we did not consider the appropriate classifier for our task. Thus, we should consider exploring other classifiers before determining the validity of FAD in detecting replication in music.

Regarding the other four metrics, our results show interesting insights. First, we find CoverID to be sensible to different replication degrees, establishing a robust threshold level at 10% of replication. Furthermore, in some of the studied cases, replication sensitivity is lowered to 5% of replication. This is a substantial finding to validate the suitability of metrics oriented to specific music characteristics, such as tempo, structure and composition.

Next, we observe that KL divergence can be sensitive to replication as pairwise comparison between baseline and degrees of replication is statistically significant. Nevertheless, the other pairwise results reveal that KL divergence is ineffective for differentiating between replication degrees. We consider this an unexpected turnout in our analysis.

Considering CLAP and DEfNet scores, both embedding-based metrics, our experiment validates their suitability to detect data replication. Not only do they show robustness by increasing their similarity value parallel to the replication degrees (i.e. higher similarity for higher level of replication), but they also show high sensitivity for different degrees of replication. All results suggest their sensitivity might be higher than we envisioned and might be able to detect replication in smaller samples (i.e. < 1.5 seconds).

As a result of these findings, we achieve our second goal within the scope of this research: to build an open model-agnostic tool based on music similarity metrics on raw audio. In this article, we have introduced the MiRA tool, leveraging the four validated similarity metrics, which can be used to evaluate any music-generative model with audio output. MiRA does not require any information about the model architecture or its characteristics. Instead, similarity evaluation relies on comparing reference and target samples.

By introducing the MiRA tool, we are contributing to the research gap of lack of evaluation methodologies directly assessing potential data replication in AI-generated music. Our study validates the use of similarity metrics to estimate training data replication. We intend to encourage the open evaluation of music generation models by researchers, developers and users concerning data replication. In addition, our research strives for the importance of ethical, social, legal and economic consequences of generative AI in the music domain, together with the need to address their risks and issues.

6.1 Limitations and Future Work

Despite our contribution to advance towards data replication assessment with music similarity metrics, there are multiple opportunities to complement our investigation.

First, we limited the scope of our experimental approach to assessing the use of different music similarity metrics for exact data replication, consequently reducing the definition of plagiarism to exact replication of fragments in the training set. We followed such an approach to validate our hypothesis and ensure an attainable method to address this issue. While this reduced scope could potentially be solved using audio fingerprinting strategies [33], we believe that by employing a diverse range of metrics we can provide a more comprehensive assessment of data replication.

Framing our aim to exact data replication also introduced a limitation in considering typical perturbations that music samples experience when training the model or during the model procedure to generate a music sample. Thus, it would be a key point for future work to validate the robustness of these metrics towards typical data augmentation techniques, such as pitch shifting and reverberation. Proving them to be robust would also enhance the capabilities of MiRA for detecting potential replication in AI-generated music. At the same time, we intend to expand the abilities of MiRA for data replication by incorporating complementary metrics, if necessary.

In addition, our experimental process was limited to the high computational costs of some of the metrics. In particular, we faced significantly large amounts of time to compute FAD and KL divergence. This is a relevant concern as we want MiRA to be an open tool that can be used by any researcher or user. Thus, considering the computational capacity required to compute the integrated metrics within is a relevant issue in our research.

Another limitation is the type of data that we use. We base our experiment on synthetic data despite our goal being oriented to AI-generated music. We must use synthetic data with a controlled percentage of replication to guarantee and assess the capabilities of detection and sensitivity of music similarity metrics. However, we would like to test the validity of the introduced tool when used in a generation context. To do so, we require not only a generative model but its details on training data and generation samples. We plan to expand our research in with AI-generated content in upcoming studies.

7. ACKNOWLEDGMENTS

The authors would like to thank Gaëtan Hadjeres and William Thong from Sony AI, as well as Dmitry Bogdanov and Pablo Alonso-Jiménez from the Music Technology Group at Universitat Pompeu Fabra, for their insightful discussions and valuable feedback throughout the development of this research.

8. ETHICS STATEMENT

The late rapid popularity growth of generative AI in the music domain brings significant ethical implications. The main challenges are linked to the role of AI within music creative processes, such as composition, potential misappropriation of data in AI-generated music, inquiries on the novelty of generations, derived authorship attribution, effects on intellectual property rights and sustainability of current business models. In addition, there are notable concerns about the cultural bias in these systems and their environmental impact.

Our research focused on the issue of assessing potential data replication in AI-generated music. We observed a lack of evaluation methodologies to examine replication in raw audio. We contributed to this issue by proposing a methodology based on audio-based music similarity metrics. We demonstrated its effectiveness and provided an open tool to evaluate AI-generated music. Our introduced approach is contributing to the transparency of music generation algorithms.

Despite the positive contribution of our investigation, we must be critical of some methodological aspects of our work. Our principal ethical concern falls under the type of data used to conduct our forced-replication experiment. In particular, we employ an internal dataset created with Spotify previews (30-second samples of music). Even if these practices are common in the ISMIR community, we see the need for guidelines for the legal assessment of MIR data included in datasets, incorporating country dependencies, origin and intended use, personal data involved (from artists and listeners) and potential future consequences¹³.

9. REFERENCES

- [1] F. Carnovalini and A. Rodà, “Computational creativity and music generation systems: An introduction to the state of the art,” *Frontiers in Artificial Intelligence*, vol. 3, 2020.
- [2] E. Gómez, M. Blaauw, J. Bonada, P. Chandna, and H. Cuesta, “Deep learning for singing processing: Achievements, challenges and impact on singers and listeners,” *ArXiv*, 2018. [Online]. Available: <https://arxiv.org/abs/1807.03046v1>
- [3] B. L. T. Sturm, M. Iglesias, O. Ben-Tal, M. Miron, and E. Gómez, “Artificial intelligence and music: Open questions of copyright law and engineering praxis,” *Arts*, vol. 8, p. 115, 2019.
- [4] Z. Yin, F. Reuben, S. Stepney, and T. Collins, “Measuring when a music generation algorithm copies too much: The originality report, cardinality score, and symbolic fingerprinting by geometric hashing,” *SN Computer Science*, vol. 3, 2022.
- [5] N. Carlini, J. Hayes, M. Nasr, M. Jagielski, V. Sehwag, F. Tramer, B. Balle, D. Ippolito, and E. Wallace, “Extracting training data from diffusion models,” in *32nd USENIX Security Symposium (USENIX Security 23)*, 2023, pp. 5253–5270.
- [6] N. Carlini, D. Ippolito, M. Jagielski, K. Lee, F. Tramer, and C. Zhang, “Quantifying memorization across neural language models,” *ArXiv*, 2023.
- [7] D. Bralios, G. Wichern, F. G. Germain, Z. Pan, S. Khurana, C. Hori, and J. L. Roux, “Generation or replication: Auscultating audio latent diffusion models,” *ArXiv*, 2023.
- [8] H. Wang, “Authorship of artificial intelligence-generated works and possible system improvement in china,” *Beijing Law Review*, vol. 14, pp. 901–912, 2023.
- [9] G. Somepalli, V. Singla, M. Goldblum, J. Geiping, and T. Goldstein, “Diffusion art or digital forgery? investigating data replication in diffusion models,” *ArXiv*, 2022.
- [10] J. Barnett, “The ethical implications of generative audio models: A systematic literature review,” *AIES 2023 - Proceedings of the 2023 AAAI/ACM Conference on AI, Ethics, and Society*, pp. 146–161, 2023.
- [11] A. Agostinelli, T. I. Denk, Z. Borsos, J. Engel, M. Verzetti, A. Caillon, Q. Huang, A. Jansen, A. Roberts, M. Tagliasacchi, M. Sharifi, N. Zeghidour, and C. Frank, “MusicLM: Generating music from text,” *ArXiv*, 2023.
- [12] K. Chen, Y. Wu, H. Liu, M. Nezhurina, T. Berg-Kirkpatrick, and S. Dubnov, “MusicLDM: Enhancing novelty in text-to-music generation using beat-synchronous mixup strategies,” *ArXiv*, 2023.
- [13] Z. Xiong, W. Wang, J. Yu, Y. Lin, and Z. Wang, “A comprehensive survey for evaluation methodologies of AI-generated music,” *ArXiv*, 2023.
- [14] R. Battle-Roca, E. Gómez, W. Liao, X. Serra, and Y. Mitsufuji, “Transparency in music-generative AI: A systematic literature review,” *Research Square preprint*, 2023.
- [15] J. Barnett, H. F. Garcia, and B. Pardo, “Exploring musical roots: Applying audio embeddings to empower influence attribution for a generative music model,” *arXiv*, 2024.

¹³ We refer to a recently documented example of research vs legal clash linked to algorithmic auditing in the music domain <https://www.rollingstone.com/pro/features/spotify-teardown-book-streaming-music-790174/>

- [16] H. F. Flores Garcia, P. Seetharaman, R. Kumar, and B. Pardo, “Vampnet: Music generation via masked acoustic token modeling,” in *Proceedings of the 24th International Society for Music Information Retrieval Conference, ISMIR 2023, Milan, Italy, 2023*.
- [17] J. Spijkervet and J. A. Burgoyne, “Contrastive learning of musical representations,” *Proceedings of the 22nd International Society for Music Information Retrieval Conference, ISMIR 2021, Online, 2021*.
- [18] Y. Wu, K. Chen, T. Zhang, Y. Hui, T. Berg-Kirkpatrick, and S. Dubnov, “Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation,” *ArXiv*, 2023.
- [19] J. Serrà, X. Serra, and R. Andrzejak, “Cross recurrence quantification for cover song identification,” *New Journal of Physics*, vol. 11, 2009.
- [20] J. Serrà, E. Gómez, P. Herrera, and X. Serra, “Chroma binary similarity and local alignment applied to cover song identification,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 6, pp. 1138–1151, 2008.
- [21] J. Serrà, E. Gómez, and P. Herrera, “Transposing chroma representations to a common key,” *IEEE CS Conference on The Use of Symbols to Represent Music and Multimedia Objects*, 2008.
- [22] M. Hoffman, D. Blei, and P. Cook, “Content-based musical similarity computation using the hierarchical dirichlet process,” in *Proceedings of the 9th International Society for Music Information Retrieval Conference, ISMIR 2008, Philadelphia, USA, 2008*.
- [23] D. Schnitzer, A. Flexer, G. Widmer, and M. Gasser, “Islands of gaussians: The self organizing map and gaussian music similarity features,” in *Proceedings of the 11th International Society for Music Information Retrieval Conference, ISMIR 2010, Utrecht, Netherlands, 2010*.
- [24] Z. Evans, C. Carr, J. Taylor, S. H. Hawley, and J. Pons, “Fast timing-conditioned latent audio diffusion,” *ArXiv*, 2024.
- [25] K. Koutini, J. Schlüter, H. Eghbal-zadeh, and G. Widmer, “Efficient training of audio transformers with patchout,” in *Interspeech 2022, 23rd Annual Conference of the International Speech Communication Association, Incheon, Korea. ISCA, 2022*, pp. 2753–2757.
- [26] F. Kreuk, G. Synnaeve, A. Polyak, U. Singer, A. Défossez, J. Copet, D. Parikh, Y. Taigman, and Y. Adi, “Audiogen: Textually guided audio generation,” 2023.
- [27] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Défossez, “Simple and controllable music generation,” *ArXiv*, 2023.
- [28] P. Alonso-Jiménez, D. Bogdanov, J. Pons, and X. Serra, “Tensorflow audio models in Essentia,” in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.
- [29] K. Kilgour, M. Zuluaga, D. Roblek, and M. Sharifi, “Fréchet Audio Distance: A Reference-Free Metric for Evaluating Music Enhancement Algorithms,” in *Proc. Interspeech 2019*, 2019, pp. 2350–2354.
- [30] A. Gui, H. Gamper, S. Braun, and D. Emmanouilidou, “Adapting frechet audio distance for generative music evaluation,” *ArXiv*, 2023.
- [31] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. J. Weiss, and K. Wilson, “Cnn architectures for large-scale audio classification,” *ArXiv*, 2017.
- [32] W. H. Kruskal and W. A. Wallis, “Use of ranks in one-criterion variance analysis,” *Journal of the American Statistical Association*, vol. 47, no. 260, pp. 583–621, 1952.
- [33] P. Cano and E. Batlle, “A review of audio fingerprinting,” *Journal of VLSI Signal Processing*, vol. 41, pp. 271–284, 11 2005.

GENERATING SAMPLE-BASED MUSICAL INSTRUMENTS USING NEURAL AUDIO CODEC LANGUAGE MODELS

Shahan Nercessian* Johannes Imort* Ninon Devis Frederik Blang
Native Instruments

firstname.lastname@native-instruments.com

*Equal contribution

ABSTRACT

In this paper, we propose and investigate the use of neural audio codec language models for the automatic generation of sample-based musical instruments based on text or reference audio prompts. Our approach extends a generative audio framework to condition on pitch across an 88-key spectrum, velocity, and a combined text/audio embedding. We identify maintaining timbral consistency within the generated instruments as a major challenge. To tackle this issue, we introduce three distinct conditioning schemes. We analyze our methods through objective metrics and human listening tests, demonstrating that our approach can produce compelling musical instruments. Specifically, we introduce a new objective metric to evaluate the timbral consistency of the generated instruments and adapt the average Contrastive Language-Audio Pretraining (CLAP) score for the text-to-instrument case, noting that its naive application is unsuitable for assessing this task. Our findings reveal a complex interplay between timbral consistency, the quality of generated samples, and their correspondence to the input prompt.

1. INTRODUCTION

The exploration of sound synthesis and the development of interfaces to manipulate timbre are fundamental topics in audio research [1]. With the evolution of sound synthesis in the digital realm, musicians have unprecedented means to manifest their artistic visions. Meanwhile, generative models for images and text have shown disruptive abilities in creating novel samples from learned distributions [2]. It becomes only natural to consider implications of such technologies when applied to a music production context.

Several generative models for neural audio synthesis have been put forth, including NSynth [3], which uses a WaveNet [4] autoencoder to create samples of pitched instruments, and GANSynth [5], which models signal phase through an instantaneous frequency representation. Furthermore, Differentiable Digital Signal Processing (DDSP)

[6] and its related works [7] introduce autoencoders with differentiable synthesizers for improved controllability, while a novel approach via a real-time variational autoencoder is presented in [8]. Additionally, GANstrument [1] leverages a feature descriptor obtained through adversarial domain confusion, highlighting the diverse methodologies employed to advance the field of audio synthesis. These models lack an interface for controlling audio generation via text input. Accordingly, we have witnessed a surge in text-to-audio systems generating convincing audio examples from text prompts [9]. One family of approaches rely on neural audio codecs [10, 11] representing audio as a set of discrete codes whose sequence can be learned using transformer-based language models. While initial approaches targeted speech [12, 13] and ambient sounds [14], follow-on works adapt methods for text-to-music generating full musical passages from text [15, 16].

Though compelling, seminal text-to-music works target generation of entire musical arrangements or otherwise lack fine-grained control over their outputs, and might not integrate well into musicians' workflows. Consequently, efforts have been made to adapt these models to sit closer in the creative process. These include StemGen [17], predicting instrument track layers from a given musical context, and VampNet [18], generating musical variations via generative filling. We align with this philosophy, intending to enable new sounds to inspire musical creativity.

In this paper, we introduce the application of neural audio codec language models for the automated creation of sample-based musical instruments using both text and audio prompts as input, building upon our preliminary work in progress in [19]. We model a musical instrument as a set of waveforms sampling the instrument's time-domain response across the dimensions of pitch (the fundamental frequency of a note) and velocity (the intensity with which a note is played). Under this paradigm, we move beyond the constraints of any one parametric synthesizer, avoiding expressivity limitations tied to its implementation. As in [1], we note that injecting inductive bias into the generative process via DDSP is interesting but complementary to our work, as such methods constrain the manifold that outputs can live on [20]. Unlike text-to-music systems, which typically generate a single audio example for a given text prompt during inference, prompt-to-instrument systems must generate an ensemble of audio samples from a fixed prompt, which must be pitch-accurate and timbrally



© S. Nercessian, J. Imort, N. Devis, and F. Blang. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** S. Nercessian, J. Imort, N. Devis, and F. Blang, "Generating Sample-Based Musical Instruments Using Neural Audio Codec Language Models", in *Proc. of the 25th Int. Society for Music Information Retrieval Conf.*, San Francisco, United States, 2024.

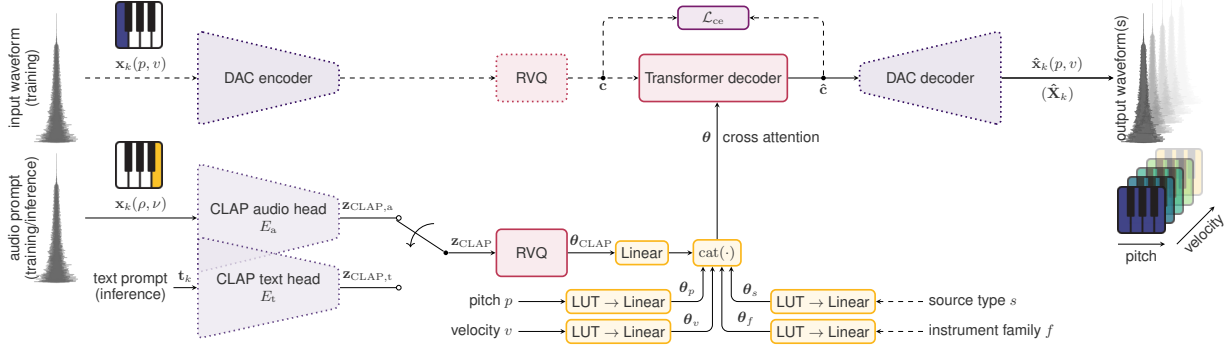


Figure 1. Overview of our proposed system. Dotted lines represent frozen pretrained modules. Dashed lines denote steps exclusive to training. CLAP’s audio or text head can be used at inference, disregarding source type and instrument family. Training operates on individual samples \mathbf{x} , while inference creates a set of samples $\hat{\mathbf{X}}$ from a consistent CLAP prompt and varied pitch/velocity cues to create a full instrument. Different piano keys/colors denote different pitches/velocities.

consistent with one another to allow for the assembly of a playable instrument. Our contributions are as follows:

- We introduce the text-to-instrument (T2I) task, in which waveforms comprising a sample-based musical instrument are generated from a user text prompt.
- We propose neural audio codec language models as solutions for both text- and audio-prompted sample-based instrument generation, expanding on a state-of-the-art generative audio model that is conditioned on a Contrastive Language-Audio Pretraining (CLAP) embedding [21], as well as pitch across the 88-key range of a standard full-length piano keyboard, velocity, instrument family and source type.
- We develop an objective metric to assess the timbral consistency (TC) of sample-based instruments.
- We propose an adaptation to the average CLAP score to be suitable for objectively assessing T2I.
- We propose and analyze three CLAP conditioning schemes through qualitative and quantitative means.
- We demonstrate the compatibility of our approach with both autoregressive (AR) and non-AR audio transformers like MAGNeT [22].

The remainder of this paper is organized as follows: Section 2 describes our proposed method, Section 3 outlines quantitative metrics for assessing performance, including the ones that we have developed, Section 4 reports our experimental results, and Section 5 draws conclusions.

2. PROPOSED METHOD

Figure 1 illustrates our proposed method, which is based on MusicGen [16] as a foundation, consisting of a neural audio codec and a language model to predict acoustic tokens from conditioning signals. We replace EnCodec [23] used in MusicGen with the Descript Audio Codec (DAC) [11], addressing codebook collapse in previous models while achieving higher audio fidelity. We also introduce a set of new conditioning signals including pitch and velocity, alongside a CLAP embedding [21]. Our conditioning signals reflect global cues θ for steering generation, which are fused with the language model via cross-attention. Using CLAP allows instrument samples to be inferred from

either audio or text prompts, and we denote their tasks as sample-to-instrument (S2I) and T2I, respectively. The aim of S2I may be considered one of pitch/velocity shifting, whereby the model transforms an audio prompt in ways transcending conventional signal processing. In T2I, text acts as a semantic interface to generate instruments whose timbres may otherwise not exist. To ensure the reproducibility of our findings, we use pretrained sub-networks without modification, training our core language models from random initialization on the standard research dataset NSynth [3]. We acknowledge that fine-tuning sub-modules within a generative model can improve a composite system, but consider this to be outside the scope of this work.

2.1 Compressed audio representation

We use the DAC encoder to create an intermediate representation of a monophonic waveform \mathbf{x} , resulting in the discrete codes \mathbf{c} , while the DAC decoder synthesizes an audio waveform $\hat{\mathbf{x}}$ from a predicted code sequence $\hat{\mathbf{c}}$. The DAC is trained on a broad spectrum of audio types, so we deem it suitable for generating tonal one-shot instrumental sounds. We model our task at a sample rate of 44.1 kHz, as this would be a minimum requirement for real-world music production use cases. We employ the corresponding pretrained model with fixed weights during training.

2.2 Language model

To model the discrete audio tokens of single-shot samples, we consider a smaller, 60M parameter variant of the transformer decoder in [16], in order to prevent overfitting, speed up inference, and conceptually demonstrate our approach. The model consists of 12 layers with 16 attention heads per layer and a transformer dimension $d = 512$. We consider scaling our models to larger sizes to be out of scope for this work. As in MusicGen [16], we predict audio from tokens of the 4 most significant [11] codebooks at each frame (of the 9 supported by DAC), selecting tokens from codebooks of size 1024. At inference time, we consider next-token prediction using AR sampling with delayed pattern interleaving [16], as well as the iterative decoding scheme proposed in [22] reporting a $7\times$ inference

speed-up. For MAGNeT-style inference, we use 20 decoding steps for the first codebook, and 10 for the remaining codebooks, respectively (compared to 345 steps for the AR scheme). As is customary, we can leverage classifier-free guidance at inference time in both cases [16, 17]. We expect AR priors to provide higher fidelity, considering the importance of onsets to perception [24] for the single-shot samples that we generate: earlier audio token predictions are likely to be perceptually more relevant than later ones.

2.3 Categorical conditioning

We use a categorical conditioning scheme for pitch p , velocity v , broad instrument family f , and source type s , that consists of a lookup table (LUT) and a fully connected layer that maps the dimension of the categorical feature space to the dimension d of the language model. For pitch, we model the $d_p = 88$ range of notes spanned by a full-length keyboard, corresponding to Musical Instrument Digital Interface (MIDI) note numbers 21-108, and note this to be a significant expansion relative to the chroma feature used in [16]. We consider $d_v = 5$ velocity layers, according to MIDI velocities 25, 50, 75, 100, and 127 within our training dataset. The instrument family (i.e. bass, brass, etc.) and source type (i.e., acoustic, electronic, etc.) attributes in our dataset serve as metadata-driven timbral cues that could optionally guide training [25], but we do not expect them to be specified at inference. We choose to include them for models trained in this work, subjecting them to dropout with 30% probability, noting that dropout can generalize their complete inclusion or exclusion.

2.4 Joint text and audio conditioning

We use the CLAP model [21], employing encoders to generate a common fixed-dimensional representation for audio/text pairs of size $d_z = 512$. This model was pretrained on musical signals, utilizing a contrastive loss to align respective audio and text embeddings, ultimately enabling the use of either modality as input to our system. The audio encoder E_a uses HTS-AT [26], while the text encoder E_t is based on RoBERTa [27]. Given an audio dataset of instrumental samples, this strategy allows for leveraging only the audio head during language model training, without requiring rich text captions in the dataset. We quantize resulting CLAP embeddings through Residual Vector Quantization (RVQ) with learned codes [16], yielding θ_{CLAP} .

A distinction between generating music and creating sample-based instruments from prompts is that the inference scenario for instrument generation utilizes a single fixed representation as input for generating a cohesive set of waveforms comprising an instrument. Consequently, we present three CLAP conditioning schemes specifically to train language models for sample-based instrument creation. These techniques amount to assigning pairs of $\mathbf{z}_{\text{CLAP},a}$ and codes \mathbf{c} as input and target training examples in various ways, where $\mathbf{z}_{\text{CLAP},a}$ is the output of the CLAP audio encoder E_a . Hence, the target codes and CLAP embedding within a training example need not be derived from the same waveform, so long as they come

from the same instrument. Excluding θ_f and θ_s for clarity, the forward pass observed during the training of a language model Θ is

$$\hat{\mathbf{c}} = \Theta(\mathbf{z}_{\text{CLAP},a}, \theta_p, \theta_v), \quad (1)$$

where $\mathbf{z}_{\text{CLAP},a} = E_a(\mathbf{x}_k(\rho, \nu))$. Here, k , ρ , and ν denote the timbre (i.e. instrument), pitch, and velocity exhibited in an underlying audio example, respectively, which we assume to be readily selectable from our training set. This $\mathbf{x}_k(\rho, \nu)$ is the input to E_a , and need not be identical to $\mathbf{x}_k(p, v)$ which is used to derive the target codes \mathbf{c} .

2.4.1 Baseline CLAP

By design, the CLAP audio encoder E_a will inevitably yield distinct numerical representations for instrumental samples of the same instrument but varying in pitch or velocity. During training, the following equation applies:

$$\mathbf{z}_{\text{CLAP},a} = E_a(\mathbf{x}_k(p, v)), \quad (2)$$

While this suffices for creating a music track from a singular representation, the scenario diverges significantly for sample-based instrument creation. Specifically, pitch and velocity are represented through both the CLAP representation as well as their respective categorical conditioners, which can reduce the overall effectiveness of the latter. We consider this adaptation of existing prompt-to-audio methodologies to serve as a baseline in this work, noting its application to this task is still novel.

2.4.2 Random CLAP

In order to disentangle the aforementioned pitch/velocity effect, we consider a randomization technique defined by

$$\mathbf{z}_{\text{CLAP},a} = E_a(\mathbf{x}_k(\tilde{\rho}, \tilde{\nu})), \quad (3)$$

with $\tilde{\rho} \sim \mathcal{U}\{21, \dots, 108\}$, and $\tilde{\nu} \sim \mathcal{U}\{25, 50, 75, 100, 127\}$. Random selection with replacement is performed throughout training. This method resembles the nearest neighbor data augmentation in [1], where we consider samples to be neighbors if they originate from the same instrument.

2.4.3 Fixed CLAP

Lastly, we consider a conditioning scheme where we use a fixed, predefined CLAP embedding for each instrument as

$$\mathbf{z}_{\text{CLAP},a} = E_a(\mathbf{x}_k(\rho_{0,f}, \nu_0)), \quad (4)$$

where $\rho_{0,f}$ is defined for each instrument family f (see Table 1) such that fixed representations are sampled within the natural range of each instrument (i.e. we make

Instrument families	Note name
Bass	C2
Brass, String, Synth lead	C3
Guitar, Keyboard, Organ, Reed, Vocal	C4
Flute, Mallet	C5

Table 1. Pitch values used for fixed CLAP conditioning.

lower-pitched selections for bass sounds). The categorical velocity ν_0 is fixed across the training set at velocity 100, conveying an instrument’s timbre played with a medium/strong intensity. If a sample matching a $\rho_{0,f}$ and ν_0 query is not available within an instrument, we opt for its nearest available pitch, followed by its nearest velocity.

Other fixed CLAP conditioning forms could also have been devised, e.g. using average per-instrument CLAP embeddings. We opt for our described approach as it ensures that each CLAP embedding used in model training originates from exactly one audio example. We assert that this fixed variant most closely aligns training to the scenario at inference. In fact, we posit that both the baseline and random CLAP approaches are data augmentation alternatives relative to this method, that increase the number of conditioning signal/target code pairs observed during training, while potentially introducing domain mismatches.

3. OBJECTIVE EVALUATION CRITERIA

We assess models across several objective criteria for S2I and T2I. Alongside the widely used Fréchet audio distance (FAD) [28] score, we introduce a novel metric to evaluate the TC of generated sample-based instruments. We also propose an adaptation of the average CLAP score to fairly evaluate text correspondence for T2I. Unless otherwise specified, we base instrument generation-specific metrics on the assumption that they are represented by $N_k = d_p d_v = 440$ audio samples. In practice, care is taken to properly aggregate/mask instrument statistics based on which samples are present.

3.1 FAD score

The FAD score allows a common framework for evaluating generative audio models using almost any audio feature descriptor [28]. We utilize a FAD metric formulated using VGGish, as in related works [15, 17]. We also report FAD scores using CLAP (audio) embeddings, since they form a pivotal component to our system, allow analysis for higher-sample rate audio (48 kHz), and have been shown to have increased correlation to perception relative to VGGish [29]. The FAD score is generically defined as

$$\text{FAD}(\mathbf{Z}_1, \mathbf{Z}_2) = \|\mu_1 - \mu_2\|_2^2 + \text{Tr}\left(\mathbf{A}_1 + \mathbf{A}_2 + (\mathbf{A}_1 \mathbf{A}_2)^{\frac{1}{2}}\right), \quad (5)$$

where $\mathbf{Z}_i \in \mathbb{R}^{d_z \times TN}$ is a collection of T d_z -dimensional embeddings extracted by a given audio descriptor, across N samples from a population $i \in [1, 2]$. Considering the 4-second long audio segments generated in this work and the strides of various models, $T = 4$ and 1 when using VGGish and CLAP, respectively. We reserve subscripts 1 and 2 to denote ground truth/test populations, respectively. Accordingly, each \mathbf{Z}_i has mean $\mu_i \in \mathbb{R}^{d_z}$ and covariance $\mathbf{A}_i \propto \mathbf{Z}_i \mathbf{Z}_i^\top \in \mathbb{R}^{d_z \times d_z}$. The first and second terms in Equation 5 quantify mean correspondence and similarities in the spread between distributions, respectively. The FAD score possesses a property allowing unpaired populations

to be compared, which we use as a criterion to assess "in-the-wild" T2I in lieu of ground truth audio.

3.2 TC score

Our system should generate timbrally consistent samples in order for them to triggered harmoniously as a sample-based instrument, and we aim to characterize this quantitatively. An apt definition for TC may seem ill-posed, since we want instrument samples to be fundamentally consistent with one another, but also expect them to exhibit some timbral variations as functions of pitch/velocity. This is particularly sought-after in high-quality virtual instruments, motivating the modeling approach in [6]. To contend with these potentially conflicting aspirations, we learn statistics from existing sample-based instruments serving as prototypes for realistic TC, and build metrics around them. We use CLAP embeddings as a basis to create an elegant embodiment in this work. To do so, we forego the mean subtraction step standard to covariance matrix computations, noting that samples are practically close to zero-mean in this respect. Hereafter, we use the terms covariance, affinity, and cosine similarity interchangeably.

We define per-instrument covariance matrices as

$$\mathbf{A}_{ij,k} = \frac{1}{N_k} \mathbf{Z}_{i,k}^\top \mathbf{Z}_{j,k}, \quad (6)$$

where $\mathbf{A}_{ij,k} \in \mathbb{R}^{N_k \times N_k}$ is the affinity between embeddings $\mathbf{Z}_{i,k}$ and $\mathbf{Z}_{j,k} \in \mathbb{R}^{d_z \times N_k}$ representing the subset of CLAP embeddings of the k th instrument within each population. Here, we compute statistics emphasizing variations across samples instead of feature dimensions. Referring to Equation 5, the L_2 -normalized quality CLAP embeddings will ensure us that $\text{Tr}(\mathbf{A}_{ii,k}) = 1 \forall i \in [1, 2]$ and $k \in [1, \dots, K]$. Accordingly, we can define

$$\text{TC}_{\text{CLAP}}(\mathbf{Z}_1, \mathbf{Z}_2) = \frac{1}{K} \sum_k \text{Tr}\left((\mathbf{A}_{11,k} \mathbf{A}_{22,k})^{\frac{1}{2}}\right), \quad (7)$$

which is bounded in $[0, 1]$ and aggregates the similarity in covariations across instruments within each population. Instead of using $\mathbf{A}_{11,k}$ for making comparisons between populations on a per-instrument basis, we consider $\mathbf{A}_{11,*} = \frac{1}{K} \sum_k \mathbf{A}_{11,k}$, averaging per-instrument affinity matrices across a ground truth evaluation set. This provides richer statistics for improved stability, and a unified method to assess TC for S2I and T2I. The TC score is then

$$\text{TC}_{\text{CLAP}^*}(\mathbf{Z}_1, \mathbf{Z}_2) = \frac{1}{K} \sum_k \text{Tr}\left((\mathbf{A}_{11,*} \mathbf{A}_{22,k})^{\frac{1}{2}}\right). \quad (8)$$

We compute $\mathbf{A}_{11,*}$ using all of the samples from the NSynth validation and test sets that are within our desired 88-key pitch range, reflecting a total of 53 instruments. The resulting covariance matrix is illustrated in Figure 2c, in which samples are ordered primarily by pitch and secondarily by velocity. Note how $\mathbf{A}_{11,*}$ deviates from "ideal TC," whereby all embeddings would be correlated with unity similarity (see Figure 2a). Moreover, a 5×5 texture

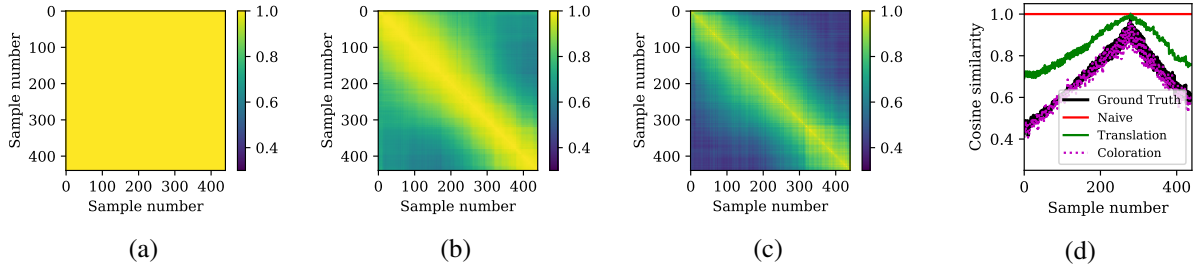


Figure 2. Covariance matrices for the text prompt $\mathbf{t}_k = \text{aggressive synth lead}$, computed using (a) naive replication, (b) translation, (c) coloration (matching the ground truth covariance $\mathbf{A}_{11,*}$ learned over the 53 instruments reflected in the NSynth validation/test sets), (d) cosine similarities relative to estimated $\hat{\rho}_k/\hat{\nu}_k$, corresponding to note E5/velocity 100.

emerges in $\mathbf{A}_{11,*}$, indicative of variations in cosine similarity amongst samples of the same pitch but differing velocities. Lastly, one may question the suitability of CLAP as a feature descriptor within this context, given its variability concerning pitch/velocity discussed in Section 2.4. Its improved correlation to perception aside [29], we assert that learning statistics over data effectively embeds potential measurement deficiencies that effectively neutralizes when we compare new population statistics against it.

3.3 Average CLAP score

3.3.1 Sample-to-instrument (S2I)

Given $N = \sum_k N_k$ and a cross-population covariance $\mathbf{A}_{ij} = \frac{1}{N} \mathbf{Z}_i^\top \mathbf{Z}_j \in \mathbb{R}^{N \times N}$, the average CLAP score computed on a per-sample basis can be expressed concisely as

$$s_{\text{CLAP}}(\mathbf{Z}_1, \mathbf{Z}_2) = \text{Tr}(\mathbf{A}_{12}) = \frac{1}{N} \sum_k N_k \text{Tr}(\mathbf{A}_{12,k}). \quad (9)$$

It can also be computed on a per-instrument basis by

$$s_{\text{CLAP}*}(\mathbf{Z}_1, \mathbf{Z}_2) = \frac{1}{K} \sum_k \text{Tr}(\mathbf{A}_{12,k}). \quad (10)$$

We opt for this version in our work, noting that the two measures are equivalent when $N_1 = N_2 = \dots = N_K$.

3.3.2 Text-to-instrument (T2I)

The average CLAP score $s_{\text{CLAP}*}$ is suitable for cases with a one-to-one match between ground truth prompts and their corresponding audio examples. However, it can deteriorate for T2I, where a single CLAP text embedding must be related to an ensemble of CLAP audio embeddings $\mathbf{Z}_{2,k}$. A naive adaptation involves comparing each audio embedding within the generated instrument to the same target text embedding. This amounts to creating $\mathbf{Z}_{1,k}$ by replicating the CLAP text embedding N_k times (whose resulting covariance is the "ideal TC" one in Figure 2a), and using it as input to Equation 10. Hence, we set out to *synthesize* a realistic ensemble of CLAP embeddings $\mathbf{Z}_{1,k}$ from a single CLAP text embedding $\mathbf{z}_{\text{CLAP},t} = E_t(\mathbf{t}_k)$, derived from the k th text prompt \mathbf{t}_k . Again, we accomplish this by leveraging statistics from available instrument data.

We construct $\mathbf{M}_{1,*} \in \mathbb{R}^{d_z \times d_p d_v}$ as the mean CLAP audio embeddings at each pitch/velocity pair across all instruments in our evaluation data, re-normalizing them upon

averaging. We posit that a text prompt implies a specific pitch/velocity (e.g., "softly plucked upright bass" suggests a low pitch/velocity). To estimate the corresponding pitch $\hat{\rho}_k$ and velocity $\hat{\nu}_k$ for a given prompt, and to identify its closest template $\hat{\mu}_{1,k}$, we use $\mathbf{M}_{1,*}$ as a template matching-based classifier onto $\mathbf{z}_{\text{CLAP},t}$. Accordingly, we can define

$$\mathbf{M}_{1,k} = \mathbf{M}_{1,*} + (\hat{\mu}_{1,k} - \mathbf{z}_{\text{CLAP},t}) \quad (11)$$

such that $\mathbf{M}_{1,k}$ is aligned to $\mathbf{z}_{\text{CLAP},t}$ at $\hat{\rho}_k/\hat{\nu}_k$. Re-normalizing, we have $\mathbf{Z}_{1,k} = \mathbf{M}_{1,k}/\|\mathbf{M}_{1,k}\|$. Figure 2b illustrates a covariance matrix derived from this approach for a given text prompt. This *translation* method improves upon naive replication, but contains higher cross-correlations than in $\mathbf{A}_{11,*}$. Finally, we derive a *coloration* transformation $\mathbf{Z}_{1,k} \leftarrow Y(\mathbf{Z}_{1,k}, \mathbf{A}_{11,*})$ through standard Eigendecomposition techniques, resulting in a $\mathbf{Z}_{1,k}$ with covariance $\mathbf{A}_{11,*}$, as in Figure 2c.

4. EXPERIMENTAL RESULTS

We train models on the NSynth dataset [3], pruning it according to our specified 88-key pitch range. We re-sample the 16 kHz dataset to 44.1 kHz, viewing it as a proxy in lieu of an equally comprehensive full-band alternative. Models are trained to minimize the cross-entropy \mathcal{L}_{ce} between predicted codes $\hat{\mathbf{c}}$ and ground truth \mathbf{c} , over 1M training steps with AdamW optimizer, a batch size of 48, and a cosine-annealed schedule as in [16] with an initial learning rate of 10^{-3} . We primarily analyze the impact of the proposed CLAP conditioning training variants with AR inference. Additionally, we train a baseline CLAP model with MAGNeT-style iterative decoding to compare its relative performance. To promote consistency in generated samples used for evaluation, we fix the random seed of our categorical samplers, ensuring that generations undergo the same random sampling trajectory. We refer readers to our supplementary materials available at <https://gen-inst.netlify.app/>.

We evaluate and analyze the models through several means. We liken S2I to a reconstruction of the NSynth test set [1] adapted to our inference condition, as a user can provide a sample at any pitch/velocity available to them and models must render its timbre over all pitch/velocity queries. We simulate this by randomly selecting a single query CLAP audio embedding for each instrument, using it to generate all other samples within the instrument. For

Model	Inference	FAD _{VGGish} ↓	FAD _{CLAP} ↓	s _{CLAP*} ↑	TC _{CLAP*} ↑
Baseline CLAP	AR	1.781	0.214	0.626	0.937
Random CLAP	AR	1.558	0.196	0.656	0.929
Fixed CLAP	AR	1.951	0.225	0.637	0.951
Baseline CLAP	MAGNeT	1.974	0.263	0.561	0.931

Table 2. Objective S2I evaluation over the NSynth test set.

Model	FAD _{VGGish} ↓	FAD _{CLAP} ↓	TC _{CLAP*} ↑	Naive	Translation	Coloration
Baseline CLAP	3.060	0.402	0.908	0.225	0.239	0.359
Random CLAP	2.416	0.315	0.883	0.168	0.224	0.361
Fixed CLAP	3.668	0.427	0.932	0.171	0.204	0.333

Table 3. Objective T2I evaluation over a curated set of text prompts (left), and using s_{CLAP*} ↑ comparing naive application of CLAP text embeddings against the proposed translation and coloration methods for synthesizing Z_{1,k} (right).

T2I, we curate 25 text prompts of varying complexity, generating instruments accordingly.

4.1 Objective evaluation

We analyze generations across S2I and T2I, using FAD (for overall expressivity and fidelity), s_{CLAP*} (for prompt correspondence), and TC_{CLAP*} (for TC) to evaluate models quantitatively. To compute FAD scores for T2I, we relate generated instruments to the NSynth test set in the absence of the ground truth audio. Lastly, we compare the different s_{CLAP*} versions for T2I introduced in Section 3.3.2.

Quantitative results for S2I and T2I are summarized in Tables 2 and 3, respectively. For S2I, the random CLAP variant outperforms other models in terms of FAD and s_{CLAP*} at the expense of reduced TC. The converse is true for the fixed CLAP variant, which outperforms in TC. While we do not prescribe which factor is most crucial to overall instrument quality, we do assert that TC is an important element for overall playability. The baseline CLAP approach slots itself in the middle with regards to all criteria. Its MAGNeT variant exhibits degraded performance, but generates samples with 7× fewer inference steps. These findings are largely mirrored in the T2I case. Interestingly, the baseline CLAP variant seemingly outperforms models in terms of s_{CLAP*} using a naively adapted measure. The translation method increases scores across all models. Lastly, we see that the random CLAP model (marginally) outperforms other variants when using the coloration method, in line with S2I. Note that this version of the measure significantly bolsters s_{CLAP*} across all models relative to naive replication and translation, so we argue that it is best-suited for computing T2I s_{CLAP*}.

4.2 Subjective evaluation

We used the Multiple Stimuli with Hidden Reference and Anchor (MUSHRA) and Mean Opinion Scores (MOS) methods [30] to evaluate model variants subjectively. The MUSHRA test was catered to S2I, and involved participants rating the quality of individual samples generated by different models against a hidden reference (i.e. a ground truth sample) and an anchor (i.e. a sample generated by a

randomly initialized model). We performed a 1-5 Likert scale MOS test for T2I scenarios, where participants evaluated the audio outputs generated from text prompts based on overall playability and TC. Our accompanying website demonstrates the nature of trials used in our evaluation.

In total, 62 participants took part in our two-phase evaluation, with results summarized in Table 4. Note that most participants possess expert listening skills and have been involved in virtual instrument creation for several years, contributing to slightly lower absolute results than anticipated. Listening test results were consistent with our objective evaluation, confirming the two assertions of our work: (1) random CLAP improves expressivity over baseline CLAP by virtue of its data augmentation, and (2) fixed CLAP improves TC over baseline CLAP because its training more closely resembles the inference condition.

Model	MUSHRA	MOS
Baseline CLAP	56.08	2.290
Random CLAP	63.35	2.661
Fixed CLAP	57.96	2.820
Ground truth	98.45	–
Anchor	0.442	–

Table 4. Summary of our subjective listening tests.

5. CONCLUSIONS

In this work, we proposed methods for generating sample-based musical instruments from text or audio prompts using neural audio codec language models. We considered different CLAP conditioning variants based on the unique challenge of our task, whereby a set of samples that are timbrally consistent must be generated from a single prompt. We proposed metrics to assess sample-based instruments through various means. Extensive evaluations showcased the effectiveness of our methods, underscoring a compromise between expressivity and TC. Future work will enable deeper control for sample generation, where adapters could be used to augment a base model [31]. We would also like to improve system fidelity, scaling models to larger sizes with fine-tuned modules [9].

6. ETHICS STATEMENT

We have intentionally pursued this task as a topic for scientific research as an alternative to more conventional prompt-to-media systems. The spirit of this work is specifically to expand sound synthesis possibilities for music creators in order to realize their artistic visions. Moreover, we feel that our resulting system and its intents pose far less risk to personal attack/misrepresentation as well as the livelihood of creatives, and is less susceptible to incrimination/impersonation attempts relative to the forms of generative models that have caused increased levels of concern within the general population [32].

Beyond our primary ethical concerns, we also recognize the environmental implications of our computational practices. Our experiments were carried out using Amazon Web Services in the *us-gov-east-1* region, with a carbon efficiency of 0.57 kgCO₂eq per kilowatt-hour. One training of our model entailed approximately 96 hours of computation on Intel Xeon E5-2686 v4 (Broadwell) hardware using a single V100 GPU, culminating in an estimated total emission of 7.93 kgCO₂eq. This estimation was facilitated by the Machine Learning Impact calculator [33]. In acknowledging our environmental impact, we underscore the importance of integrating sustainability considerations into the research process, reflecting on the imperative to balance innovation with ecological responsibility.

7. REFERENCES

- [1] G. Narita, J. Shimizu, and T. Akama, “GANStrument: Adversarial Instrument Sound Synthesis with Pitch-Invariant Instance Conditioning,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Jun. 2023.
- [2] H. Chang, H. Zhang, J. Barber, A. Maschinot, J. Lezama, L. Jiang, M.-H. Yang, K. P. Murphy, W. T. Freeman, M. Rubinstein, Y. Li, and D. Krishnan, “Muse: Text-To-Image Generation via Masked Generative Transformers,” in *Proceedings of the International Conference on Machine Learning*, Jul. 2023.
- [3] J. Engel, C. Resnick, A. Roberts, S. Dieleman, D. Eck, K. Simonyan, and M. Norouzi, “Neural audio synthesis of musical notes with WaveNet autoencoders,” in *Proceedings of the International Conference on Machine Learning*, Aug. 2017.
- [4] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, “WaveNet: A generative model for raw audio,” *arXiv:1609.03499*, 2016.
- [5] J. Engel, K. K. Agrawal, S. Chen, I. Gulrajani, C. Donahue, and A. Roberts, “GANSynth: Adversarial Neural Audio Synthesis,” in *Proceedings of the International Conference on Learning Representations*, May 2019.
- [6] J. Engel, L. Hantrakul, C. Gu, and A. Roberts, “DDSP: Differentiable Digital Signal Processing,” in *Proceedings of the International Conference on Learning Representations*, April 2020.
- [7] D. Y. Wu, W. Y. Hsiao, F. R. Yang, O. Friedman, W. Jackson, S. Bruzenak, Y. W. Liu, and Y. H. Yang, “DDSP-Based Singing Vocoders: A New Subtractive Based Synthesizer and A Comprehensive Evaluation,” in *Proceedings of the International Society for Music Information Retrieval Conference*, Dec. 2022.
- [8] A. Caillon and P. Esling, “RAVE: A Variational Autoencoder for Fast and High-Quality Neural Audio Synthesis,” *arXiv:2111.05011*, Nov. 2021.
- [9] Z. Evans, C. Carr, J. Taylor, S. H. Hawley, and J. Pons, “Fast timing-conditioned latent audio diffusion,” *arXiv:2402.04825*, Feb. 2024.
- [10] N. Zeghidour, A. Luebs, A. Omran, J. Skoglund, and M. Tagliasacchi, “SoundStream: An End-to-End Neural Audio Codec,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, Nov. 2021.
- [11] R. Kumar, P. Seetharaman, A. Luebs, I. Kumar, and K. Kumar, “High-fidelity audio compression with improved RVQGAN,” *Conference on Neural Information Processing Systems*, Dec. 2023.
- [12] Z. Borsos, R. Marinier, D. Vincent, E. Kharitonov, O. Pietquin, M. Sharifi, D. Roblek, O. Teboul, D. Grangier, M. Tagliasacchi, and N. Zeghidour, “AudioLM: a Language Modeling Approach to Audio Generation,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, Jun. 2023.
- [13] C. Wang, S. Chen, Y. Wu, Z. Zhang, L. Zhou, S. Liu, Z. Chen, Y. Liu, H. Wang, J. Li, L. He, S. Zhao, and F. Wei, “Neural Codec Language Models are Zero-Shot Text to Speech Synthesizers,” *arXiv:2301.02111*, Jan. 2023.
- [14] F. Kreuk, G. Synnaeve, A. Polyak, U. Singer, A. Défossez, J. Copet, D. Parikh, Y. Taigman, and Y. Adi, “AudioGen: Textually Guided Audio Generation,” in *Proceedings of the International Conference on Learning Representations*, 2023.
- [15] A. Agostinelli, T. I. Denk, Z. Borsos, J. Engel, M. Verzetti, A. Caillon, Q. Huang, A. Jansen, A. Roberts, M. Tagliasacchi, M. Sharifi, N. Zeghidour, and C. Frank, “MusicLM: Generating Music From Text,” *arXiv:2301.11325*, Jan. 2023.
- [16] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Défossez, “Simple and Controllable Music Generation,” in *Proceedings of the Conference on Neural Information Processing Systems*, Dec. 2023.
- [17] J. D. Parker, J. Spijkervet, K. Kosta, F. Yesiler, B. Kuznetsov, J. C. Wang, M. Avent, J. Chen, and

- D. Le, “StemGen: A music generation model that listens,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Apr. 2024.
- [18] H. F. Garcia, P. Seetharaman, R. Kumar, and B. Pardo, “VampNet: Music generation via masked acoustic token modeling,” in *Proceedings of the International Society for Music Information Retrieval Conference*, Nov. 2023.
- [19] S. Nercessian and J. Imort, “InstrumentGen: Generating sample-based musical instruments from text,” in *Neural Information Processing Systems Workshop on Machine Learning for Audio*, Dec. 2023.
- [20] B. Hayes, J. Shier, G. Fazerkas, A. McPherson, and C. Saitis, “A Review of Differentiable Digital Signal Processing for Music and Speech Synthesis,” *Frontiers in Signal Processing*, Jan. 2024.
- [21] Y. Wu, K. Chen, T. Zhang, Y. Hui, T. Berg-Kirkpatrick, and S. Dubnov, “Large-Scale Contrastive Language-Audio Pretraining with Feature Fusion and Keyword-to-Caption Augmentation,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Jun. 2023.
- [22] A. Ziv, I. Gat, G. L. Lan, T. Remez, F. Kreuk, J. Copet, A. Défossez, G. Synnaeve, and Y. Adi, “Masked audio generative modeling,” in *Proceedings of the International Conference on Learning Representations*, May 2024.
- [23] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, “High Fidelity Neural Audio Compression,” *Transactions on Machine Learning Research*, Sep. 2023.
- [24] C. Hawthorne, E. Elsen, J. Song, A. Roberts, I. Simon, C. Raffel, J. Engel, S. Oore, and D. Eck, “Onsets and frames: Dual-objective piano transcription,” in *Proceedings of the International Society for Music Information Retrieval Conference*, Sep. 2018.
- [25] V. Vapnik and R. Izmailov, “Learning using privileged information: Similarity control and knowledge transfer,” *Journal of Machine Learning Research*, Nov. 2015.
- [26] K. Chen, X. Du, B. Zhu, Z. Ma, T. Berg-Kirkpatrick, and S. Dubnov, “HTS-AT: A Hierarchical Token-Semantic Audio Transformer for Sound Classification and Detection,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2022.
- [27] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, “RoBERTa: A Robustly Optimized BERT Pretraining Approach,” *arXiv:1907.11692*, Jul. 2019.
- [28] K. Kilgour, M. Zuluaga, D. Roblek, and M. Sharifi, “Frechet audio distance: A metric for evaluating music enhancement algorithms,” *arXiv:1812.08466*, Dec. 2018.
- [29] A. Gui, H. Gamper, S. Braun, and D. Emmanouilidou, “Adapting Frechet audio distance for generative music evaluation,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Apr. 2024.
- [30] J. Camp, T. Kenter, L. Finkelstein, and R. Clark, “MOS vs. AB: Evaluating text-to-speech systems reliably using clustered standard errors,” in *Proceedings of Interspeech*, Aug. 2023.
- [31] K. Sohn, N. Ruiz, K. Lee, D. C. Chin, I. Blok, H. Chang, J. Barber, L. Jiang, G. Entis, Y. Li, Y. Hao, I. Essa, M. Rubinstein, and D. Krishnan, “StyleDrop: Text-to-Image Generation in Any Style,” in *Proceedings of the Conference on Neural Information Processing Systems*, Dec. 2023.
- [32] J. Barnet, “The ethical implications of generative audio models: A systematic literature review,” in *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, Aug. 2023.
- [33] A. Lacoste, A. Luccioni, V. Schmidt, and T. Dandres, “Quantifying the carbon emissions of machine learning,” *arXiv:1910.09700*, 2019.

HIERARCHICAL GENERATIVE MODELING OF MELODIC VOCAL CONTOURS IN HINDUSTANI CLASSICAL MUSIC

Nithya Shikarpur^{1,2} Krishna Maneesha Dendukuri¹ Yusong Wu^{1,2}
Antoine Caillon⁴ Cheng-Zhi Anna Huang^{1,2,3,4}

¹ Mila, Quebec Artificial Intelligence Institute, ² Université de Montréal,
³ Canada CIFAR AI Chair, ⁴ Google DeepMind

snnithya@mit.edu, krishnamaneeshad@gmail.com, wu.yusong@mila.quebec
{acaillon, annahuang}@google.com

ABSTRACT

Hindustani music is a performance-driven oral tradition that exhibits the rendition of rich melodic patterns. In this paper, we focus on generative modeling of singers’ vocal melodies extracted from audio recordings, as the voice is musically prominent within the tradition. Prior generative work in Hindustani music models melodies as coarse discrete symbols which fails to capture the rich expressive melodic intricacies of singing. Thus, we propose to use a finely quantized pitch contour, as an intermediate representation for hierarchical audio modeling. We propose GaMaDHaNi, a modular two-level hierarchy, consisting of a generative model on pitch contours, and a pitch contour to audio synthesis model. We compare our approach to non-hierarchical audio models and hierarchical models that use a self-supervised intermediate representation, through a listening test and qualitative analysis. We also evaluate audio model’s ability to faithfully represent the pitch contour input using Pearson correlation coefficient. By using pitch contours as an intermediate representation, we show that our model may be better equipped to listen and respond to musicians in a human-AI collaborative setting by highlighting two potential interaction use cases (1) primed generation, and (2) coarse pitch conditioning.

1. INTRODUCTION

Hindustani music is a performance-driven music tradition that has a high level of melodic intricacy [1]. Despite the recent advances in generative modeling for music [2, 3], this genre remains difficult to model for several reasons including (1) a lack of a readily available and widely accepted abstract representation reflecting the genre faithfully (like Western symbolic notation), (2) as a niche musical form, the scarcity of available datasets restricts the ability to model the raw waveform directly.

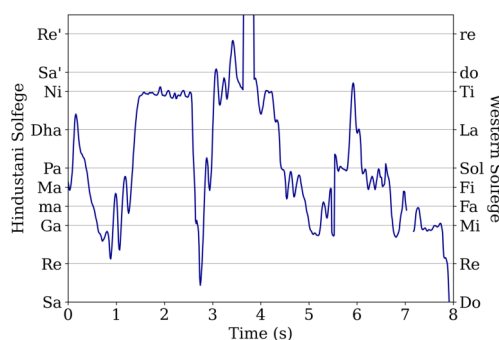


Figure 1. Extracted pitch from Hindustani vocal audio highlighting the melodic intricacies involved. Solfège notation is highlighted as a horizontal grid.

Symbolic notation is a well-defined discrete representation of music including lead sheet, MIDI, piano roll, text, and markup language. Musical notation used in Hindustani pedagogy uses a similar discrete representation by highlighting the prominent notes which fails to faithfully capture the fine melodic intricacies connecting these notes as seen in Fig. 1. Previous work on generative modeling for Hindustani music has side-stepped the lack of well-defined abstract representations with two methods: (1) using musical notation from textbooks or music theory [4–6], (2) leveraging MIDI extracted from audio [7, 8]. However, both methods ignore the rich melodic ornamentation present in this music. Computational analyses for the genre have addressed the difficulty in data representation by using the fundamental frequency contour, hereby referred to as ‘pitch’, as an intermediate representation for several melodic tasks including music style classification [9], motif discovery and matching [10–12] and *raga* recognition [13–15]. With evidence that pitch faithfully represents the melody for computational tasks, we are motivated to incorporate it in the context of generative modeling.

In this work, we present GaMaDHaNi¹ (Generative Modular Design of Hierarchical Networks), a modular hierarchical generative model for Hindustani singing. We employ a two-level hierarchy of data representation in-



© N. Shikarpur, K. M. Dendukuri, Y. Wu, A. Caillon and C. Z. A. Huang. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** N. Shikarpur, K. M. Dendukuri, Y. Wu, A. Caillon and C. Z. A. Huang, “Hierarchical Generative Modeling of Melodic Vocal Contours in Hindustani Classical Music”, in *Proc. of the 25th Int. Society for Music Information Retrieval Conf.*, San Francisco, United States, 2024.

¹Listen to audio samples and access code here: <https://snnithya.github.io/gamadhani-samples/>

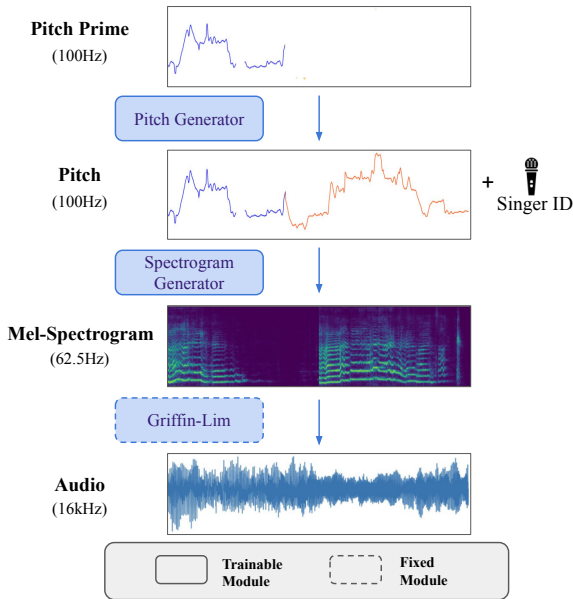


Figure 2. The overall hierarchical generation structure of GaMaDHaNi comprising of the Pitch Generator, the Spectrogram Generator and a vocoder. During inference, given an optional short melodic input, i.e. ‘prime’, each of the generators produce a pitch continuation and a spectrogram conditioned on the resulting pitch respectively.

cluding pitch and spectrogram. The Pitch Generator and Spectrogram Generator are trained to generate these respectively, with the generated spectrogram converted to audio using a vocoder. Fig. 2 highlights the model’s high-level structure. We choose a finely quantized pitch contour as an intermediate representation due to its close relation to melodic content, strongly established in prior literature [9–15]. We model pitch under two paradigms: as discrete tokens using an autoregressive transformer and as continuous values using a diffusion model. In addition, with a relatively small dataset of 120 hours, we find that the pitch intermediate representation is effective at learning melodically diverse ideas (Sec. 4.4). As possible use cases for interaction, (1) we explore using the model to continue a given melodic prompt, termed ‘prime’, as seen in Fig. 2, and (2) we extend the hierarchy upwards to include a coarse pitch target, thereby enabling user-driven steering of the generation process.

We note that our current generation pipeline lacks incorporation of several key elements crucial to Hindustani music, specifically tonic frequency, and raga and tala, i.e. melodic and rhythmic frameworks. This work establishes a preliminary foundation for exploring the potential of generating music within this form while maintaining its characteristic melodic intricacies.

A summary of our core contributions include:

- We propose GaMaDHaNi, the first model capable of generating Hindustani vocal contours while maintaining the rich melodic complexity in the music.
- We present a hierarchical approach to modeling a waveform using an intermediate pitch representation

that works on a small dataset (120 hours).

- Through listening tests and qualitative observations, we show that our hierarchical approach performs better than baselines.

2. RELATED WORK

2.1 Music Representations in Indian Art Music

Past work on melody-based computational tasks for Indian Art Music include music style classification [9], motif discovery and matching [10–12], and raga recognition [14–16]. Previous work shows that fine quantization outperforms coarse quantization in pitch contours for tasks including raga recognition [16, 17] and motif matching [11]. Thus motivated by their ability to capture melodic information we use finely quantized pitch as an intermediate representation. Additionally, for Carnatic music, previous work on compact representations for Gamakas (type of note ornamentation) [18], and non-uniform pitch quantization schemes that can preserve raga-characteristics [19, 20] present forms of representation that are more condensed than the pitch contour while being adequately detailed which could be an interesting inclusion for future work.

2.2 Generative Modeling for Hindustani Music

Hindustani music is an improvised form of music where melodic movements are guided by a melodic framework (raga) [1]. Past work on the generation of this music is of two types: rule-based and data-driven models. AI-Raga [4] is a rule-based AI system developed to generate musical notation of compositions and improvisations that adhere to raga grammar based on an elaborate set of rules termed ‘generative theory of music’ [21]. Another work develops a Finite State Machine (FSM) to generate improvisations based on raga-specific melodic movements situated in theory [5]. An initial attempt at data-driven models learned from the musical notation of *alaps*, i.e. slow improvisation, in textbooks using bigrams in an FSM [6]. RMMM [7] explores the use of LSTM [22] and transformer-based [23] architectures to generate MIDI extracted from a corpus of Hindustani music. Other work also proposes generating MIDI with GANs [8, 24]. All models discussed in this section approach modeling data as solfege notation. While doing so, one gives up on the transitory melodic regions between notes of the melody, which is inherent to Hindustani music. AI-Raga [4] partially addresses this by using domain-informed tuning systems, and a simulation of transitory glides between notes. We propose to address this problem by incorporating a fine pitch data representation. Additionally, in contrast to previous work, we propose to generate audio waveform rather than symbolic data.

2.3 Hierarchical Audio Generation

Within the domain of music generation, hierarchical learning offers two distinct advantages: enhanced learning abilities on data-constrained tasks and multi-level controllability. MIDI-DDSP [25] takes advantage of the hierarchy in

the process of creating realistic audio of instrument performance given a sequence of MIDI data including notes, high-level performance attributes and low-level synthesis attributes. Our approach leverages a different hierarchy based on pitch as opposed to MIDI notes, and we generate pitch from scratch without relying on any symbolic input. Moreover, we choose to directly generate audio spectrograms instead of DDSP synthesis parameters since the latter is designed mainly for instrumental sound.

Another approach to hierarchical models for audio includes the generation of pre-trained compressed representations of audio, i.e. neural audio codecs [26, 27], framed as a language modeling task as seen in MusicLM [28] and MusicGen [3]. We study the effectiveness of this approach as a baseline in our experiments in Sec. 4.3, by comparing Encodec [26] and pitch as intermediate representations.

The use of fundamental frequency contours as an intermediate representation has been widely adopted in the context of Text To Speech synthesis (TTS) and Singing Voice Synthesis (SVS). Both fields follow a hierarchy including an input-conditioned acoustic model which mainly generates a subset of pitch, duration, and spectral features followed by a vocoder. The input could be text in the case of TTS [29–31] and musical score for SVS [32, 33]. CDAR [34] is a TTS model that seeks to control the prosody of generated speech by allowing users to edit parts of the spoken pitch contour while maintaining the realism of the prosody. We thus choose to adopt pitch as an intermediate representation with a strong precedence for its use and controllability in speech and singing applications.

3. METHOD

In this work, we seek a generative model for Hindustani vocal music by learning the joint distribution of amplitude mel-spectrograms s and pitch f following

$$p(s, f) = p_\phi(s|f)p_\theta(f), \quad (1)$$

where p_ϕ and p_θ are parameterized with neural networks called *Spectrogram* and *Pitch* Generators respectively. The generated spectrogram is converted to audio using a vocoder. Pitch conditioning f to p_ϕ is taken from our dataset for training and sampled from p_θ for inference.

3.1 Pitch Generator

We study the modeling of vocal pitch as the primary component in our hierarchical generation pipeline. Vocal pitch f are represented as integer-valued sequences sampled at 100Hz, with 90% of the values ranging from 86Hz to 899Hz, quantized with a fine resolution of 10 cents. To model such sequences, we investigate two distinct methods. The first employs an autoregressive, language-like model to predict the discrete pitch sequence, whereas the second leverages recent advancements in diffusion-based modeling for iterative generation of the entire sequence.

3.1.1 Discrete autoregressive model

We use a vanilla decoder-only transformer, to autoregressively predict the next token of a pitch sequence. In this

task, the pitch values f are considered to be discrete tokens in a vocabulary V , each mapped to an embedding vector of size d through an embedding matrix $E \in R^{|V| \times d}$. The model is trained with cross-entropy loss.

3.1.2 Continuous diffusion model

We use a simple yet effective diffusion variant, Iterative α -Deblending (IADB) [35] as the training objective of our model that generates finely quantized pitch f . IADB defines a simplified diffusion process that is a linear interpolation between noise $x_0 \sim X_0 = \mathcal{N}(0, 1)$ and data $x_1 \sim X_1 = X_{data}$:

$$x_\alpha = (1 - \alpha)x_0 + \alpha x_1. \quad (2)$$

We leverage a deterministic iterative deblending process proposed in [35] to sample a data point $x_1 \sim X_1$ from noise $x_0 \sim X_0$. With the total number of iterations in the process as T , and given a time step $t \in \{0, 1, 2, \dots, T\}$, we define the blending parameter $\alpha_t = \frac{t}{T}$ and an α -blended point x_{α_t} . Thus, the iterative deblending is defined as:

$$x_{\alpha_{t+1}} = (1 - \alpha_{t+1})\bar{x}_0 + \alpha_{t+1}\bar{x}_1, \quad (3)$$

where $(\bar{x}_0, \bar{x}_1) = E_{(X_0 \times X_1)|x_{\alpha_t}, \alpha_t}$ is the expected value of the posterior samples given x_{α_t}, α_t . Heitz et. al. [35] show that using expected posteriors \bar{x}_0, \bar{x}_1 in the deblending process (Eq. 3) instead of x_0, x_1 converges to the same point, while making the sampling process deterministic.

Taking the derivative of x_{α_t} with respect to the blending parameter α_t , the training objective becomes,

$$D_\theta(x_{\alpha_t}|\alpha_t) \approx \frac{dx_{\alpha_t}}{d\alpha_t} = (\bar{x}_1 - \bar{x}_0), \quad (4)$$

Taking a trained model D_θ , we perform an iterative sampling procedure to generate outputs:

$$x_{\alpha_{t+1}} = x_{\alpha_t} + (\alpha_{t+1} - \alpha_t)D_\theta(x_{\alpha_t}, \alpha_t), \quad (5)$$

3.2 Spectrogram Generator

On the next level of the hierarchy, we train a model to generate a spectrogram conditioned on pitch, which is then converted to an audio signal using a vocoder. This method uses IADB as described in Sec. 3.1.2, while additionally conditioned on singer and pitch. Each singer ID is embedded as a discrete vector, and the processed pitch is time-downsampled to match the spectrogram’s time axis. Both conditioning signals are concatenated as additional channels to the mel-spectrogram input. Thus given a conditioning signal c , the training objective $D_\phi(x_{\alpha_t}|\alpha_t, c)$ is similar to Eq. 4 but is additionally conditioned on c .

The singer and pitch values are conditioned using classifier-free guidance (CFG) [36]. Given a conditioning strength w , CFG is implemented such that $\overline{D_\phi}(x_{\alpha_t}|c)$ is used during the iterative sampling, defined as,

$$\overline{D_\phi}(x_{\alpha_t}|\alpha_t, c) = (1-w)D_\phi(x_{\alpha_t}|\alpha_t) + wD_\phi(x_{\alpha_t}|\alpha_t, c) \quad (6)$$

4. EXPERIMENTS

In this paper, we consider the Spectrogram Generator as a tool to convert melodic ideas from the Pitch Generator into perceivable audio. As a result, we evaluate both the Generators with a focus on quality of pitch generation and the spectrogram’s fidelity in representing that pitch.

Through our experiments, we aim to motivate our choices for (1) a hierarchical approach to generation, (2) the use of pitch as an intermediate representation, through listening tests. We also qualitatively evaluate the overall melodic quality of generations. Additionally, we assess the Spectrogram Generator by testing pitch adherence: the ability of the model to reliably reproduce the pitch conditioning through quantitative and qualitative analyses. We leave evaluation of other aspects of the Spectrogram Generator such as audio quality, singer adherence to future work. Readers are encouraged to listen to relevant supplementary audio samples on our project website while going through this and the following sections.

4.1 Dataset

We use a combination of the Saraga and Hindustani Raga Recognition datasets [37, 38]. Audio files in the combined dataset contain audio of vocal performances including the tanpura, i.e. a drone, along with the melodic and rhythmic accompaniment across 56 unique singers. It spans about 120 hours across 362 audio files, where the files range from 88 seconds (s) to 1.2 hours with a median duration of 20 minutes. The dataset is randomly split into training and validation sets at a 90:10 ratio. Furthermore, each audio file is split into 60 s segments resulting in 7174 and 719 segments in the training and validation sets respectively. Due to different inductive biases in the models used, they all have different receptive fields and are thus trained on sequences with lengths varying from 8.2 s-12 s, randomly sampled from the 60 s segments during training.

The vocals are isolated using 2-stem source separation with HT Demucs [39] and further, the pitch is extracted using CREPE [40] and is sampled at 100 Hz. We algorithmically reduce the number of pitch detection errors using a loudness-based pitch filtering approach; using a sliding window to calculate area under the loudness curve, we retain only corresponding pitch values exceeding an empirically set threshold. We normalize the pitch to a logarithmic scale such that an arbitrarily chosen frequency, 440Hz is 0 on this scale, and quantize it into 10-cent bins. Additionally, during training, the pitch is transposed by a random multiple of 10 cents within a range of $[-400, 400]$ cents.

Artifacts in the dataset Our source separation model, HT Demucs [39], allows some leakage from other instruments including mainly the *sarangi* (stringed melodic accompaniment) and the *tabla* (rhythmic accompaniment) as artifacts in the vocal stem due to the out of distribution nature of Hindustani music data for the model. These ‘leaked’ sounds are generated in our models too (both our proposed model and the baselines established). Additionally, instances of speech are found in some generated samples as it is present in our dataset. The Carnatic FTA-Net

[41], presents a domain-informed model trained to extract pitch contours from Carnatic vocal audio. Owing to the similarities between Carnatic and Hindustani music, an interesting direction for future work would be to adopt their methodologies in our data processing pipeline.

4.2 Model Architectures

Below we present model specific architectures and data preprocessing for the Pitch Generators (Autoregressive and Diffusion) and the Spectrogram Generator.

Pitch Generator (Discrete Autoregressive) This model was trained on 12s (1200 token) sequences. The quantized pitch f is converted into a sequence of discrete embedding vectors e , using an embedding space $E \in \mathbb{R}^{|V| \times d}$ where effective vocabulary size is $|V| = 796$ and embedding dimension is $d = 512$. The model is a decoder-only transformer [23] with 8 layers, with each layer having an output dimension of 512. AliBi positional method [42] is used to encode the position of tokens in the sequence. A cosine learning schedule with linear warm-up is used. Samples are generated with a temperature of 0.99 and using top k sampling with $k=40$.

Pitch Generator (Continuous Diffusion) This model was trained on 10.24s (1024 elements) sequences. The quantized pitch contour is limited to a range of 400 integers. This distribution is converted into a continuous Gaussian using the quantile function which maps a variable’s probability distribution to another probability distribution. This model is implemented as a U-Net with three down-sampling and up-sampling layers each with a stride of 4, 2 and 2 respectively. Each layer is made of four 1-D convolution layers with weight normalization [43] and Mish non-linearity [44]. The bottleneck involves 4 attention layers with 8 heads each.

Spectrogram Generator This model is trained on 8.2s (512 elements) of mel-spectrogram sequences. The relevant pitch conditioning is linearly interpolated and down-sampled to match the sequence length of the spectral data. The spectral data is produced with 192 mels and a hop size of 256 (0.016 s) given 16 kHz audio and is converted to a continuous Gaussian distribution using the quantile transform function as well. Apart from additional channels for singer and pitch conditioning, the architecture is the same as that used by the Pitch Generator (Continuous Diffusion) (Sec 4.2). For simplicity, spectrograms are converted to audio using the Griffin-Lim algorithm [45]. Future work could harness the power of recent developments in neural vocoders including HiFi-GAN [46].

4.2.1 Conditioning signals

In addition to pitch, the Spectrogram Generator utilizes singer conditioning to help maintain the consistency of the voice in generated audio as seen in the supplementary audio samples. Each singer is assigned a unique ID and mapped to an embedding vector of size $d_{singer} = 128$. Conditioning was implemented with CFG as discussed in Sec. 3.2 with a strength of $w = 3$ for pitch and singer conditioning. This value was determined based on empirical

studies as an optimal balance between fidelity to pitch and minimizing artifacts due to incorrect pitch extraction.

4.3 Baseline Models

Through our baseline models, we aim to motivate two major architectural choices: (1) hierarchy in the model and (2) an intermediate pitch representation. These models thus include a non-hierarchical baseline, a hierarchical baseline with a self-supervised intermediate representation (hierarchical Encodec baseline), and the ground truth.

Non-hierarchical Baseline In this baseline, we highlight a naive approach of modeling audio directly with no hierarchy. We train a diffusion model with the IADB objective directly on processed audio mel-spectrograms. The model architecture is similar to other diffusion models used in this paper (Sec. 4.2) and was trained on the same dataset as our model with sequences of length 8.2s.

Hierarchical Encodec Baseline We train a hierarchical autoregressive baseline on a self-supervised intermediate representation, Encodec [26]. Through this model, we aim to compare the effect of self-supervised and pitch intermediate representations. To this end, we train MSPrior [47, 48], a decoder-only transformer adapted for real-time use, on Encodec tokens [26] extracted using the 24 kHz Encodec model with a target bandwidth of 3 kbps (4 channels per token). This model was trained on only the Hindustani Raga Recognition Dataset (which constitutes about $\frac{5}{6}$ th of our dataset) with a sequence length of 900 (12 s). We use a temperature of 0.99 for sampling.

Ground Truth To set the gold standard of melodic quality, we use ground truth pitch for comparison. As the listening test focuses on evaluating the Pitch Generator, we standardize audio quality across all models (except the hierarchical Encodec baseline which already generates waveform) by synthesizing the ground truth pitch with our Spectrogram Generator. We use five singers (3 low and 2 high voice range) with reasonable representation in the dataset as singer conditioning. Depending on the range of the generated pitch, we randomly select from the appropriate set of singers to generate audio for the contour.

4.4 Human Evaluation on Melodic Quality

To evaluate the musical quality and characteristics of generated samples, we conduct a listening study and offer qualitative observations supported by audio examples in our supplementary material.

Listening study We compare five systems: non-hierarchical baseline, hierarchical Encodec baseline, autoregressive and diffusion variants of our method, and ground truth. Participants were presented with 8.2 s audio samples, from two random systems and asked to rate which one is more musically interesting, on a 5-point Likert scale. We recruited 15 participants who are trained in Hindustani or Carnatic music. Although Carnatic music is stylistically different from Hindustani music, the two share the context of raga and tala giving participants enough context to evaluate samples for this study. Participants' primary instruments were the voice or other melodic instruments includ-

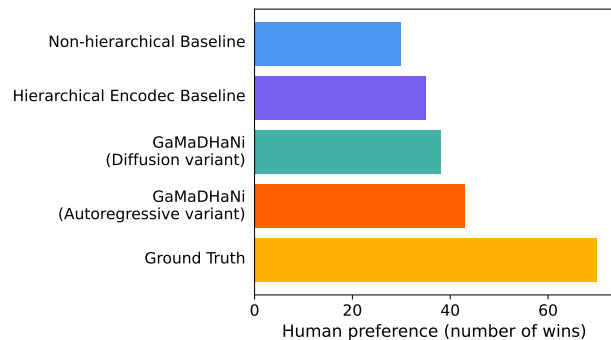


Figure 3. Results from the listening study, showing how many times each system was preferred.

ing the harmonium, sarangi, sarod, sitar, flute, or violin. We collected 240 ratings, with each system involved in 96 comparisons.

Results Fig. 3 shows the number of wins in each system. We ran a Kruskal-Wallis H test and confirmed that there are statistically significant pairs among the combinations. According to a post-hoc analysis using the Wilcoxon signed-rank test with Bonferroni correction (with $p < 0.05/10$), we find that our hierarchical model with an autoregressive Pitch Generator outperforms the non-hierarchical baseline. Given the small sample size, we also compare all systems against each other by aggregating ratings and considering them as independent samples. Using the Independent (Mann-Whitney U) test with Bonferroni correction, we find that both our models, discrete autoregressive and continuous diffusion outperform the non-hierarchical baseline significantly. Through these experiments, we establish that our model outperforms the non-hierarchical baseline.

Diversity in Generation Participants did not prefer our methods significantly more than the hierarchical Encodec baseline. This baseline tends to hold a single note or move through a few stable notes without much dynamism. This understandably was preferred by participants as *vilambit alap* or slower improvisation, a common way to establish a raga in Hindustani music, involves the use of such long and stable notes. With only 8.2 s duration audio samples, the listeners do not have enough time to notice the lack of dynamic movement. In contrast, our proposed methods can render both slow and fast movements, resulting in more variety as seen in generated samples. We hypothesize that this could be due to the different intermediate representations of both models, i.e. due to the importance of intricate melodic movements, a model trained to explicitly generate fine pitch would be able to capture melodic complexity.

Consistency of vocal timbre We note that generations from the hierarchical model, which includes singer conditioning, display more consistency in the timbre of voice; the baseline models sometimes abruptly switch vocal timbre in the middle of generation.

4.5 Pitch Adherence in Spectrogram Generator

Although the Spectrogram Generator loss lacks an explicit term for pitch adherence, we evaluate it by calculating the

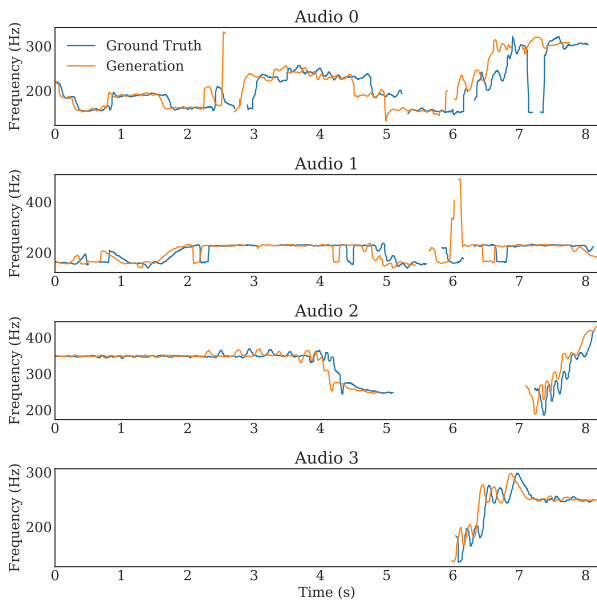


Figure 4. Examples of ground truth pitch (blue) and extracted pitch contour from the generated sample (orange) to highlight pitch adherence with low and high correlation, r (top to bottom). **Low correlation:** Audio 0 ($r = 0.1$) and 1 ($r = 0.11$) are examples of errors in pitch detection. **High correlation:** Audio 2 ($r = 0.94$) and 3 ($r = 0.99$)

Pearson correlation coefficient between the conditioning pitch and the pitch extracted from the generated audio. For this, we choose four singers (two male and two female) to generate audio conditioned on 32 random contours from the validation set resulting in a total of 128 contours to evaluate. We achieve a mean correlation of 0.71 between input and loudness-filtered extracted pitch.

Visual inspection reveals that differences between the input and extracted pitch sequences are pronounced when artifacts due to errors in pitch detection, source separation, or ground truth are present in either sequence. We present instances of samples with high and low correlation in Fig. 4. In addition, we note an inconsistent difference in timing between the ground truth and generated contour in Fig. 4. Future work could investigate pitch-specific training objectives and alternative conditioning representations to improve the precision of the generated audio’s pitch in time. Overall, based on visual analysis, we note that our model faithfully reconstructs the pitch conditioning shape.

5. INTERACTION USE CASES

We show two interactive use cases of GaMaDHaNi: (1) continuing an input melodic sequence or ‘prime’, and (2) guiding generation with coarse solfege-like notation.

5.1 Primed Generation

We investigate using our model for melodic sequence continuation. To this end, we input a four-second pitch sequence from our dataset termed ‘prime’ into our Pitch Generator, and ask the model to continue the sequence. The model can generate realistic-sounding continuations with

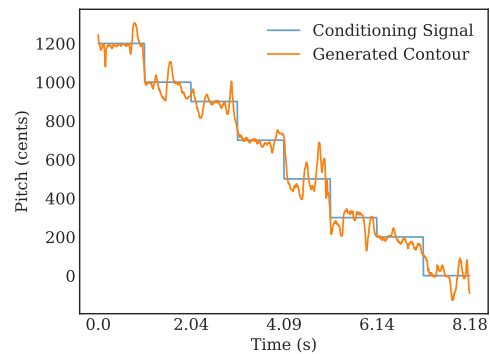


Figure 5. A staircase descending scale (in blue) as a coarse input. This input is then processed as described in Sec. 5.2 and fed into the model. The generated fine-grain contour (in orange) has glides (mindh) and fast jerky movement (gamak) characteristic to Hindustani music.

interesting patterns, as seen in Fig. 2 and in our audio samples. Future work could involve creating an interactive pipeline that would allow our model to directly take input from the user allowing a human-machine collaboration.

5.2 Coarse Pitch Conditioning

To explore further possibilities for interaction, we evaluate the model’s ability to adhere to solfege-like conditioning given to the Pitch Generator. To this end, a ‘coarse pitch’ signal is inferred by calculating a moving average of the pitch with a window size of 1s and a hop size of 0.01s. The Pearson correlation coefficient between the input and generated coarse pitch is 0.97, and between the ground truth and generated pitch is 0.79. Both values are averaged over 64 random samples from the validation set. Thus solfege input, once converted into a similar coarse pitch signal, can be used to guide the model’s generation as seen in Fig. 5, where the model renders a solfege-based descending scale into realistic-sounding audio. Although simple, this is an interesting avenue for interactive generation that we plan to explore in the future.

6. CONCLUSION

We present a modular hierarchical system to generate melodically rich Hindustani vocal audio using a relatively small dataset. Our model has comparable or better performance than established baselines while including an interpretable intermediate pitch representation. We present interesting forms of interaction including primed generations and coarse pitch conditioning that could be developed further to achieve interactive human-machine music making.

There are interesting future directions such as the use of tonic, raga and rhythmic aspects as conditioning for generation. Additionally, the Spectrogram Generator could adopt more advanced vocoders and conditioning signals such as loudness and phoneme features for better results.

7. ETHICS STATEMENT

This work, to our knowledge, is the first model trained to explicitly generate Hindustani vocal music and thus we find it important to emphasize that this work is intended to foster human-AI collaboration, creating a more accessible environment for creative exploration and is by no means intended to replace music teachers or musicians. While we acknowledge the ethical concerns involved in modeling singing voices, we include singer conditioning in our approach with the sole intention of maintaining voice consistency in the generated samples. Additionally, we note that this work utilizes datasets contributed by artists or institutes holding distribution rights to ensure responsible use with informed consent. These datasets were released with appropriate permissions to process audio recordings for research purposes. However, despite our current model's limited scope, future enhancements may pose a risk of mimicking the identities of existing singers, necessitating the establishment of protective guidelines for artists.

8. ACKNOWLEDGMENT

We thank all of our listening study participants for their invaluable contributions and insights. We also appreciate their promptness in completing the tests, which greatly facilitated this work.

9. REFERENCES

- [1] W. Van der Meer, *Hindustani music in the 20th century*. Martinus Nijhoff Publishers, The Hague, 1980.
- [2] A. Agostinelli, T. I. Denk, Z. Borsos, J. Engel, M. Verzetti, A. Caillon, Q. Huang, A. Jansen, A. Roberts, M. Tagliasacchi *et al.*, "Musiclm: Generating music from text," *arXiv preprint arXiv:2301.11325*, 2023.
- [3] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Défossez, "Simple and controllable music generation," in *Proc. of the Advances in Neural Information Processing Systems*, 2024.
- [4] V. Vidwans, "Computational music," accessed on 2024-4-11. [Online]. Available: <https://computationalmusic.com/>
- [5] D. Das and M. Choudhury, "Finite state models for generation of hindustani classical music," in *Proc. of the International Symposium on Frontiers of Research in Speech and Music*, 2005, pp. 59–64.
- [6] H. V. Sahasrabudde, "Analysis and synthesis of hindustani classical music," accessed: 2024-07-22. [Online]. Available: https://www.cse.iitb.ac.in/~hvs/paper_1992.html
- [7] S. Gopi and F. William, "Introductory studies on raga multi-track music generation of indian classical music using ai," *The International Conference on AI and Musical Creativity*, 2023.
- [8] S. Adhikary, M. S. M, S. S. K, S. Bhat, and K. P. L, "Automatic music generation of indian classical music based on raga," in *Proc. of the IEEE International Conference for Convergence in Technology (I2CT)*, 2023.
- [9] A. Vidwans, K. K. Ganguli, and P. Rao, "Classification of Indian classical vocal styles from melodic contours," in *Proc. of the CompMusic Workshop*, 2012.
- [10] S. Gulati, J. Serra, V. Ishwar, and X. Serra, "Mining melodic patterns in large audio collections of indian art music," in *Proc. of the International Conference on Signal-Image Technology and Internet-Based Systems*. IEEE, 2014.
- [11] K. K. Ganguli, A. Rastogi, V. Pandit, P. Kantan, and P. Rao, "Efficient melodic query based audio search for hindustani vocal compositions." in *Proc. of the International Society for Music Information Retrieval (ISMIR)*, 2015.
- [12] T. Nuttall, G. Plaja-Roglans, L. Pearson, and X. Serra, "In search of sañcaras: Tradition-informed repeated melodic pattern recognition in carnatic music," in *Proc. of the International Society for Music Information Retrieval (ISMIR)*, 2022.
- [13] P. Chordia and S. Şentürk, "Joint recognition of raag and tonic in north indian music," *Computer Music Journal*, vol. 37, no. 3, pp. 82–98, 2013.
- [14] S. Gulati, J. Serra, V. Ishwar, S. Sentürk, and X. Serra, "Phrase-based rāga recognition using vector space modeling," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016.
- [15] M. Clayton, P. Rao, N. N. Shikarpur, S. Roychowdhury, and J. Li, "Raga classification from vocal performances using multimodal analysis." in *Proc. of the International Society for Music Information Retrieval (ISMIR)*, 2022.
- [16] P. Chordia and S. Şentürk, "Joint recognition of raag and tonic in north indian music," *Computer Music Journal*, vol. 37, no. 3, pp. 82–98, 2013.
- [17] G. K. Koduria, S. Gulatia, P. Rao, and X. Serra, "Rāga recognition based on pitch distribution methods," *Journal of New Music Research*, 2012.
- [18] S. K. Subramanian, "Modelling gamakas of carnatic music as a synthesizer for sparse prescriptive notation," Ph.D. dissertation, Master's thesis, National University of Singapore, 2013.
- [19] H. Ranjani, A. Srinivasamurthy, D. Paramashivan, and T. V. Sreenivas, "A compact pitch and time representation for melodic contours in indian art music," *The Journal of the Acoustical Society of America*, vol. 145, no. 1, pp. 597–603, 2019.

- [20] V. S. Viraraghavan, A. Pal, H. Murthy, and R. Aravind, "State-based transcription of components of carnatic music," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.
- [21] V. Vidwans, *The Music of Minds and Machines*. FLAME University, Pune, 2023.
- [22] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.
- [24] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. of the Advances in Neural Information Processing Systems*, 2014.
- [25] Y. Wu, E. Manilow, Y. Deng, R. Swavely, K. Kastner, T. Cooijmans, A. Courville, C.-Z. A. Huang, and J. Engel, "Midi-ddsp: Detailed control of musical performance via hierarchical modeling," in *Proc. of the International Conference on Learning Representations*, 2022.
- [26] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, "High fidelity neural audio compression," *arXiv preprint arXiv:2210.13438*, 2022.
- [27] N. Zeghidour, A. Luebs, A. Omran, J. Skoglund, and M. Tagliasacchi, "Soundstream: An end-to-end neural audio codec," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 495–507, 2021.
- [28] A. Agostinelli, T. I. Denk, Z. Borsos, J. Engel, M. Verzetti, A. Caillon, Q. Huang, A. Jansen, A. Roberts, M. Tagliasacchi *et al.*, "Musiclm: Generating music from text," *arXiv preprint arXiv:2301.11325*, 2023.
- [29] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Simultaneous modeling of spectrum, pitch and duration in hmm-based speech synthesis," in *Proc. of the European conference on speech communication and technology*, 1999.
- [30] Y. Fan, Y. Qian, F.-L. Xie, and F. K. Soong, "Tts synthesis with bidirectional lstm based recurrent neural networks," in *Proc. of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2014.
- [31] H. Li, Y. Kang, and Z. Wang, "Emphasis: An emotional phoneme-based acoustic model for speech synthesis system," in *Proc. of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2018.
- [32] P. Lu, J. Wu, J. Luan, X. Tan, and L. Zhou, "Xiaoic-eSing: A High-Quality and Integrated Singing Voice Synthesis System," in *Proc. of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2020.
- [33] Y. Yi, Y. Ai, Z. Ling, and L. Dai, "Singing voice synthesis using deep autoregressive neural networks for acoustic modeling," in *Proc. of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2019.
- [34] M. Morrison, Z. Jin, J. Salamon, N. J. Bryan, and G. J. Mysore, "Controllable Neural Prosody Synthesis," in *Proc. of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2020.
- [35] E. Heitz, L. Belcour, and T. Chambon, "Iterative α -(de) blending: A minimalist deterministic diffusion model," in *Proc. of the ACM SIGGRAPH Conference*, 2023, pp. 1–8.
- [36] J. Ho and T. Salimans, "Classifier-free diffusion guidance," in *Proc. NeurIPS Workshop on Deep Generative Models and Downstream Applications*, 2021.
- [37] A. Srinivasamurthy, S. Gulati, R. C. Repetto, and X. Serra, "Saraga: open datasets for research on indian art music," *Empirical Musicology Review*, vol. 16, no. 1, pp. 85–98, 2021.
- [38] S. Gulati, J. Serrà Julià, K. K. Ganguli, S. Sentürk, and X. Serra, "Time-delayed melody surfaces for rāga recognition," in *Proc. of the International Society for Music Information Retrieval (ISMIR)*, 2016.
- [39] S. Rouard, F. Massa, and A. Défossez, "Hybrid transformers for music source separation," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023.
- [40] J. W. Kim, J. Salamon, P. Li, and J. P. Bello, "Crepe: A convolutional representation for pitch estimation," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.
- [41] G. Plaja-Roglans, T. Nuttall, L. Pearson, X. Serra, and M. Miron, "Repertoire-specific vocal pitch data generation for improved melodic analysis of carnatic music," *Transactions of the International Society for Music Information Retrieval*, vol. 6, no. 1, pp. 13–26, 2023.
- [42] O. Press, N. A. Smith, and M. Lewis, "Train short, test long: Attention with linear biases enables input length extrapolation," in *Proc. of the International Conference on Learning Representations*, 2021.
- [43] T. Salimans and D. P. Kingma, "Weight normalization: A simple reparameterization to accelerate training of deep neural networks," in *Proc. of the Advances in Neural Information Processing Systems*, 2016.

- [44] D. Misra, “Mish: A self regularized non-monotonic activation function,” in *Proc. of the British Machine Vision Conference*, 2020.
- [45] D. Griffin and J. Lim, “Signal estimation from modified short-time fourier transform,” *IEEE Transactions on acoustics, speech, and signal processing*, vol. 32, no. 2, pp. 236–243, 1984.
- [46] J. Kong, J. Kim, and J. Bae, “Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis,” in *Proc. of the Advances in neural information processing systems*, 2020.
- [47] A. Caillon, “Msprior,” <https://github.com/caillonantoine/msprior>, 2023.
- [48] —, “Hierarchical temporal learning for multi-instrument and orchestral audio synthesis,” Ph.D. dissertation, Sorbonne université, 2023.

SYMPAC: SCALABLE SYMBOLIC MUSIC GENERATION WITH PROMPTS AND CONSTRAINTS

Haonan Chen¹ Jordan B. L. Smith² Janne Spijkervet¹ Ju-Chiang Wang¹
Pei Zou¹ Bochen Li¹ Qiuqiang Kong³ Xingjian Du¹

¹ ByteDance Inc. San Jose, USA

² Queen Mary University of London

³ Department of Electronic Engineering, The Chinese University of Hong Kong

haonanchen@bytedance.com

ABSTRACT

Progress in the task of symbolic music generation may be lagging behind other tasks like audio and text generation, in part because of the scarcity of symbolic training data. In this paper, we leverage the greater scale of audio music data by applying pre-trained MIR models (for transcription, beat tracking, structure analysis, etc.) to extract symbolic events and encode them into token sequences. To the best of our knowledge, this work is the first to demonstrate the feasibility of training symbolic generation models solely from auto-transcribed audio data. Furthermore, to enhance the controllability of the trained model, we introduce SympAC (Symbolic Music Language Model with Prompting and Constrained Generation), which is distinguished by using (a) *prompt bars* in encoding and (b) a technique called *Constrained Generation via Finite State Machines (FSMs)* during inference time. We show the flexibility and controllability of this approach, which may be critical in making music AI useful to creators and users.

1. INTRODUCTION

The success of language models — especially large ones — has demonstrated that with more data and larger models, using a simple language model objective can endow a model with powerful natural language generation capabilities. On the other hand, although symbolic music and natural language share many similarities, no music model has yet seemed to match the capabilities of generative text models. One reason for this gap is the insufficient amount of symbolic music data.

To address this, previous efforts in symbolic music generation have involved combining limited manually annotated data with data obtained by automatic transcription [1], or collecting private symbolic training datasets [2]. By contrast, in this work, we demonstrate

that a high-quality, multi-track symbolic music generation model can be trained just using results from running Music Information Retrieval (MIR) models on audio music data. In this way, our framework eliminates the need for manually annotated symbolic music data, allowing for expansion purely through audio datasets.

On the other hand, there has been a recent surge of efforts that directly generate the auditory modality of music [3–5]. This is useful for some applications, but typically precludes fine-grained control and editing the outcome, which is crucial for composers who wish to shape their musical ideas precisely. In contrast, outputting symbolic data gives composers the ability to interactively shape and modify their musical ideas.

Considering such advantages, the problem of how to integrate user input to control the generation of symbolic music has been a popular research topic. In previous works, two methods for incorporating control signals are usually used. The first approach is based on a Variational Autoencoder (VAE) [6, 7], wherein the control is exerted within the VAE’s latent space. The second approach is to embed control information directly into the encoding of symbolic music and implant control inputs during inference [8–11].

In this work, we introduce the **SympAC** framework (**S**ymbolic Music Language Model with **P**rompting **A**nd **C**onstrained Generation), designed to work with decoder-only language models to enable user input controls. The SympAC framework consists of the following two parts. First, inspired by the prompting mechanism used in the natural language domain [12, 13], we introduce *prompt bars* in our symbolic music encoding, which consolidates all control signals into a separate prompt section before encoding the actual musical notes. This design is essential for a decoder-only language model to have the full context of control signals during the generation of music. Second, in the controlled symbolic music generation setting, the generated tokens should not only comply with the encoding grammar but also adhere to user inputs. Thus we propose to use *Constrained Generation via Finite State Machines (FSMs)*, which constrains the sampling of tokens at each time step to a subspace. We will discuss the advantages of SympAC over previous methods in Section 2, and provide more details of how SympAC can be used for various types of user inputs in Sections 3 and 4.



We collected roughly one million in-house audio samples and extracted MIR information for each, using pre-trained models for beat tracking [14], chord detection [15], section detection [16, 17], multi-track transcription [18], and music tagging [19]. The MIR results were transformed into various tokens, and then integrated into an extended REMI [10, 20] encoding to train a language model based on Llama [21] architecture. To summarize, our main contributions are:

Scalability: We demonstrate that a high-quality symbolic music generation model can be trained solely with transcribed data, without the need of manually annotated symbolic music, and can be scaled by amassing more audios.

Controllability: We propose the SympAC framework, which enables flexible user input controls on a decoder-only language model while retaining good quality.

2. RELATED WORK

2.1 Training Data For Symbolic Music

In Table 1, we summarize some popular music datasets in the symbolic and audio domains, together with our in-house audio dataset, and compare their sizes. The Lakh MIDI Dataset [24] is one of the biggest public datasets, containing 170K multitrack pieces in MIDI format. Many researchers use publicly available symbolic music datasets for training, but some collect and use large-scale ones that are not disclosed; e.g., MusicBERT [2] was trained on the Million-MIDI Dataset (MMD).

Although the combined size of the public datasets in Table 1 is large, combining them is not straightforward since they vary in format. For example, the Maestro dataset consists of transcriptions of piano performances where note timings reflect actual performance timings, whereas datasets like Lakh are quantized to metrical time with alignment to beats. The inclusion of instrument tracks and additional information (e.g., chords, sections) also differs between datasets. To expand the scale of training data by combining these datasets, it is necessary to unify their formats first, which may be tedious and introduce errors.

On the other hand, publicly available audio datasets are much larger in scale. The Million Song Dataset (MSD) [26], for example, contains 1M songs, or 709M notes in total after being run through a 5-track transcription model [18]. The recently published DISCO-10M [27] is of an even larger scale. Furthermore, by using a single set of MIR models to annotate all the audio data, we do not need to be concerned about the issue of inconsistent data formats. This makes it easier to scale up the training dataset.

2.2 Encoding For Symbolic Music

Since the introduction of the Music Transformer [28], language models based on the transformer architecture have become a popular choice for symbolic music generation. One of the most critical research questions has been how to encode symbolic music that is amenable to processing by such a model, which, in the context of language models, involves converting the piece into a sequence of tokens.

Early transformer-based models for symbolic music predominantly employed a MIDI-like encoding scheme, by treating MIDI event sequences almost identically as input token sequences [8, 9, 29]. Later, the Revamped MIDI (REMI) encoding [20] was proposed, which modified the MIDI encoding by replacing time shift events with duration events for each note and introducing bar and beat concepts to adopt metrical time instead of absolute time. These modifications facilitated the model’s learning of rhythmic patterns within the music, improving the quality of the output. Building upon REMI, several extensions have been proposed to support encoding multitrack [9] and various control tokens [10]. Our work is based on the multitrack REMI encoding, and given the MIR models we have, it incorporates control tokens such as genre, chord, and section tokens to the encoding.

2.3 Controllable Symbolic Music Generation

Previous methods for controlling symbolic music generation have typically fallen into two categories. The first is based on Variational Autoencoders (VAEs) [6, 7]. VAEs aim to find a latent space for representing music that encodes distinct musical attributes in independent dimensions. This disentanglement allows for specific attributes of generated music (e.g., rhythm, genre, or timbre) to be individually manipulated by altering corresponding dimensions in the latent space without affecting other attributes, thereby enhancing the controllability of music generation.

The second approach is to include control tokens in the encoding of symbolic music. For example, MMM [9] includes instruments and note density tokens in the encoding, which can be specified at inference. Similarly, FIGARO [10] uses “expert descriptions” indicating time signature, note density, mean pitch, mean velocity and mean duration as well as instruments and chords. It then uses an encoder-decoder model to learn a mapping from descriptions to sequences of a piece of music. Driven by the development of Large Language Models (LLMs), recent work has also explored using natural language to control symbolic music generation [30–33]. Natural language text can also be treated as control tokens, with the key distinction that it usually requires pre-training the LLM on text.

In our work, the proposed SympAC framework is designed to work with a decoder-only language model. In a controlled generation setting, prompt bars that conform with user input control signals are generated first. The generation of musical part comes after that, in which the model will have full context of control signals from prompt bars. These two generation stages are both controlled by an FSM, which takes into consideration the grammar of the encoding and user inputs. There are two main differences between SympAC and previous works

1. We encode control signals as tokens and use FSM to enforce input control signals during inference. In contrast, for VAE-based control methods, control signals are converted into latent embeddings, and the model is not guaranteed to follow these control signals.
2. Since we use a decoder-only language model, the to-

Dataset	#Songs	#Notes	Format	Multitrack	Public
Maestro [22]	1.1K	6M	MIDI	N	Y
GiantMIDI-Piano [23]	10.9K	39M	MIDI	N	Y
Lakh [24]	170K	910M	MIDI	Y	Y
MMD [2]	1.5M	2,075M	MIDI	Y	N
FMA [25]	100K	N/A *	Audio	Y	Y
MSD [26]	1M	709M	Audio	Y	Y
DISCO-10M [27]	15M	N/A *	Audio	Y	Y
In-House Dataset (IHD)	1M	3,688M	Audio	Y	N

Table 1: Comparison of different symbolic and audio music datasets. * Since we did not run transcription on FMA or DISCO-10M, we don’t have the number of notes information for them.

kens in prompt bars are also learned simultaneously. Consequently, the user is only required to input a portion of the control information, with the model being able to automatically generate missing controls. In contrast, an encoder-decoder framework like the one described in [10] would require a complete encoder input during inference, which lacks flexibility.

3. METHOD

3.1 Symbolic Music Encoding And Prompt Bars

Our data representation is based on the REMI+ [10] representation, an extension of REMI [20] that supports multitrack data. An illustration of our encoding is shown in Fig. 1. The fundamental unit of our encoding is a bar, of which there are two types: *prompt bar* and *song bar*. The token sequence of a song bar can be divided into four parts:

- The *meta* part includes four tokens for the bar, `genre`, `sec` (for section type name), and `bpm_level` (which indicates the tempo range).
- The *chord* part consists of alternating `position` and `chord` tokens.
- Each *instrument track* part consists of a `track` token, followed by one or more groups of `position`, `duration` and `pitch` tokens.
- The *drum track* part consists of a `track<drum>` token, followed by one or more groups of `position` and `drum` (drum MIDI) tokens.

Here are further explanations of `position`, `duration` and `track` tokens¹:

- `position`: Represents the starting position of subsequent `chord`, `pitch` or `drum` token within a bar. Each bar is divided into 16 steps, so that position ranges from 0/16 to 15/16.
- `duration`: Ranges from the minimum time division of 1/16 bar to a maximum of 2 bars, or 32/16.
- `track`: A track token will only exist if there is at least one note in the bar for the corresponding instrument. This allows the user to control which instruments are used within a bar.

¹ Details of all token types are provided in supplementary materials

Prompt bars contain a subset of tokens in song bars, retaining only tokens that represent *control signals*. In our case, these include `genre`, `section`, `tempo`, `chords` and `tracks`. As future work, this encoding could be extended to include more control signals (e.g. note density for a track). The encoding of a full piece of music will consist of: all prompt bars in the piece; then, a special `end_of_prompt` token; then, all song bars in the piece; and finally a special `end_of_song` token.

During training stage, the model is trained to predict tokens in prompt bars as well, not distinguishing them from tokens in song bars. As mentioned previously, this design enables the user to input partial control signals (or no input at all), and the model is able to infer the missing ones.

Algorithm 1 Constrained Generation via FSM

```

1: procedure CONSTRAINEDSAMPLING( $\mathcal{M}, \mathcal{V}, \mathcal{R}$ )
2:    $s_0 \leftarrow x_0$  start token (bar in our encoding)
3:    $q_0 \leftarrow$  initial state
4:    $t \leftarrow 0$ 
5:   while not end of sequence do
6:      $\mathcal{V}_{t+1} \leftarrow$  GETSUBVOCAB( $\mathcal{R}, q_t, x_t$ )
7:      $q_{t+1} \leftarrow$  UPDATESTATE( $\mathcal{R}, q_t, x_t$ )
8:      $x_{t+1} \leftarrow$  SAMPLE( $\mathcal{M}, \mathcal{V}_{t+1}$ )
9:      $s_{t+1} \leftarrow s_t \circ x_{t+1}$ 
10:     $t \leftarrow t + 1$ 
11:  end while
12:  return  $s_t$ 
13: end procedure

```

3.2 Constrained Generation via FSM

In the controlled symbolic music generation setting, there are two types of constraints:

Grammar constraint: The encoding of symbolic music follows a specific format. For example, for our proposed encoding shown in Fig. 1, a bar token will always be followed by a `genre` token.

User input constraint: Generated token sequence should conform with user inputs. For example, if the user wants to generate “rock” style music, the `genre` token can only be `genre<rock>`.

Since we are already aware of these constraints in advance, there is no need to sample from the entire vocabulary space during inference. Instead, we can sample from

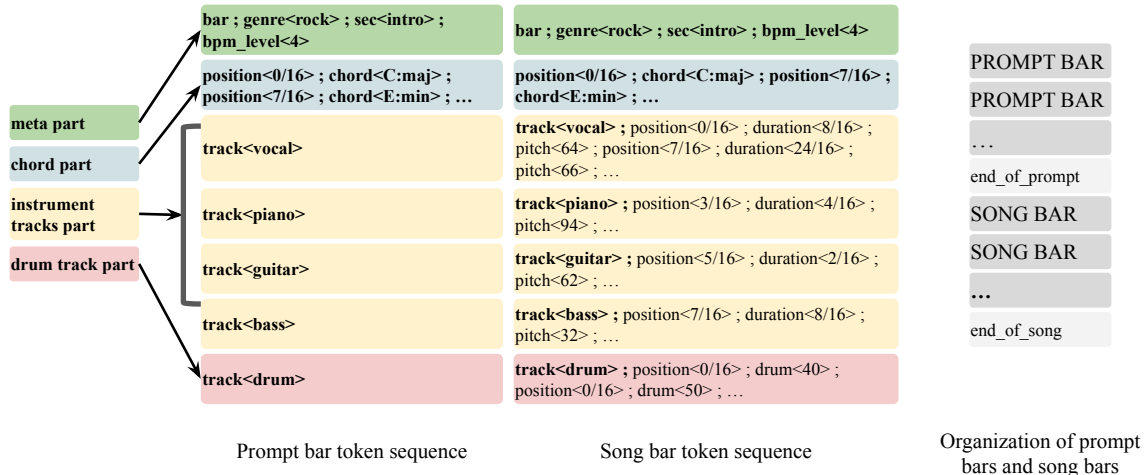


Figure 1: Illustration of our symbolic music encoding.

a subspace that is in accordance with the constraints.

To achieve this, we employ a Finite State Machine (FSM) to interact with the language model \mathcal{M} during inference. Let x_t denote the token generated by \mathcal{M} at time step t . The FSM takes x_t , the current state q_t and the predetermined rule set \mathcal{R} , and outputs a subset of the vocabulary \mathcal{V}_{t+1} , from which the language model \mathcal{M} can sample at time $t + 1$. We call this procedure Constrained Generation via FSM, which is formally defined in Algorithm 1. This algorithm is analogous to regular expression matching, where it checks if a given input string conforms to a specified pattern. Here the pattern and input string are equivalent to rule set \mathcal{R} and token sequence s_t respectively.

4. EXPERIMENTS AND RESULTS

To validate our contributions, we conduct experiments to assess whether the system is scalable (i.e., improves when scaling up training data) and controllable (i.e., there is consistency between generation output and user inputs).

In Sec. 4.3, we conduct a quantitative analysis to compare models trained on different amounts of training data, in order to assess scalability. In Sec. 4.4, we examine two common types of control inputs: chord progression and section structure. The impact of these control inputs is tested through both quantitative metrics and qualitative examples. Lastly, in Sec. 4.5, we compare our models trained on different datasets with other baseline symbolic music generation systems through subjective evaluation.

4.1 Datasets

We use three datasets in our experiments. We always use each dataset individually; i.e., we never merge the datasets to train a single model. The datasets are:

Lakh MIDI Dataset (LMD) [24]. A dataset in MIDI format, containing around 170K songs. We use this to compare with models trained on transcribed audio data.

Million Song Dataset (MSD) [26]. A public dataset used extensively by MIR researchers. We use the 30–60s pre-view audio clips, representing the highlight of the song.

In-House Dataset (IHD). We use a licensed internal collection with about 1M full songs in audio format, covering a wide range of Western modern genres.

4.2 Training Settings

We train a decoder language model with the Llama [21] architecture. We set the number of layers, number of attention heads and embedding dimensions to be 12, 12 and 768 respectively, resulting in a model with about 86M trainable parameters. We concatenate token sequences of all pieces into a 1-D array, and randomly pick a window of size 10,240 as one training sample. As the average sequence lengths of LMD, MSD and IHD are 900, 1500 and 8000 respectively, this window size would contain 11.4, 6.8 and 1.3 pieces on average for each dataset.

When training data are limited, data augmentation and data filtering (to ensure that unusual data do not pollute the training) are commonly used. However, we adopt neither approach, for two reasons. First, since we have a large dataset of audio samples, the training data are likely to cover a broad spectrum of examples already, reducing the need to filter out unusual data points. Second, augmentation may alter the training data in unwanted ways. For example, a common augmentation approach is to transpose all the pitches in a piece [11, 28]. However, this may distort the pitch ranges of each instrument: e.g., if the input bass parts are transposed up and down, the model will not learn the correct range of realistic bass notes.

Metric Class	IHD 100%	IHD 10%	IHD 1%
Chord	0.112	0.119	0.347
Structure	0.348	0.220	0.786
Vocal Note	0.416	0.892	1.086
Guitar Note	0.222	0.257	0.397
Piano Note	0.178	0.403	0.686
Bass Note	0.180	0.867	1.038
Drum Note	0.650	2.902	1.248

Table 2: Average Kullback-Leibler Divergence (KLD) of metrics in different metric classes for models trained on different dataset against a held-out validation set.

4.3 Unconditioned Generation

Intuitively, increasing the amount of data should enhance the performance of the model. In this experiment, we use objective metrics to validate this. Designing objective metrics to evaluate symbolic music remains an open question. A common approach is to prepare a reference dataset, calculate embeddings or metrics of the generated samples and reference set, and then compare these using distance metrics such as the Fréchet Distance or Kullback-Leibler Divergence (KLD). For a detailed review on evaluation methods for symbolic music, see [34, 35].

In our experiment, we prepare a held-out validation set with 3000 samples. We use a range of metrics that can be categorized into the following classes: chord, structure, instrument note (including vocals, guitar, piano and bass) and drum note. Detailed definitions are provided in supplements. In general, the metrics in each class are as follows:

- **Chord:** chord label, chord root, chord transition;
- **Structure:** section label, section label bigram, instrument labels per bar;
- **Instrument Note:** note pitch, note duration, pitch class, min/max pitch per bar, max number of notes per bar, uniformity of number of notes per bar;
- **Drum Note:** drum key, max number of notes per bar, uniformity of number of notes per bar, and unique drums per bar.

We compare models trained on 100%, 10% and 1% of the IHD data, and do generation in an unconditioned setting. For each model, we generate 800 samples to compute metric distributions. KLD values are then computed between distributions of generated samples and distribution of the validation set for each metric. Lower KLD indicates that two distributions are closer, suggesting the generated samples sound more similar to the validation set. We report the average KLD values for the same class, and provide a full list of KLDs for each metric in supplements.

The results are shown in Table 2. We can see that the model trained with 100% IHD data has the lowest KLD against the validation set on 6 out of 7 classes, and the model trained on only 1% data has the highest KLD on 6 out of 7 classes. The results confirm that a model trained on more data can generate samples closer to the training data. Furthermore, we observed that the benefit of using more data is greater for the 'Note' metrics than for the 'Chord' or 'Structure' ones. This is likely because note tokens are more numerous and have complex distributions, which needs larger scale of data to learn. Counterintuitively, the KLD for 'Structure' was better when using 10% of the data instead of 100%. We speculate that since the structure tokens are scarcest, this could be the result of a lucky alignment between the validation set at the 10% of the data used, but this deserves more study.

4.4 Controlled Generation

The SymPAC framework aims to give users flexible control over the music generation process. However, we need to

verify that this control is effective: do the notes generated agree with the control inputs? To this end, we conduct controlled generation experiments on two input scenarios: chord progression inputs and section structure inputs.

Chord Progression Inputs. In this experiment, we randomly pick 20 top trending chord progressions from *Hook-Theory*² as the chord progression inputs. We only include major and minor triad chords. We then let the model generate 64 bars of music by looping the chord progressions. To evaluate the match between the input chord progression and the output, we apply a symbolic chord detection method on the generated samples. Details about the method can be referred in the supplementary materials.

The accuracy of detected chord from the input chord progression is shown in Table 4. As shown in the result, the models trained on MSD, IHD 100% and IHD 10% all have similar overall accuracy, with MSD slightly outperforming the others. But the model trained on IHD 1% (just 10K songs) is much worse than the other three. This suggests that a dataset at the scale of 100K songs is sufficient to model low-level control signals like chord, given the model and encoding we are using here. We also provide examples in supplementary audios of outputs when given unusual chord progressions.

Section Structure Inputs. In this experiment, we take 10 typical section sequences as inputs (listed in supplements), ranging in length from 4 to 13 sections (16 to 68 bars), and use each model to generate 100 outputs per prompt. We compare the same 4 models from the previous section. For each generated output, we leverage a Music Structure Analysis (MSA) algorithm [36] to predict its structure, and compare this to the input structure. The MSA algorithm's predictions may be inaccurate, but we still expect that a greater match between the intended and estimated structure indicates more success at controlling the structure. We use Foote's algorithm [37] for segmentation and the 2D-Fourier magnitude algorithm [38] for section labeling, with a beat-wise feature embedding that averages the pitch-wise MIDI piano rolls within a beat interval. We evaluate the results using `mir_eval` [39], and report three metrics: boundary prediction f -measure with a 3-second tolerance (HR3F); pairwise clustering f -measure (PWF); and the normalized entropy score f -measure (Sf). To test directly how similar the repeated sections are, we also report PWF and Sf when the ground-truth segmentation is used.

We find that all metrics are worse (lower) when the system is trained on MSD or IHD 1%, and improve substantially when at least 10% of the data are used (Table 5). This is expected, since the audio clips in MSD are only excerpts and thus not instructive for modelling full-song structure.

Fig. 2 shows the piano roll of a typical output, where the match between the intended and predicted structure was average (Sf = 0.508). Even so, the match between the intended and realized structure is evident in the piano roll: the chorus sections are similar but not identical to each other, and so are the verse sections.

In both controlled generation experiments, the gap be-

² <https://www.hooktheory.com>

Model	Coherence	Richness	Arrangement	Structure	Overall
FIGARO	3.12 ± 0.82	2.73 ± 0.92	2.85 ± 0.96	2.62 ± 0.80	2.74 ± 0.89
MMT	2.37 ± 0.35	2.27 ± 0.36	2.37 ± 0.34	2.08 ± 0.30	2.16 ± 0.35
Ours (IHD)	3.55 ± 0.53	3.58 ± 0.38	3.45 ± 0.49	3.73 ± 0.32	3.60 ± 0.39
Ours (LMD)	3.25 ± 0.34	3.25 ± 0.35	3.28 ± 0.30	3.20 ± 0.61	3.25 ± 0.46
Ours (MSD)	3.16 ± 0.27	3.17 ± 0.33	3.09 ± 0.32	3.15 ± 0.29	3.07 ± 0.28

Table 3: Results of subjective evaluation, mean opinion score (MOS)

Training Dataset	Accuracy
IHD 100%	87.2%
IHD 10%	86.9%
IHD 1%	74.0%
MSD	87.6%

Table 4: Accuracy of chord progressions in controlled generation with chord input.

Dataset	Regular			Oracle	
	HR3F	PWF	Sf	PWF	Sf
IHD 100%	0.60	0.50	0.50	0.72	0.80
IHD 10%	0.60	0.49	0.49	0.70	0.79
IHD 1%	0.54	0.47	0.47	0.62	0.73
MSD	0.57	0.47	0.47	0.63	0.74

Table 5: Accuracy of structure predicted from generated songs with no guidance (left) and with ground truth segmentation (right).

tween 100% and 10% IHD is very small, indicating that 10% IHD data combined with SymPAC is sufficient for achieving good adherence to control inputs. However, it is important to remember that the metrics of these two experiments only reflect whether the control signals are well-followed, not the overall quality of the generated pieces.

4.5 Subjective Evaluation

The models tested so far were all trained on transcribed audio data, so it is worth comparing with models trained directly on MIDI data. In this experiment, we compare our model trained on LMD, MSD and IHD, and also two baselines, FIGARO [10] and MMT [11], in a subjective listening test. We recruited 12 participants with the background of MIR researchers or music producers. Similar to [11], we asked each participant to rate 10 audio samples generated by each model on a 5-point Likert scale on five criteria: coherence, richness, arrangement, structure and overall ³.

The result is summarized in Table 3. All of our proposed models outperform the baselines in all dimensions. Our model trained on IHD has higher performance than the other two training data setups, which attests to the viability of leveraging audio data by running MIR models

³ These criteria are described as: (1) Coherence: The rhythm is stable; The chord progression develops logically; Dissonant notes are not excessive. (2) Richness: The melody and accompaniment are interesting and diverse. (3) Arrangement: Collaboration among multiple instruments is harmonious and natural; Arrangements for different instruments are diverse and reasonable. (4) Structure: The piece includes a clear and engaging structure with appropriate repetitions and variations; The piece has obvious connections and reasonable developments between sections. (5) Overall: I like this piece in general.

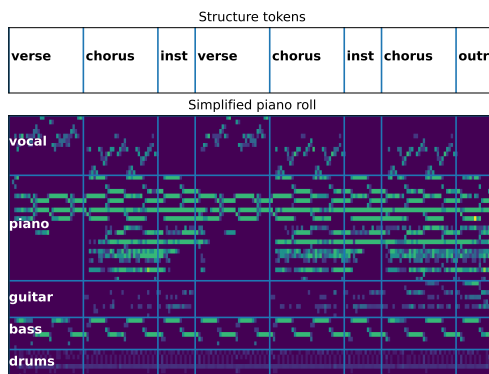


Figure 2: Constrained generation output with user-defined structure using IHD 100% model. The simplified piano roll gives beat-averaged values and excludes empty lines.

at scale. The result using LMD was better than MSD, despite having fewer songs; this could be due to LSD having more notes than MSD (see Tab 1), or due to it containing full songs instead of only excerpts. We only compare FIGARO and Ours (LMD) with a statistical test, since these were trained on the same dataset. Mann-Whitney U tests found significant differences in Richness ($p = .005$), Structure ($p = .0005$), and Overall ($p = .027$) ratings, but not in Coherence ($p = .85$) or Arrangement ($p = .122$).

5. CONCLUSIONS AND FUTURE WORK

We trained a language model for symbolic music generation leveraging audio data and pre-trained MIR models. We proposed the SymPAC framework, which includes prompt bars in encoding and Constrained Generation via FSM during inference time. We showed how combining these two components enables a user to control the generation process, and we evaluated the results through quantitative and qualitative analysis.

Future work could improve at least two aspects of this system: (1) We quantified position and duration to 1/16 per bar, which does not support 3/4 or 6/8 time signatures well. Also, the chord detection model we used only supports 12 major and minor chords, limiting the user input options. We can expand the encoding to support finer-grained quantization and more advanced chords. (2) Our token sequence length is long: 8000 on average for samples in IHD. We could use tokenization methods such as Byte Pair Encoding [40] or use compound word tokens [41] to compress sequences and improve training efficiency.

6. REFERENCES

- [1] K. Choi, C. Hawthorne, I. Simon, M. Dinculescu, and J. Engel, “Encoding musical style with transformer autoencoders,” in *International Conference on Machine Learning*. PMLR, 2020, pp. 1899–1908.
- [2] M. Zeng, X. Tan, R. Wang, Z. Ju, T. Qin, and T.-Y. Liu, “MusicBERT: Symbolic music understanding with large-scale pre-training,” in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 2021, pp. 791–800.
- [3] A. Agostinelli, T. I. Denk, Z. Borsos, J. Engel, M. Verzett, A. Caillon, Q. Huang, A. Jansen, A. Roberts, M. Tagliasacchi *et al.*, “MusicLM: Generating music from text,” *arXiv preprint arXiv:2301.11325*, 2023.
- [4] Q. Huang, D. S. Park, T. Wang, T. I. Denk, A. Ly, N. Chen, Z. Zhang, Z. Zhang, J. Yu, C. Frank *et al.*, “Noise2Music: Text-conditioned music generation with diffusion models,” *arXiv preprint arXiv:2302.03917*, 2023.
- [5] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Défossez, “Simple and controllable music generation,” *Advances in Neural Information Processing Systems*, vol. 36, 2024.
- [6] A. Roberts, J. Engel, C. Raffel, C. Hawthorne, and D. Eck, “A hierarchical latent vector model for learning long-term structure in music,” in *International Conference on Machine Learning*. PMLR, 2018, pp. 4364–4373.
- [7] S.-L. Wu and Y.-H. Yang, “MuseMorphose: Full-song and fine-grained piano music style transfer with one transformer VAE,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 1953–1967, 2023.
- [8] C. Payne, “MuseNet,” OpenAI, 25 Apr. 2019. [Online]. Available: openai.com/blog/musenet
- [9] J. Ens and P. Pasquier, “MMM: Exploring conditional multi-track music generation with the transformer,” *arXiv preprint arXiv:2008.06048*, 2020.
- [10] D. von Rütte, L. Biggio, Y. Kilcher, and T. Hofmann, “FIGARO: Generating symbolic music with fine-grained artistic control,” *arXiv preprint arXiv:2201.10936*, 2022.
- [11] H.-W. Dong, K. Chen, S. Dubnov, J. McAuley, and T. Berg-Kirkpatrick, “Multitrack music transformer,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.
- [12] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever *et al.*, “Language models are unsupervised multitask learners,” *OpenAI blog*, vol. 1, no. 8, 2019.
- [13] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, “Language models are few-shot learners,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 1877–1901, 2020.
- [14] Y.-N. Hung, J.-C. Wang, X. Song, W.-T. Lu, and M. Won, “Modeling beats and downbeats with a time-frequency transformer,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 401–405.
- [15] W.-T. Lu, J.-C. Wang, M. Won, K. Choi, and X. Song, “SpecTNT: A time-frequency transformer for music audio,” in *Proc. ISMIR*, 2021.
- [16] J.-C. Wang, Y.-N. Hung, and J. B. L. Smith, “To catch a chorus, verse, intro, or anything else: Analyzing a song with structural functions,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 416–420.
- [17] J.-C. Wang, J. B. L. Smith, and Y.-N. Hung, “MuSFA: Improving music structural function analysis with partially labeled data,” *arXiv preprint arXiv:2211.15787*, 2022.
- [18] W.-T. Lu, J.-C. Wang, and Y.-N. Hung, “Multi-track music transcription with a time-frequency perceiver,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.
- [19] M. Won, K. Choi, and X. Serra, “Semi-supervised music tagging transformer,” in *Proc. ISMIR*, 2021, pp. 768–776.
- [20] Y.-S. Huang and Y.-H. Yang, “Pop music transformer: Beat-based modeling and generation of expressive pop piano compositions,” in *Proceedings of the ACM International Conference on Multimedia*, 2020, pp. 1180–1188.
- [21] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. C. Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. Kloumann, A. Korenev, P. S. Koura, M.-A. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E. M. Smith, R. Subramanian, X. E. Tan, B. Tang, R. Taylor, A. Williams, J. X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, S. Edunov, and T. Scialom, “Llama 2: Open foundation and fine-tuned chat models,” 2023.
- [22] C. Hawthorne, A. Stasyuk, A. Roberts, I. Simon, C.-Z. A. Huang, S. Dieleman, E. Elsen, J. Engel, and

- D. Eck, “Enabling factorized piano music modeling and generation with the maestro dataset,” in *International Conference on Learning Representations*, 2018.
- [23] Q. Kong, B. Li, J. Chen, and Y. Wang, “GiantMIDI-Piano: A large-scale MIDI dataset for classical piano music,” *Transactions of the International Society for Music Information Retrieval*, vol. 5, no. 1, pp. 87–98, 2022.
- [24] C. Raffel, *Learning-based methods for comparing sequences, with applications to audio-to-midi alignment and matching*. Columbia University, 2016.
- [25] M. Defferrard, K. Benzi, P. Vandergheynst, and X. Bresson, “FMA: A dataset for music analysis,” in *Proc. ISMIR*, 2017. [Online]. Available: <https://arxiv.org/abs/1612.01840>
- [26] T. Bertin-Mahieux, D. P. Ellis, B. Whitman, and P. Lamere, “The million song dataset,” in *Proc. ISMIR*, 2011.
- [27] L. Lanzendörfer, F. Grötschla, E. Funke, and R. Wattenhofer, “DISCO-10M: A large-scale music dataset,” *Advances in Neural Information Processing Systems*, vol. 36, 2024.
- [28] C.-Z. A. Huang, A. Vaswani, J. Uszkoreit, N. Shazeer, C. Hawthorne, A. M. Dai, M. D. Hoffman, and D. Eck, “Music transformer: Generating music with long-term structure,” *arXiv preprint arXiv:1809.04281*, 2018.
- [29] C. Donahue, H. H. Mao, Y. E. Li, G. W. Cottrell, and J. McAuley, “LakhNES: Improving multi-instrumental music generation with cross-domain pre-training,” *arXiv preprint arXiv:1907.04868*, 2019.
- [30] Z. Sheng, K. Song, X. Tan, Y. Ren, W. Ye, S. Zhang, and T. Qin, “SongMASS: Automatic song writing with pre-training and alignment constraint,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 15, 2021, pp. 13 798–13 805.
- [31] A. S. Hussain, S. Liu, C. Sun, and Y. Shan, “M²UGen: Multi-modal music understanding and generation with the power of large language models,” *arXiv preprint arXiv:2311.11255*, 2023.
- [32] P. Lu, X. Xu, C. Kang, B. Yu, C. Xing, X. Tan, and J. Bian, “MuseCoco: Generating symbolic music from text,” *arXiv preprint arXiv:2306.00110*, 2023.
- [33] R. Yuan, H. Lin, Y. Wang, Z. Tian, S. Wu, T. Shen, G. Zhang, Y. Wu, C. Liu, Z. Zhou *et al.*, “ChatMusician: Understanding and generating music intrinsically with llm,” *arXiv preprint arXiv:2402.16153*, 2024.
- [34] L.-C. Yang and A. Lerch, “On the evaluation of generative models in music,” *Neural Computing and Applications*, vol. 32, no. 9, pp. 4773–4784, 2020.
- [35] S. Ji, X. Yang, and J. Luo, “A survey on deep learning for symbolic music generation: Representations, algorithms, evaluations, and challenges,” *ACM Computing Surveys*, vol. 56, no. 1, pp. 1–39, 2023.
- [36] O. Nieto, G. J. Mysore, C.-i. Wang, J. B. L. Smith, J. Schlüter, T. Grill, and B. McFee, “Audio-based music structure analysis: Current trends, open challenges, and applications,” *Transactions of the International Society for Music Information Retrieval*, vol. 3, no. 1, pp. 246–263, 2020.
- [37] J. Foote, “Automatic audio segmentation using a measure of audio novelty,” in *Proceedings of the IEEE International Conference on Multimedia and Expo*, 2000, pp. 452–455.
- [38] O. Nieto and J. P. Bello, “Music segment similarity using 2D-Fourier magnitude coefficients,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 664–668.
- [39] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, and D. P. Ellis, “mir_eval: A transparent implementation of common MIR metrics,” in *Proc. ISMIR*. Curitiba, Brazil: Citeseer, 2014, pp. 367–372.
- [40] N. Fradet, N. Gutowski, F. Chhel, and J.-P. Briot, “Byte pair encoding for symbolic music,” *arXiv preprint arXiv:2301.11975*, 2023.
- [41] W.-Y. Hsiao, J.-Y. Liu, Y.-C. Yeh, and Y.-H. Yang, “Compound word transformer: Learning to compose full-song music over dynamic directed hypergraphs,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 1, 2021, pp. 178–186.

UNSUPERVISED COMPOSABLE REPRESENTATIONS FOR AUDIO

Giovanni Bindi, Philippe Esling

Institut de Recherche et Coordination Acoustique-Musique (IRCAM)

CNRS UMR 9912, Sorbonne Université

{bindi, esling}@ircam.fr

ABSTRACT

Current generative models are able to generate high-quality artefacts but have been shown to struggle with compositional reasoning, which can be defined as the ability to generate complex structures from simpler elements. In this paper, we focus on the problem of compositional representation learning for music data, specifically targeting the fully-unsupervised setting. We propose a simple and extensible framework that leverages an explicit compositional inductive bias, defined by a flexible auto-encoding objective that can leverage any of the current state-of-art generative models. We demonstrate that our framework, used with diffusion models, naturally addresses the task of unsupervised audio source separation, showing that our model is able to perform high-quality separation. Our findings reveal that our proposal achieves comparable or superior performance with respect to other blind source separation methods and, furthermore, it even surpasses current state-of-art supervised baselines on signal-to-interference ratio metrics. Additionally, by learning an a-posteriori masking diffusion model in the space of composable representations, we achieve a system capable of seamlessly performing unsupervised source separation, unconditional generation, and variation generation. Finally, as our proposal works in the latent space of pre-trained neural audio codecs, it also provides a lower computational cost with respect to other neural baselines.

1. INTRODUCTION

Generative models recently became one of the most important topic in machine learning research. Their goal is to learn the underlying probability distribution of a given dataset in order to accomplish a variety of downstream tasks, such as sampling or density estimation. These models, relying on deep neural networks as their core architecture, have demonstrated unprecedented capabilities in capturing intricate patterns and generating complex and realistic data [1]. Although these systems are able to generate impressive results that go beyond the replication of training data, some doubts have recently been raised about their ac-

tual reasoning and extrapolation abilities [2, 3]. Notably, a critical question remains on their capacity to perform *compositional reasoning*. The principle of compositionality states that the meaning of a complex expression is dependent on the meanings of its individual components and the rules employed to combine them [4, 5]. This concept also plays a significant role in machine learning [6], with a particular emphasis in the fields of NLP and vision. Indeed, compositionality holds a strong significance in the *interpretability* of machine learning algorithms [7], ultimately providing a better understanding of the behaviour of such complex systems. In line with recent studies on compositional inductive biases [8, 9], taking a compositional approach would allow to build better representation learning and more effective generative models, but research on compositional learning for audio is still lacking.

In this work, we specifically focus on the problem of compositional representation learning for audio and propose a generic and simple framework that explicitly targets the learning of composable representations in a fully unsupervised way. Our idea is to learn a set of low-dimensional latent variables that encode semantic information which are then used by a generative model to reconstruct the input. While we build our approach upon recent diffusion models, we highlight that our framework can be implemented with any state-of-the-art generative system. Therefore, our proposal effectively combines diffusion models and auto-encoders and represents, to the best of our knowledge, one of the first contributions that explicitly target the learning of unsupervised compositional semantic representations for audio. Although being intrinsically modality-agnostic, we show that our system can be used to perform *unsupervised source separation* and we validate this claim by performing experiments on standard benchmarks, comparing against both unsupervised and supervised baselines. We show that our proposal outperforms all unsupervised methods, and even supervised methods on some metrics. Moreover, as we are able to effectively perform latent source separation, we complement our decomposition system with a prior model that performs *unconditional generation* and *variation generation* [10]. Hence, our method is able to take an audio mixture as input, and generate several high-quality variations for one of the instrumental part only, effectively allowing to control regeneration of a source audio material in multi-instrument setups. Furthermore, we train a masking diffusion model in the latent space of composable representation and show



that our framework is able to handle both decomposition and generation in an effective way without any supervision. We provide audio examples, additional experiments and source code on a supporting webpage¹

2. BACKGROUND

In this section, we review the fundamental components of our methodology. Hence, we briefly introduce the principles underlying diffusion models and a recent variation rooted in autoencoders, referred to as Diffusion Autoencoder [11], which serves as the basis for our formulation.

Notation. Throughout this paper, we suppose a dataset $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^n$ of *i.i.d.* data points $\mathbf{x}_i \in \mathbb{R}^d$ coming from an unknown distribution $p^*(\mathbf{x})$. We denote $\theta \in \Theta \subseteq \mathbb{R}^p$, $\phi \in \Phi \subseteq \mathbb{R}^q$ and $\psi \in \Psi \subseteq \mathbb{R}^r$ as the set of parameters learned through back-propagation [12].

2.1 Diffusion models

Diffusion models (DMs) are a recent class of generative models that can synthesize high-quality samples by learning to reverse a stochastic process that gradually adds noise to the data. DMs have been successfully applied across diverse domains, including computer vision [13], natural language processing [14], audio [15] and video generation [16]. These applications span tasks such as unconditional and conditional generation, editing, super-resolution and inpainting, often yielding state of the art results.

This model family has been introduced by [17] and has its roots in statistical physics, but there now exist many derivations with different formalisms that generalise the original formulation. At their core, DMs are composed of a *forward* and *reverse* Markov chain that respectively adds and removes Gaussian noise from data. Recently, [18] established a connection between DM and denoising score matching [19, 20], introducing simplifications to the original training objective and demonstrating strong experimental results. Intuitively, the authors propose to learn a function ϵ_θ that takes a noise-corrupted version of the input and predicts the noise ϵ used to corrupt the data. Specifically, the *forward* process gradually adds Gaussian noise to the data $\mathbf{x} \rightarrow \mathbf{x}_t$ according to an increasing noise variance schedule β_1, \dots, β_T , following the distribution

$$q(\mathbf{x}_t | \mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1 - \beta_t} \mathbf{x}_{t-1}, \beta_t \mathbf{I}), \quad (1)$$

with $T \in \mathbb{N}$ and $t \in \{1, \dots, T\}$. Following the notation $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^t \alpha_s$, diffusion models approximate the *reverse* process by learning a function $\epsilon_\theta : \mathbb{R}^d \times \mathbb{R} \rightarrow \mathbb{R}^d$ that predicts $\epsilon \sim \mathcal{N}(\epsilon, \mathbf{0}, \mathbf{I})$ by

$$\min_{\theta \in \Theta} \mathbb{E}_{t, \mathbf{x}_0, \epsilon} [\|\epsilon_\theta(\sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon, t) - \epsilon\|], \quad (2)$$

with ϵ_θ usually implemented as a U-Net [21] and the step $t \sim \mathcal{U}[0, T]$.

Deterministic diffusion. More recently, [22] introduced Denoising Diffusion Implicit Models (DDIM), extending the diffusion formulation with non-Markovian

modifications, thus enabling deterministic diffusion models and substantially increasing their sampling speed. They also established an equivalence between their objective function and the one from [18], highlighting the generality of their formulation. Finally, [23] further generalized this approach and proposed Iterative α -(de)Blending (IADB), simplifying the theory of DDIM while removing the constraint for the target distribution to be Gaussian. In fact, given a base distribution² $p_n(\mathbf{x}_0)$, we corrupt the input data by linear interpolation $\mathbf{x}_\alpha = (1 - \alpha)\mathbf{x}_0 + \alpha\mathbf{x}$ with $\mathbf{x}_0 \sim p_n(\mathbf{x}_0)$ and learn a U-Net ϵ_θ by optimizing, e.g.,

$$\min_{\theta \in \Theta} \mathbb{E}_{\alpha, \mathbf{x}, \mathbf{x}_0} [\|\epsilon_\theta(\mathbf{x}_\alpha, \alpha) - \mathbf{x}\|_2^2], \quad (3)$$

with $\alpha \sim \mathcal{U}[0, 1]$. This is known as the *c* variant of IADB, which is the closest formulation to DDIM. In our implementation, we instead use the *d* variant of IADB, which has a slightly different formulation that we do not report for brevity. We experimented with both variants and did not find significant discrepancies in performances.

Diffusion Autoencoders. All the methods described in the preceding paragraph specifically target unconditional generation. However, in this work we are interested in conditional generation and, more specifically, in a conditional encoder-decoder architecture. For this reason, we build upon the recent work by [11] named Diffusion Autoencoder (DiffAE). The central concept in this approach involves employing a learnable encoder to discover high-level semantic information, while using a DM as the decoder to model the remaining stochastic variations. Therefore, the authors equip a DDIM model ϵ_ϕ with a semantic encoder $E_\theta : \mathbb{R}^d \rightarrow \mathbb{R}^s$ with $s \ll d$ that is responsible for compressing the high-level *semantic* information³ into a latent variable $\mathbf{z} \in \mathbb{R}^s$ as $\mathbf{z} = E_\theta(\mathbf{x})$. The DDIM model is, therefore, conditioned on such semantic representation and trained to reconstruct the data via

$$\min_{\theta \in \Theta, \phi \in \Phi} \mathbb{E}_{t, \mathbf{x}_0, \epsilon} [\|\epsilon_\phi(\sqrt{\alpha} \mathbf{x}_0 + \sqrt{1 - \alpha} \epsilon, \mathbf{z}, t) - \epsilon\|] \quad (4)$$

with $\alpha = \prod_{s=1}^t (1 - \beta_s)$ and β_i being the variance at the *i*-th step. Since the DiffAE represents the state of the art for encoder-decoder models based on diffusion, we build our compositional diffusion framework upon this formulation, which we describe in the following section.

3. PROPOSED APPROACH

In compositional representation learning, we hypothesize that the information can be deconstructed into specific, identifiable parts that collectively makes up the whole input. In this work, we posit these parts to be distinct instruments in music but we highlight that this choice is uniquely dependent on the target application. Due to the lack of a widely-accepted description of compositional representations, we formulate a simple yet comprehensive definition that can subsequently be specialized to address particular

² For simplicity we assume $p_n(\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_0; \mathbf{0}, \mathbf{I})$.

³ In the domain of vision this could be the identity of a person or the type of objects represented in an image.

¹ https://github.com/ismir-24-sub/unsupervised_compositional_representations

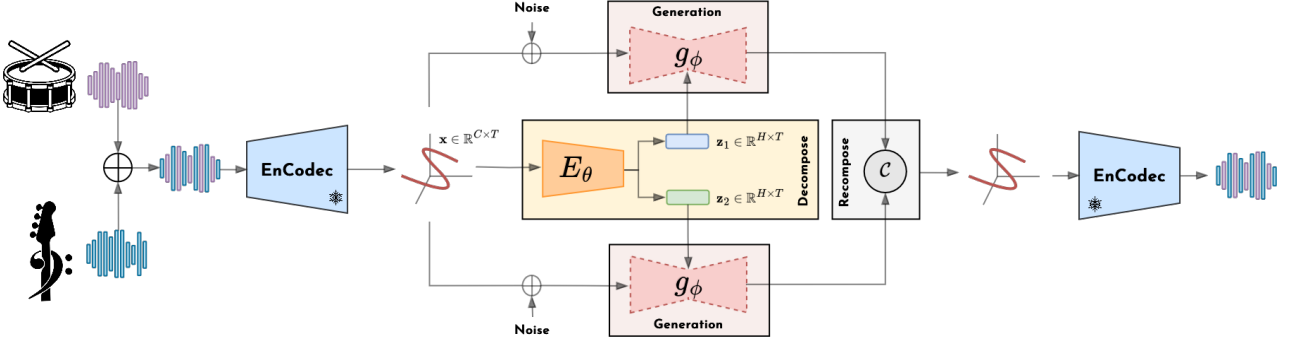


Figure 1. The overall architecture of our decomposition model. We first mix the sources, map the data \mathbf{x} to the latent space through a frozen, pre-trained EnCodec model, and then decompose it into a set of latent variables (two shown here). These variables then condition a parameter-sharing diffusion model whose generation are then recomposed by an operator \mathcal{C} .

cases [24, 25]. Specifically, we start from the assumption that observations $\mathbf{x} \in \mathbb{R}^d$ are realizations of an underlying latent variable model and that each concept is described by a corresponding latent $\mathbf{z}_i \in \mathcal{Z}_i$, where $i \in \{1, \dots, N\}$ with N being the total number of possible entities that compose our data. Then, we define a compositional representation of \mathbf{x} as

$$\mathbf{x} = \mathcal{C}(\hat{\mathbf{z}}_1, \dots, \hat{\mathbf{z}}_N) = \mathcal{C}(f_1(\mathbf{z}_1), \dots, f_N(\mathbf{z}_N)), \quad (5)$$

where $\mathcal{C} : \hat{\mathcal{Z}}_1 \times \hat{\mathcal{Z}}_2 \times \dots \times \hat{\mathcal{Z}}_N \rightarrow \mathbb{R}^d$ is a *composition operator* and each $f_i : \mathcal{Z}_i \rightarrow \hat{\mathcal{Z}}_i$ is a *processing function* that maps each latent variable to another *intermediate* space. By being intentionally broad, this definition does not impose any strong specific constraints a priori, such as the requirement for each subspace to be identical or the algebraic structure of the latent space itself. Hence, to implement this model, we rather need to consider careful intentional design choices and inductive biases. In this work, we constrain the intermediate space to be the data space itself, i.e. $\hat{\mathcal{Z}}_i = \mathbb{R}^d$ for all $i = 1, \dots, N$ and we focus on the learning of the latent variables and the processing functions. Finally, we set the composition operator to be a pre-defined function such as *mean* or *max* and leave its learning to further investigations.

3.1 Decomposition

In this section, we detail our proposed model, as depicted in Figure 1. Globally, we follow an encoder-decoder paradigm, where we encode the data $\mathbf{x} \in \mathbb{R}^d$ into a set of latent representations $\mathcal{Z} = \{\mathbf{z}_1, \dots, \mathbf{z}_N\}$, where $\mathbf{z}_i \in \mathcal{Z} \subseteq \mathbb{R}^h$ for each $i = 1, \dots, N$. This is done through an encoder network $E_\theta : \mathbb{R}^d \rightarrow \mathcal{Z} \times \dots \times \mathcal{Z}$ that maps the input \mathbf{x} to the set of variables \mathcal{Z} , i.e. $[\mathbf{z}_1, \dots, \mathbf{z}_N] = E_\theta(\mathbf{x})$. Each latent variable is then decoded separately through a parameter-sharing diffusion model, which implements the *processing function* $f : \mathcal{Z} \rightarrow \mathbb{R}^d$ in Equation 5, mapping the latents to the data space. Finally, we reconstruct the input data \mathbf{x} through the application of a *composition operator* \mathcal{C} and train the system end-to-end through a vanilla iterative α -(de)Blending (IADB) loss. Specifically, we learn a U-Net network $g_\phi : \mathbb{R}^d \times \mathbb{R} \times \mathbb{R}^h \rightarrow \mathbb{R}^d$ and a

semantic encoder E_θ via the following objective

$$\min_{\theta \in \Theta, \phi \in \Phi} \mathbb{E}_{\alpha, \mathbf{x}, \mathbf{x}_0} [\|\hat{g}_\phi(\mathbf{x}_\alpha, \alpha) - \mathbf{x}\|_2^2], \quad (6)$$

with $\alpha \sim \mathcal{U}[0, 1]$, $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{x}_0; \mathbf{0}, \mathbf{I})$ and

$$\hat{g}_\phi(\mathbf{x}_\alpha, \alpha) = \mathcal{C}(g_\phi(\mathbf{x}_\alpha, \alpha, \mathbf{z}_1), \dots, g_\phi(\mathbf{x}_\alpha, \alpha, \mathbf{z}_N)), \quad (7)$$

with $\mathbf{x}_\alpha = (1 - \alpha)\mathbf{x}_0 + \alpha\mathbf{x}$ and $[\mathbf{z}_1, \dots, \mathbf{z}_N] = E_\theta(\mathbf{x})$. We chose the IADB paradigm due to its simplicity in implementation and intuitive nature, requiring minimal hyperparameter tuning.

At inference time, we reconstruct the input by progressively denoising an initial random sample coming from the prior distribution, conditioned on the components obtained through the semantic encoder.

A note on complexity. We found that using a single diffusion model proves effective instead of training N separate models for N latent variables. Consequently, we opt for training a parameter-sharing neural network g_ϕ . Nonetheless, the computational complexity of our framework is therefore N times that of a single DiffAE.

3.2 Recomposition

One of our primary objectives is to endow models with *compositional generation*, a concept we define as the ability to generate novel data examples by coherently re-composing distinct parts extracted from separate origins. This definition aligns with numerous related studies that posit compositional generalization as an essential requirement to bridge the gap between human reasoning and computational learning systems [26]. In this work, we allow for compositional generation by learning a prior model in the components' space. Specifically, once we have a well-trained decomposition model $D_{\theta, \phi} = (E_\theta, g_\phi)$ we learn a diffusion model in \mathcal{Z} in order to obtain a full generative system. We define $\mathbf{z} = [\mathbf{z}_1, \dots, \mathbf{z}_N] = E_\theta(\mathbf{x})$ and train a IADB model to recover \mathbf{z} from a masked view $\tilde{\mathbf{z}}$. At training time, with probability p_{mask} , we mask each latent variable \mathbf{z}_i with a mask $\mathbf{m}_i \in \{0, 1\}^{dim(\mathcal{Z})}$ and optimize the diffusion model ϵ_ψ by solving

$$\min_{\psi \in \Psi} \mathbb{E}_{\alpha, \mathbf{z}, \mathbf{z}_0, \mathbf{m}} [\|\mathbf{z} - \epsilon_\psi(\mathbf{z}_\alpha, \alpha, \mathbf{m})\|_2^2], \quad (8)$$

Algorithm 1 Training prior model

Input: dataset \mathcal{D} , U-Net ϵ_ψ , pre-trained semantic encoder E_θ , masking probability p_{mask} , learning rate γ .
while not converged **do**
 for \mathbf{x} in \mathcal{D} **do**
 $\mathbf{z} = [\mathbf{z}_1, \dots, \mathbf{z}_N] = E_\theta(\mathbf{x})$.
 Sample $\alpha \sim \mathcal{U}[0, 1]$ and $\mathbf{z}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.
 $\tilde{\mathbf{z}}_\alpha = (1 - \alpha)\mathbf{z}_0 + \alpha\mathbf{z}$
 Draw $\mathbf{m} \in \{0, 1\}^{dim(\mathcal{Z}) \times \dots \times dim(\mathcal{Z})}$
 $\mathbf{z}_\alpha = \tilde{\mathbf{z}}_\alpha \odot \mathbf{m} + (1 - \mathbf{m}) \odot \mathbf{z}$
 $\mathcal{L}(\psi, \mathbf{z}, \alpha, \mathbf{m}) = \|\mathbf{z} - \epsilon_\psi(\mathbf{z}_\alpha, \alpha, \mathbf{m})\|^2$
 Update $\psi \leftarrow \psi - \gamma \nabla_\psi \mathcal{L}(\psi, \mathbf{z}, \alpha, \mathbf{m})$
 end for
end while
Return: ϵ_ψ

where $\mathbf{z}_\alpha = \tilde{\mathbf{z}}_\alpha \odot \mathbf{m} + (1 - \mathbf{m}) \odot \mathbf{z}$ and $\tilde{\mathbf{z}}_\alpha = (1 - \alpha)\mathbf{z}_0 + \alpha\mathbf{z}$. Here, $\mathbf{z}_0 \sim \mathcal{N}(\mathbf{z}_0; \mathbf{0}, \mathbf{I})$ and $\tilde{\mathbf{z}}_\alpha$ denotes the α -blended source \mathbf{z} . At each training iteration we randomly mask $\tilde{\mathbf{z}}_\alpha$ via \mathbf{m} and train the diffusion model ϵ_ψ to recover the masked elements given the unmasked view \mathbf{z} . Our masking strategy allows for dropping each latent separately as well as all the latents simultaneously, effectively leading to a model that is able to perform both conditional and unconditional generation at the same time. In our application case, the conditional generation task reduces to the problem of generating variations. As our decomposition model proves to be effective in separating the stems of a given mixture, we obtain a system that is able to generate missing stems given the masked elements. Hence, this also addresses the accompaniment generation task. Algorithm 1 resumes the training process of the prior model.

4. EXPERIMENTS AND RESULTS

This section provides an overview of the experiments aimed at assessing the performance of our proposal in both decomposition (section 4.1) and recomposition (section 4.2) scenarios. Prior to diving into the specifics of each experiment, we provide a brief overview of the shared elements across our experiments, including data, evaluation metrics, and neural network architectures.

Data. We rely on the Slakh2100 dataset [27], a widely recognized benchmark in source separation, comprising 2100 tracks automatically mixed with separate stems. We selected this dataset because of its large-scale nature and the availability of ground truth separated tracks. Following recent approaches in generative models [28, 29], we rely on a pre-trained neural codec to map the audio data to an intermediate latent space, where we apply our approach. Specifically, we employ the EnCodec model [30], a Vector Quantized-VAE (VQ-VAE) model [31] that incorporates Residual Vector Quantization [32] to achieve state-of-the-art performances in neural audio encoding. We take 24 kHz mixtures from the Slakh2100 dataset, which we then feed to the pre-trained EnCodec model to extract the continuous representation obtained by decoding the discrete codes. EnCodec maps raw audio to latent trajectories with

MS-STFT	FAD (LC-A)	FAD (LC-M)
4.7	0.05	0.04

Table 1. EnCodec reconstruction quality, measured in terms of MS-STFT and FAD and computed following the procedure described in section 4.

a sampling rate of 75 Hz. Specifically, we take audio crops of approximately 7s (6.82s), which are mapped via EnCodec to a latent code $\mathbf{x} \in \mathbb{R}^{128 \times 512}$.

Evaluation metrics. Throughout this section, we report quantitative *reconstruction* metrics in terms of both Mean Squared Error (MSE) and Multi-Scale Short-Time Fourier Transform (MS-STFT) [33, 34] for latent and audio data, respectively. We perform the MS-STFT evaluation using five STFT with window sizes {2048, 1024, 512, 256, 128} following the implementation of [34]. In order to evaluate the quality of the generated samples and the adherence to the training distribution, we also compute Fréchet Audio Distance (FAD) [35, 36] scores. Specifically, we obtain the FAD scores via the `fadt` library [36], employing both the LAION-CLAP-Audio (LC-A) and LAION-CLAP-Music (LC-M) models [37], as it was shown in [36] that these embedding models correlate well with perceptual tests measuring subjective quality of pop music. In assessing FAD scores, we utilize the complete test set of Slakh2100, while for MSE and MS-STFT values, we randomly select 512 samples of 7s (~ 1 hour) from the same test set and report their mean and standard deviation. Finally, in order to provide the reader a reference value, we report in Table 1 the reconstruction metrics for the pre-trained EnCodec.

When assessing the effectiveness of *source separation* models, we adhere to common practice by relying on the `museval` Python library [38] to compute standard separation metrics: Source-to-Interference Ratio (SIR), Source-to-Artifact Ratio (SAR), and Source-to-Distortion Ratio (SDR) [39]. These metrics are widely accepted for evaluating source separation models, where SDR reflects sound quality, SIR indicates the presence of other sources, and SAR evaluates the presence of artifacts in a source. Specifically, following [39] we compute their scale-invariant (SI) versions and, hence, provide our results in terms of SI-SDR, SI-SIR and SI-SAR. The values shown are expressed in terms of mean μ and standard deviation σ computed on 512 samples of ~ 7s from the Slakh2100 test set.

Architectures. We use a standard U-Net [21] with 1D convolution and an encoder-decoder architecture with skip connections. Each processing unit is a ResNet block [40] with group normalization [41]. Following [42], we feed the noise level information through Positional Encoding [43], conditioning each layer with the AdaGN mechanism. We also add multi-head self-attention [43] in the bottleneck layers of the U-Net. The semantic encoder mirrors the U-Net encoder block without the attention mechanism and maps the data $\mathbf{x} \in \mathbb{R}^{128 \times 512}$ to a set of variables $\mathbf{z} = [\mathbf{z}_1 \dots \mathbf{z}_i \dots \mathbf{z}_N]$ whose dimensionality is $\mathbf{z}_i \in \mathbb{R}^{1 \times 512}$.

Finally, these univariate latent variables condition the U-Net via a simple concatenation, which proved to be a sufficiently effective conditioning mechanism for the model to converge. We use the same U-Net architecture for both the decomposition and recomposition diffusion models.

4.1 Decomposition

In order to show the effectiveness of our decomposition method described in section 3.1, we perform multiple experiments on Slakh2100. Throughout this section, we fix the number of training epochs to 250 and use the AdamW optimizer [44] with a fixed learning rate of 10^{-4} as our optimization strategy. The U-Net and semantic encoder have 13 and 8 million trainable parameters, respectively. Finally, we use 100 sampling steps at inference time.

First, we show in Table 2 that our model can be used to perform unsupervised latent source separation and compare it against several non-neural baselines [45–49], as well as a recent study that explicitly targets neural latent blind source separation [50]. We also report the results obtained by Demucs [51], which is the current top performing fully-supervised state-of-the-art method in audio source separation. As the only non-neural baseline, LASS, has been trained and evaluated on the Drums + Bass subset, we perform our analysis on this split and subsequently perform an ablation study over the other sources.

Model	SI-SDR (\uparrow)	SI-SIR (\uparrow)	SI-SAR (\uparrow)
rPCA [45]	-2.8 (4.8)	5.2 (7.3)	<u>5.6</u> (4.6)
REPET [48]	-0.5 (4.8)	6.8 (7.0)	3.0 (5.2)
FT2D [49]	-0.2 (4.7)	5.1 (7.0)	3.1 (4.7)
NMF [46]	1.4 (5.0)	8.9 (7.6)	2.9 (4.5)
HPSS [47]	2.3 (4.8)	9.9 (7.5)	5.1 (4.6)
LASS [50]	-3.3 (10.8)	17.7 (11.6)	-1.6 (11.2)
Ours	<u>5.5</u> (4.6)	41.7 (9.3)	<u>5.6</u> (4.6)
Demucs [51]	11.9 (5.0)	<u>37.6</u> (8.7)	12.0 (5.0)

Table 2. Blind source separation results for the *Drums + Bass* subset. Our model is trained with the *mean* composition operator. The results are expressed in dB as the mean (standard deviation) across 512 elements randomly sampled from the test set of Slakh2100.

As we can see, our model outperforms the other baselines in terms of SI-SDR and SI-SIR and performs on par with respect to SI-SAR. Interestingly, our model outperforms the Demucs supervised baseline in terms of SI-SIR, which is usually interpreted as the amount of other sources that can be heard in a source estimate. In order to test LASS performances, we used their open source checkpoint which is trained on the Slakh2100 dataset, and followed their evaluation strategy. Unfortunately, we were not able to reproduce their results in terms of SDR but we found that their model performs well in terms of SI-SIR, which they did not measure in the original paper. Moreover, as LASS comprises training one transformer model per source, we found their inference phase to be more com-

Operator	MSE (\downarrow) $\times 10^4$	MS-STFT (\downarrow)
<i>Sum</i>	1.87820 (0.13418)	3.6 (0.1)
<i>Mean</i>	1.87020 (0.13183)	3.6 (0.1)
<i>Min</i>	2.54182 (0.17714)	4.5 (0.1)
<i>Max</i>	2.43302 (0.17510)	4.3 (0.1)

Table 3. Reconstruction quality in latent space (MSE) and audio (MS-STFT) of our decomposition-recomposition model for different recomposition operators for the *Drums + Bass* subset.

putationally demanding than ours. Finally, among non-neural baselines, we see that the HPSS model outperforms the others. This seems reasonable as HPSS is specifically built for separating percussive and harmonic sources and hence naturally fits this evaluation context.

Moreover, in order to show the robustness of our approach against different sources and number of latent variables, we train multiple models on different subset of the Slakh2100 dataset, namely *Drums + Bass*, *Piano + Bass* and *Drums + Bass + Piano*. The interested reader can refer to our supplementary material and listen to the separation results.

Subsequently, we show that our objective in Equation 6 is robust across different composition operators. We show that, for simple functions such as *sum*, *min*, *max* and *mean* our model is able to effectively converge and provide accurate reconstructions. Again, we provide this analysis by training our model on the *Drums + Bass* subset of Slakh2100, fixing the number of components to 2. We report quantitative results in terms of two reconstruction metrics, the Mean Squared Error (MSE) and Multi-Scale STFT distance (MS-STFT) in Table 3. As we can see, *sum* and *mean* operators provided the best results, while *min* and *max* proved to be less effective. Nonetheless, the audio reconstruction quality measured in terms of MS-STFT provided reconstruction scores that are lower or comparable with respect to those obtained by evaluating EnCodec performances.

4.2 Recomposition

As detailed in section 3.2, once we are able to decompose our data into a set of composable representations we can then learn a prior model for generation from this new space. Since our decomposition model is able to compress meaningful information through the semantic encoder, we can learn a second latent diffusion model on this compressed representation to obtain a full generative model able to both decompose and generate data.

Here, we validate our claims by training a masked diffusion model for the *Drums + Bass* split of the Slakh2100 dataset. In Table 4, we show that our model can indeed produce good-quality unconditional generations by comparing it against a fully unconditional model. We measure the generation quality in terms of FAD scores computed against both the original as well as the encoded test data. Here, by original data we mean the audio coming

	Original		Encoded	
	FAD _(LC-A) (↓)	FAD _(LC-M) (↓)	FAD _(LC-A) (↓)	FAD _(LC-M) (↓)
Unconditional	0.09	0.09	0.06	0.06
$p_{mask} = 0.8$	0.12	0.11	0.08	0.07
Bass	0.03	0.03	0.01	0.01
Drums	0.09	0.08	0.05	0.05

Table 4. Audio quality of unconditional generations by our generative model. We demonstrate that we can jointly learn an unconditional and conditional model by showing that the FAD scores of $p_{mask} = 0.8$ are comparable to those of an unconditional latent diffusion model.

	Type	MSE $\times 10^3$	MS-STFT
Real	Drums	2.3259 (0.1287)	13.6 (0.4)
	Bass	1.4393 (0.0874)	9.38 (0.2)
Rand	Drums	4.8170 (0.1136)	20.5 (0.6)
	Bass	4.8814 (0.1157)	21.7 (0.7)

Table 5. Diversity of variations generated by our prior model, measured via the MSE and MS-STFT distances against ground truth and random components.

from the test split of Slakh2100, while the encoded data represents the same elements reconstructed with our decomposition algorithm. As we train on the representations obtained through the semantic encoder, the natural benchmark for the unconditional generation is given by the reconstructions that we can obtain through our decomposition model, which represents the bottleneck in terms of quality. Nonetheless, we show that the FAD scores do not drop substantially when comparing against the original audio, showing that we can indeed achieve a good generation quality. In the same table, we report the partial generation FAD scores. Instead of generating both components unconditionally, we generate the Bass (Drums) given the Drums (Bass), and measure the FAD against the original and the encoded test data, as done for the unconditional case. Given the presence of a ground-truth element, the FAD scores are lower, which is to be expected. Specifically, we can see that the drums generation is a more complex task with respect to the bass generation, as the model needs to synthesize more elements such as the kick, snare and hi-hats, matching the timing of a given bassline.

Lastly, as we strive for high-quality generations, we also aim to enhance diversity within our generations. Table 5 shows the diversity scores for partial generations obtained with our model. We measure diversity in terms of MSE and MS-STFT scores computed, respectively, in the latent and audio space. We compare our partial generations against real and random components, in order to provide the lower and upper bound for generation diversity. Specifically, given the Drums (Bass) we generate the Bass (Drums) and we compute both MSE and MS-STFT scores against the ground truth (Real) and random elements (Rand) coming from the test set of Slakh2100. From the values reported in Table 5, we can deduce that our model produces meaningful variations. We invite the interested readers to listen

to our results on our support website.

5. DISCUSSION AND FURTHER WORKS

While our model proves to be effective for compositional representation learning, it still has shortcomings. Here, we briefly list the weaknesses of our proposal and highlight potential avenues for future investigations.

Factors of convergence. In this paper, we used En-Codec which already provides some disentanglement and acts as a sort of initialization strategy for our method. We argue that this property, jointly with the low dimensionality of the latent space enforced by our encoder leads our decomposition model to converge efficiently, not requiring further inductive biases towards source separation.

Limitations. First, there is no theoretical guarantee that the learned latent variables are bound to encode meaningful information. Exploring more refined approaches, as proposed by [52], could be interesting in order to incorporate a more principled method for learning disentangled latent representations. Furthermore, we observed that the dimensionality of the latent space significantly influences the representation content. A larger dimensionality allows the model to encode all the information in each latent, hindering the learning of distinct factors. Conversely, a smaller dimensionality may lead to under-performance, preventing the model to correctly converge. It could be interesting to investigate strategies such as Information Bottleneck [53] to introduce a mechanism to explicitly trade off expressivity with compression. Finally, using more complex functions as well as learnable operators is an interesting research direction for studying the interpretability of learned representations.

6. CONCLUSIONS

In this work, we focus on the problem of learning unsupervised compositional representations for audio. We build upon recent state-of-the-art diffusion generative models to design an encoder-decoder framework with an explicit inductive bias towards compositionality. We validate our approach on audio data, showing that our method can be used to perform latent source separation. Despite the theoretical shortcomings, we believe that our proposal can serve as a useful framework for conducting research on the topics of unsupervised compositional representation learning.

7. REFERENCES

- [1] S. Bond-Taylor, A. Leach, Y. Long, and C. G. Willcocks, “Deep generative modelling: A comparative review of vaes, gans, normalizing flows, energy-based and autoregressive models,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 11, pp. 7327–7347, 2022.
- [2] A. Srivastava, A. Rastogi, A. Rao, A. A. M. Shoeb, A. Abid, A. Fisch, A. R. Brown, A. Santoro, A. Gupta, A. Garriga-Alonso, A. Kluska, A. Lewkowycz, A. Agarwal, A. Power, A. Ray, A. Warstadt, A. W. Kocurek, A. Safaya, A. Tazarv, A. Xiang, A. Parrish, A. Nie, and et al., “Beyond the imitation game: Quantifying and extrapolating the capabilities of language models,” *Transactions on Machine Learning Research*, 2023. [Online]. Available: <https://openreview.net/forum?id=uyTL5Bvosj>
- [3] M. Yuksekogonul, F. Bianchi, P. Kalluri, D. Jurafsky, and J. Zou, “When and why vision-language models behave like bags-of-words, and what to do about it?” in *International Conference on Learning Representations*, 2023. [Online]. Available: <https://openreview.net/forum?id=KRLUvxh8uaX>
- [4] P. Pagin and D. Westerståhl, “Compositionality i: Definitions and variants,” *Philosophy Compass*, vol. 5, no. 3, pp. 250–264, 2010. [Online]. Available: <https://compass.onlinelibrary.wiley.com/doi/abs/10.1111/j.1747-9991.2009.00228.x>
- [5] T. Janssen, “19 Compositionality: Its Historic Context,” in *The Oxford Handbook of Compositionality*. Oxford University Press, 02 2012. [Online]. Available: <https://doi.org/10.1093/oxfordhb/9780199541072.013.0001>
- [6] B. M. Lake, T. D. Ullman, J. B. Tenenbaum, and S. J. Gershman, “Building machines that learn and think like people,” *Behavioral and Brain Sciences*, vol. 40, p. e253, 2017.
- [7] J. Mu and J. Andreas, “Compositional explanations of neurons,” in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, ser. NIPS’20. Red Hook, NY, USA: Curran Associates Inc., 2020.
- [8] G. Hinton, “How to Represent Part-Whole Hierarchies in a Neural Network,” *Neural Computation*, vol. 35, no. 3, pp. 413–452, 02 2023. [Online]. Available: https://doi.org/10.1162/neco_a_01557
- [9] B. Lake and M. Baroni, “Human-like systematic generalization through a meta-learning neural network,” *Nature*, vol. 623, no. 7985, pp. 115–121, Nov. 2023, publisher Copyright: © 2023, The Author(s).
- [10] G. Mariani, I. Tallini, E. Postolache, M. Mancusi, L. Cosmo, and E. Rodolà, “Multi-source diffusion models for simultaneous music generation and separation,” in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: <https://openreview.net/forum?id=h922Qhkmx1>
- [11] K. Preechakul, N. Chatthee, S. Wizadwongsa, and S. Suwajanakorn, “Diffusion autoencoders: Toward a meaningful and decodable representation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 10 619–10 629.
- [12] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors.” *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.
- [13] T. Karras, M. Aittala, T. Aila, and S. Laine, “Elucidating the design space of diffusion-based generative models,” in *Proc. NeurIPS*, 2022.
- [14] Z. He, T. Sun, Q. Tang, K. Wang, X. Huang, and X. Qiu, “DiffusionBERT: Improving generative masked language models with diffusion models,” in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, A. Rogers, J. Boyd-Graber, and N. Okazaki, Eds. Toronto, Canada: Association for Computational Linguistics, Jul. 2023, pp. 4521–4534. [Online]. Available: <https://aclanthology.org/2023.acl-long.248>
- [15] H. Liu, Z. Chen, Y. Yuan, X. Mei, X. Liu, D. Mandic, W. Wang, and M. D. Plumbley, “AudioLDM: Text-to-audio generation with latent diffusion models,” *Proceedings of the International Conference on Machine Learning*, 2023.
- [16] J. Ho, T. Salimans, A. A. Gritsenko, W. Chan, M. Norouzi, and D. J. Fleet, “Video diffusion models,” in *ICLR Workshop on Deep Generative Models for Highly Structured Data*, 2022. [Online]. Available: <https://openreview.net/forum?id=BBelR2NdDZ5>
- [17] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, “Deep unsupervised learning using nonequilibrium thermodynamics,” in *International conference on machine learning*. PMLR, 2015, pp. 2256–2265.
- [18] J. Ho, A. Jain, and P. Abbeel, “Denoising diffusion probabilistic models,” *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.
- [19] Y. Song and S. Ermon, “Generative modeling by estimating gradients of the data distribution,” in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2019/file/3001ef257407d5a371a96dcd947c7d93-Paper.pdf

- [20] P. Vincent, “A connection between score matching and denoising autoencoders,” *Neural computation*, vol. 23, no. 7, pp. 1661–1674, 2011.
- [21] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer, 2015, pp. 234–241.
- [22] J. Song, C. Meng, and S. Ermon, “Denoising diffusion implicit models,” in *International Conference on Learning Representations*, 2021. [Online]. Available: <https://openreview.net/forum?id=St1giarCHLP>
- [23] E. Heitz, L. Belcour, and T. Chambon, “Iterative α -(de)blending: A minimalist deterministic diffusion model,” in *ACM SIGGRAPH 2023 Conference Proceedings*, ser. SIGGRAPH ’23. New York, NY, USA: Association for Computing Machinery, 2023. [Online]. Available: <https://doi.org/10.1145/3588432.3591540>
- [24] J. Andreas, “Measuring compositionality in representation learning,” in *International Conference on Learning Representations*, 2019. [Online]. Available: <https://openreview.net/forum?id=HJz05o0qK7>
- [25] T. Wiedemer, P. Mayilvahanan, M. Bethge, and W. Brendel, “Compositional generalization from first principles,” in *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. [Online]. Available: <https://openreview.net/forum?id=LqOQ1uJmSx>
- [26] J. A. Fodor and Z. W. Pylyshyn, “Connectionism and cognitive architecture: A critical analysis,” *Cognition*, vol. 28, no. 1-2, pp. 3–71, 1988.
- [27] E. Manilow, G. Wichern, P. Seetharaman, and J. Le Roux, “Cutting music source separation some Slakh: A dataset to study the impact of training data quality and quantity,” in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2019.
- [28] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, “High-resolution image synthesis with latent diffusion models,” 2021.
- [29] F. Schneider, O. Kamal, Z. Jin, and B. Schölkopf, “Moûsai: Text-to-music generation with long-context latent diffusion,” 2023.
- [30] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, “High fidelity neural audio compression,” *arXiv preprint arXiv:2210.13438*, 2022.
- [31] A. van den Oord, O. Vinyals, and K. Kavukcuoglu, “Neural discrete representation learning,” in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS’17. Red Hook, NY, USA: Curran Associates Inc., 2017, p. 6309–6318.
- [32] N. Zeghidour, A. Luebs, A. Omran, J. Skoglund, and M. Tagliasacchi, “Soundstream: An end-to-end neural audio codec,” *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, vol. 30, p. 495–507, nov 2021. [Online]. Available: <https://doi.org/10.1109/TASLP.2021.3129994>
- [33] R. Yamamoto, E. Song, and J.-M. Kim, “Parallel wavegan: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram,” in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 6199–6203.
- [34] A. Caillon and P. Esling, “Rave: A variational autoencoder for fast and high-quality neural audio synthesis,” 2021.
- [35] K. Kilgour, M. Zuluaga, D. Roblek, and M. Sharifi, “Fréchet audio distance: A metric for evaluating music enhancement algorithms,” 2019.
- [36] A. Gui, H. Gamper, S. Braun, and D. Emmanouilidou, “Adapting frechet audio distance for generative music evaluation,” in *Proc. IEEE ICASSP 2024*, 2024. [Online]. Available: <https://arxiv.org/abs/2311.01616>
- [37] Y. Wu, K. Chen, T. Zhang, Y. Hui, T. Berg-Kirkpatrick, and S. Dubnov, “Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation,” in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, 2023.
- [38] F.-R. Stöter, A. Liutkus, and N. Ito, “The 2018 signal separation evaluation campaign,” in *Latent Variable Analysis and Signal Separation: 14th International Conference, LVA/ICA 2018, Surrey, UK*, 2018, pp. 293–305.
- [39] J. L. Roux, S. Wisdom, H. Erdogan, and J. R. Hershey, “Sdr - half-baked or well done?” 2018.
- [40] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [41] Y. Wu and K. He, “Group normalization,” in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 3–19.
- [42] P. Dhariwal and A. Q. Nichol, “Diffusion models beat GANs on image synthesis,” in *Advances in Neural Information Processing Systems*, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., 2021. [Online]. Available: <https://openreview.net/forum?id=AAWuCVzaVt>
- [43] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, 2017.

- [44] I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” in *International Conference on Learning Representations*, 2019. [Online]. Available: <https://openreview.net/forum?id=Bkg6RiCqY7>
- [45] P.-S. Huang, S. D. Chen, P. Smaragdis, and M. Hasegawa-Johnson, “Singing-voice separation from monaural recordings using robust principal component analysis,” in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 57–60.
- [46] D. D. Lee and H. S. Seung, “Learning the parts of objects by non-negative matrix factorization,” *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.
- [47] D. Fitzgerald, “Harmonic/percussive separation using median filtering,” *13th International Conference on Digital Audio Effects (DAFx-10)*, 01 2010.
- [48] Z. Rafii and B. Pardo, “Music/voice separation using the similarity matrix,” in *Proceedings of the 13th International Society for Music Information Retrieval Conference, ISMIR 2012*, ser. Proceedings of the 13th International Society for Music Information Retrieval Conference, ISMIR 2012, 2012, pp. 583–588, 13th International Society for Music Information Retrieval Conference, ISMIR 2012 ; Conference date: 08-10-2012 Through 12-10-2012.
- [49] P. Seetharaman, F. Pishdadian, and B. Pardo, “Music/voice separation using the 2d fourier transform,” in *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2017, pp. 36–40.
- [50] E. Postolache, G. Mariani, M. Mancusi, A. Santilli, L. Cosmo, and E. Rodolà, “Latent autoregressive source separation,” in *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, ser. AAAI’23/IAAI’23/EAAI’23. AAAI Press, 2023. [Online]. Available: <https://doi.org/10.1609/aaai.v37i8.26131>
- [51] A. Défossez, N. Usunier, L. Bottou, and F. Bach, “Music source separation in the waveform domain,” 2020. [Online]. Available: <https://openreview.net/forum?id=HJx7uJStPH>
- [52] Y. Wang, Y. Schiff, A. Gokaslan, W. Pan, F. Wang, C. De Sa, and V. Kuleshov, “InfoDiffusion: Representation learning using information maximizing diffusion models,” in *Proceedings of the 40th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, Eds., vol. 202. PMLR, 23–29 Jul 2023, pp. 36 336–36 354. [Online]. Available: <https://proceedings.mlr.press/v202/wang23ah.html>
- [53] N. Tishby, F. C. Pereira, and W. Bialek, “The information bottleneck method,” 2000.

LYRICALLY SPEAKING: EXPLORING THE LINK BETWEEN LYRICAL EMOTIONS, THEMES AND DEPRESSION RISK

Pavani Chowdary¹

Bhavyajeet Singh¹

Rajat Agarwal¹

Vinoo Alluri¹

¹ Music Cognition Group, International Institute of Information Technology, Hyderabad, India

{boddu.pavani, bhavyajeet.singh, rajat.agarwal}@research.iiit.ac.in, vinoo.alluri@iiit.ac.in

ABSTRACT

Lyrics play a crucial role in affecting and reinforcing emotional states by providing meaning and emotional connotations that interact with the acoustic properties of the music. Specific lyrical themes and emotions may intensify existing negative states in listeners and may lead to undesirable outcomes, especially in listeners with mood disorders such as depression. Hence, it is important for such individuals to be mindful of their listening strategies. In this study, we examine online music consumption of individuals at risk of depression in light of lyrical themes and emotions. Lyrics obtained from the listening histories of 541 Last.fm users, divided into At-Risk and No-Risk based on their mental well-being scores, were analyzed using natural language processing techniques. Statistical analyses of the results revealed that individuals at risk for depression prefer songs with lyrics associated with low valence and low arousal. Additionally, lyrics associated with themes of *denial*, *self-reference* and *blame* were preferred. This study opens up the possibility of an approach to assessing depression risk from the digital footprint of individuals and potentially developing personalized recommendation systems.

Keywords: depression, lyrics, lastfm, emotions, themes

1. INTRODUCTION

Depression is one of the leading causes of disability in young adults globally, according to the World Health Organization [1]. It has the potential to hinder and curb development in personal and social avenues of life, making it a debilitating condition. This underscores the imperative to identify and address it in the early stages.

Music plays an important role in regulating mood and emotions [2]. Musical preferences and music listening habits are known to invoke and reinforce moods and emotions and satisfy psychological needs [3, 4]. Emotionally vulnerable young adults were found to have more intense

relationships with music [5]. An increased emotional reliance on music was also observed in such individuals [6]. However, certain music engagement behaviors and strategies are associated with indicators of poor mental health and do not always lead to the alleviation of existing depressive symptoms [7]. Individuals who are depressed or at risk of depression are often unconscious of using music as a tool to improve emotional states [8], which might lead to adverse outcomes. This highlights the importance of addressing and studying music listening behaviors of individuals prone to depression risk for developing intervention methods to come up with listening strategies that may lead to positive outcomes.

Online music streaming platforms such as Spotify¹, Last.fm², and Apple Music³ offer their users a large variety of songs across genres and make it possible to study the musical digital footprints of their users. Last.fm allows the extraction of the listening histories of its users and the corresponding metadata, which prompted several studies that utilized Last.fm to study naturally occurring user listening behaviors in light of depression risk [9, 10, 11]. However, the relationship between lyrics, specifically the semantics and emotional connotations of lyrics, and depression has received little to no attention in the literature, while lyrics were found to play a crucial role in affecting emotional states [12]. Lyrics were also found to be essential for depicting *sadness* in music [13]. This study seeks to address this gap by examining the relationship between lyrical emotions and themes extracted from user listening histories and depression risk.

2. BACKGROUND AND RELATED WORK

We highlight previous studies that used online music listening histories from Last.fm and preferences to identify different trends and characteristics in music listening behaviors of individuals at risk of depression. Surana et al. [9] was the first such study, which used user-annotated tags from Last.fm, to identify emotion- and genre-tag preferences of individuals at risk of depression. The results of this study revealed that At-Risk individuals consume music that is tagged with emotions representing sadness, such as *sad*, *depressed*, *dead*, *low* and *miserable*, and belonging to



© P. Chowdary, B. Singh, R. Agarwal and V. Alluri. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** P. Chowdary, B. Singh, R. Agarwal and V. Alluri, "Lyrically Speaking: Exploring the Link Between Lyrical Emotions, Themes and Depression Risk", in *Proc. of the 25th Int. Society for Music Information Retrieval Conf.*, San Francisco, United States, 2024.

¹ www.spotify.com

² www.last.fm

³ music.apple.com

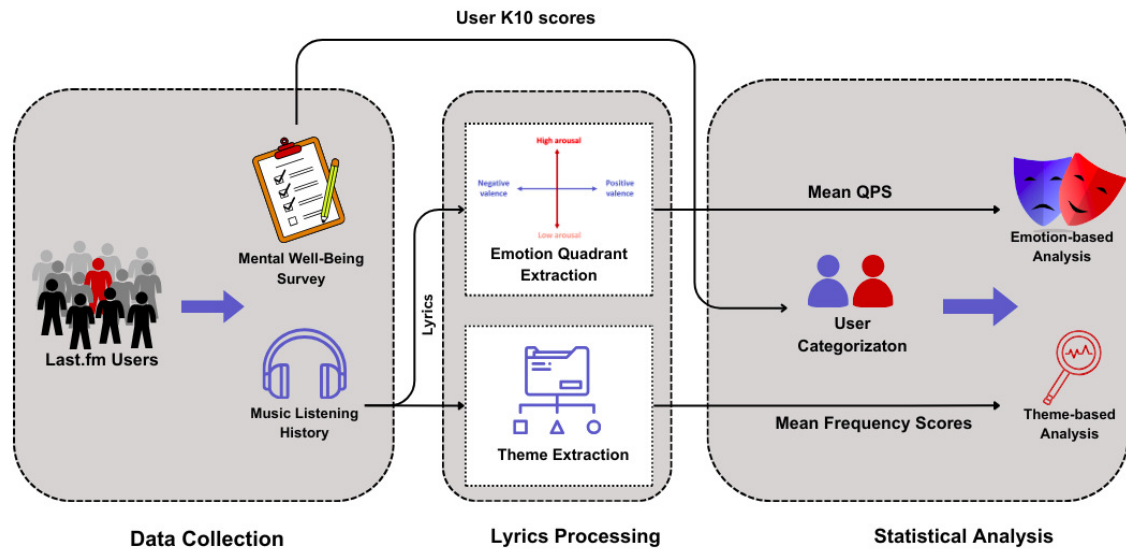


Figure 1: Methodology

genres such as *neo-psychedelic*, *dream-pop* and *indiepop*. In a later study, Surana et al. [10] studied emotions in relation to acoustic features of music as dynamic measures across the span of six months by dividing the listening history into sessions based on periods of inactivity to observe any dynamic patterns in music listening behaviors of At-Risk individuals. This study found that individuals at risk of depression rely more heavily on music and tend to listen to the same songs repeatedly. It was also found that they tend to listen to sad music for longer periods.

Shriram et al. [11] were the first to study lyrics in this context in terms of lyrical repetitiveness and compressibility. The results revealed that At-Risk individuals prefer music with lower lyrical simplicity (lower compressibility) and greater information content, especially for music that is characterized as sad.

However, no study to date has explored the link between lyrical emotions and themes extracted from online listening histories and depression risk, to the best of our knowledge. This link is crucial to investigate because lyrics reinforce negative states and, in dire situations, lead to maladaptive outcomes. The only study that has looked into lyrical themes associated with maladaptive listening strategies, which is known to be a proxy for depression risk [14], was done by Singh et al. [15]. They explored the link between lyrical themes extracted using DICTION⁴ and unhealthy music engagement strategies characterized by the Unhealthy-Healthy music scale (HUMS) [14], which indirectly indicates depression risk. This study revealed that individuals who engage in unhealthy and maladaptive listening strategies listen to music with lyrical themes representing *self-reference* and *blame*. However, this has been done in the context of online discourse surrounding depression on Reddit. This raises the question of whether similar behavior can be observed in the lyrical content derived from the listening histories of individuals at risk on music

streaming platforms. In this study, we investigated the relationship between individuals' online listening histories and their risk of depression in the context of lyrical emotions and themes.

Based on previous research in the field [9, 10, 11, 15], we hypothesize the following:

- Building on prior research demonstrating a preference for *sad* music among At-Risk individuals [9, 10], we hypothesize that these individuals exhibit a greater preference for music with lyrics associated with low valence and low arousal.
- In line with the established link between lyrical themes and unhealthy music engagement behaviors shown by Singh et al. [15], we hypothesize that At-Risk individuals consume music that is higher in terms of themes such as *self-reference*, and *blame*.

3. METHODOLOGY

Figure 1 summarizes the procedure used in our study, which is described as follows.

3.1 Dataset

We used the dataset from Surana et al. [9, 10] for our analysis. The dataset consists of the six-month music listening history of 541 Last.fm users (Mean Age = 25.4, SD = 7.3), of which 444 were male, 82 were female, and 15 identified as other. This data was acquired by means of a survey that was posted on Reddit⁵ and Facebook⁶ Last.fm pages. Informed consent was taken from the participants and participation was completely voluntary. They were informed that the study posed no risks and that their confidentiality would be maintained. The analysis was performed at a group level, ensuring no individuals could be identified.

⁴ www.dictionsoftware.com

⁵ www.reddit.com

⁶ www.facebook.com

3.1.1 Measure of Depression Risk

To assess mental well-being, Kessler’s Psychological Distress Scale (K10) questionnaire [16] is utilized, which measures psychological distress with a focus on symptoms of anxiety and depression. Following the approach of Surana et al. [9], participants scoring 29 or higher on the K10 questionnaire were classified as being in the “At-Risk” group for depression, while those scoring below 20 constitute the “No-Risk” group. Out of the total users, 193 individuals were in the No-Risk group, and 142 were in the At-Risk group.

3.1.2 Listening History and Lyrics

Lyrics for the tracks in the listening histories are extracted from Genius.com and MetroLyrics.com. Lyrics for approximately 76% of the entire repository of songs were obtained. Songs with no lyrics comprised around 4% of the dataset.

3.2 Lyrics Processing

3.2.1 Lyrics-Emotion Mapping

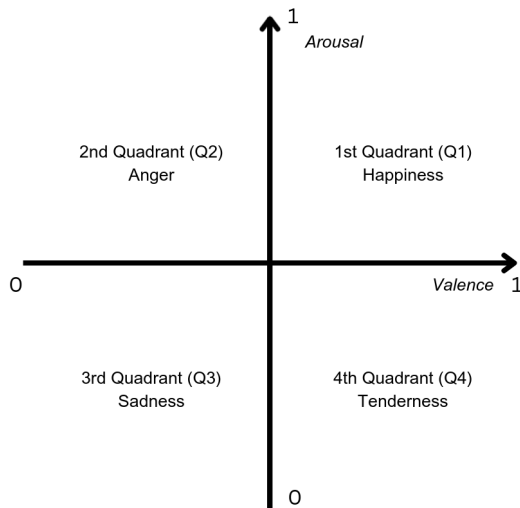


Figure 2: Two-dimensional Valence-Arousal space

Lyrics were projected onto the Russell’s Complex Model of Affect [17], which is used to organize emotions along two orthogonal dimensions: *Valence*, which represents pleasantness and *Arousal*, which represents energy. As can be seen in Figure 2, The first quadrant represents high valence and high arousal (happiness), the second quadrant represents low valence and high arousal (anger), the third quadrant represents low valence and low arousal (sadness), and the fourth quadrant represents high valence and low arousal (tenderness). We employed a model proposed in Agarwal et al. [18] to map each song’s lyrical content to a quadrant in the VA space. The architecture of the model is a deep neural network architecture that employs XLNet [19], an advanced bidirectional transformer, to perform multitask learning for emotional classification

based on song lyrics. The model is trained on the Moody-Lyrics [20] and MER [21] datasets, which consist of songs uniformly distributed across the four quadrants of the Russell’s Valence-Arousal circumplex model. The model was used to return a single quadrant label for each song by mapping the song’s lyrical content to the 2D Valence-Arousal space, which was used to categorize the tracks into one of the four quadrants.

To quantify user preferences for quadrant categories, we computed a quadrant prevalence (QPS) score for each quadrant, which is determined by the proportion of tracks from each user’s listening history, within the respective quadrants, as shown in Equation 1. The top 100 most frequently listened songs were identified and assigned weights based on listening frequency. The QPS for each user and quadrant was then calculated as the average weighted frequency of songs belonging to that specific quadrant in their listening history.

$$QPS(u_j, q_k) = \frac{\sum_{s_i \in L(u_j)} w(s_i) \cdot I(q(s_i) = q_k)}{\sum_{s_i \in L(u_j)} w(s_i)} \quad (1)$$

where,

$QPS(u_j, q_k)$: QPS of quadrant q_k for user u_j

s_i : a song in the top 100 songs of the user’s listening history $L(u_j)$

$w(s_i)$: the weight (listening frequency) of song s_i .

$q(s_i)$: quadrant assigned to song s_i

$I(\cdot)$: indicator function that equals 1 if the condition inside is true (song s_i belongs to quadrant q_k) and 0 otherwise.

3.2.2 Lyrics-Semantic Themes Mapping

Following a similar approach to Singh et al. [15], we used DICTION to analyze the lyrics and identify underlying semantic themes. DICTION operates through the use of dictionaries that contain lists of words associated with specific linguistic, emotional and cognitive contexts. There are 5 themes and 35 sub-themes in total. We chose the themes *Self-reference*, *Blame*, *Optimism*, *Hardship*, *Satisfaction*, *Inspiration*, *Exclusion* and *Denial*. The frequency scores corresponding to all the themes and sub-themes for each song were obtained, based on the occurrence of the words from the lyrics in the dictionary lists. Similar to the quadrant prevalence scores, mean frequency scores (MFS) were computed for all the themes per user, as shown in Equation 2.

$$MFS(u_j, t_k) = \frac{\sum_{s_i \in L(u_j)} w(s_i) \cdot T_k(s_i)}{\sum_{s_i \in L(u_j)} w(s_i)} \quad (2)$$

where,

$MFS(u_j, t_k)$: MFS of theme t_k for user u_j

s_i : a song in the top 100 songs of listening history $L(u_j)$

$w(s_i)$: the weight (listening frequency) of song s_i .

$T_k(s_i)$: the frequency score assigned to song s_i for theme t_k .

3.3 Statistical Testing

We divided the users into At-Risk and No-Risk groups based on their K10 scores, as mentioned before. For each quadrant, we performed a two-tailed Mann-Whitney U (MWU) test on the quadrant prevalence score between the At-Risk and No-Risk groups. Similarly, a MWU test was performed for the mean frequency score between At-Risk and No-Risk groups, for each theme selected.

4. STATISTICAL TESTING AND RESULTS

4.1 Lyrical Emotion-Based Results

A statistically significant difference ($p < 0.05$) was observed for the QPS corresponding to Q3, which is representative of low valence and low arousal values (U-statistic = 15727.5, $p = 0.02$), with a higher median for the At-Risk group, as can be seen in 3. We observed no significant differences between the distributions of the At-Risk and No-Risk groups for the other quadrants. The same trend was observed when the top 250 songs were considered.

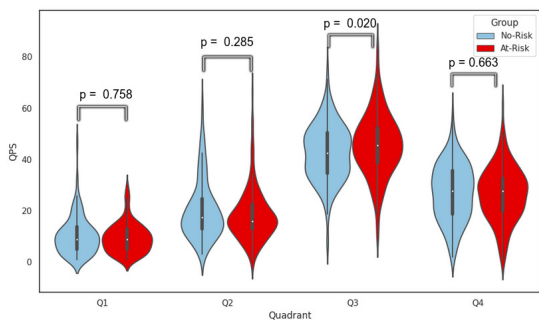


Figure 3: Violin plots of mean QPS per quadrant for At-Risk and No-Risk groups

4.2 Lyrical Theme-based Results

We found significant differences ($p < 0.05$) between At-Risk and No-Risk groups in the case of *denial*, *self-reference* and *blame*. As can be inferred from Table 1, The median for the At-Risk group is higher in the cases of the themes *denial*, *self-reference* and *blame*.

	Theme	U-statistic	p-value
At-Risk > No-Risk	Denial	15950.0	0.010
	Self-reference	15968.0	0.009
	Blame	15691.5	0.023

Table 1: MWU Test results for the Mean Frequency Scores between the At-Risk and No-Risk groups; Here At-Risk > No-Risk refers to the themes where the median is greater in the case of the At-Risk group

5. DISCUSSION

This study is the first of its kind to explore the association between risk for depression and the emotional and the-

matic connotations of the lyrical content of the music individuals engage with online, as opposed to studies in lab settings or self-reported data. Our results are in concordance with our initial hypotheses, in addition to revealing novel findings.

The At-Risk group exhibited a higher median score for Q3 (low valence, low arousal) than the No-Risk group. This finding suggests a greater prevalence of sadness-related lyrical content in the music listened to by the At-Risk group. The stronger association of the At-Risk group with sadness aligns with the results of past research studies on the topic. These results suggest that At-Risk individuals tend to consume music with lyrics that reflect their negative emotional states.

As hypothesized, the themes *self-reference* and *blame* were more prevalent in the At-Risk group. Additionally, the themes *denial* was also shown to be preferred by At-Risk individuals. Since the themes of *blame* and *self-reference* in lyrics were found to be associated with unhealthy listening strategies [15], listening to music associated with these themes would not be beneficial to individuals at risk of depression, and in some cases may lead to negative outcomes. This highlights the importance of mindful consumption strategies for music for people at risk of depression.

In conclusion, our results show that certain music engagement strategies are maladaptive in nature and should be avoided to prevent the worsening of mood, building on top of previous studies in the area. These results can potentially aid in developing intervention strategies based on lyrical content that should be avoided for better outcomes from music listening.

5.1 Limitations

A limitation of this study is the exclusive focus on lyrical emotions and themes. The interaction of these with the acoustic properties of music and lyrical complexity, as well as in the context of depression risk, could be explored, which could possibly yield a better understanding. The interaction between lyrical themes and emotional connotations is also something that is yet to be studied. Another limitation is that this study exclusively focuses on music with English lyrics. We have also used a predetermined set of themes offered by DICTION, which may not be enough to capture several lyrical themes. An approach to solve this would be to use Large Language Models (LLMs) to generate themes from a repository of songs and then using them for the scoring.

5.2 Future Work

The results from this paper can be used in building music recommendation systems for depressed individuals that tailor the recommendations, keeping in mind the emotions and themes that are associated with mood worsening and maladaptive behaviors to maximize the positive outcomes through music listening. These results could be combined with other measures associated with such behaviors. This work also opens up the possibility of early depression risk

prediction from online music listening behaviors, in terms of lyrics, by cementing lyrical emotions and themes as indicators for depression risk. These measures, in addition to acoustic features [10] and other indicators such as lyrical complexity and social tags [11, 9], could potentially be used to develop a multi-modal depression risk prediction system.

6. REFERENCES

- [1] W. Depression, "Other common mental disorders: global health estimates," *Geneva: World Health Organization*, vol. 24, 2017.
- [2] A. C. North, D. J. Hargreaves, and J. J. Hargreaves, "Uses of music in everyday life," *Music perception*, vol. 22, no. 1, pp. 41–77, 2004.
- [3] M. Baltazar and S. Saarikallio, "Toward a better understanding and conceptualization of affect self-regulation through music: A critical, integrative literature review," *Psychology of Music*, vol. 44, no. 6, pp. 1500–1521, 2016.
- [4] T. Schäfer, "The goals and effects of music listening and their relationship to the strength of music preference," *PLoS one*, vol. 11, no. 3, p. e0151634, 2016.
- [5] D. J. Hargreaves, A. C. North, and M. Tarrant, "135Musical Preference and Taste in Childhood and Adolescence," in *The Child as Musician: A handbook of musical development*. Oxford University Press, 06 2006.
- [6] K. S. McFerran, S. Garrido, and S. Saarikallio, "A critical interpretive synthesis of the literature linking music and adolescent mental health," *Youth & Society*, vol. 48, no. 4, pp. 521–538, 2016.
- [7] J. Stewart, S. Garrido, C. Hense, and K. McFerran, "Music use for mood regulation: Self-awareness and conscious listening choices in young people with tendencies to depression," *Frontiers in Psychology*, vol. 10, 2019.
- [8] K. McFerran and S. Saarikallio, "Depending on music to feel better: Being conscious of responsibility when appropriating the power of music," *The Arts in Psychotherapy*, vol. 41, 01 2013.
- [9] A. Surana, Y. Goyal, M. Shrivastava, S. Saarikallio, and V. Alluri, "Tag2risk: Harnessing social music tags for characterizing depression risk," *arXiv preprint arXiv:2007.13159*, 2020.
- [10] A. Surana, Y. Goyal, and V. Alluri, "Static and dynamic measures of active music listening as indicators of depression risk," Oct. 2020.
- [11] J. Shriram, S. Paruchuri, and V. Alluri, "How much do lyrics matter? analysing lyrical simplicity preferences for individuals at risk of depression," Aug. 2021.
- [12] A. M. Demetriou, A. Jansson, A. Kumar, and R. M. Bittner, "Vocals in music matter: the relevance of vocals in the minds of listeners," in *International Society for Music Information Retrieval Conference*, 2018. [Online]. Available: <https://api.semanticscholar.org/CorpusID:53875501>
- [13] E. Brattico, V. Alluri, B. Bogert, T. Jacobsen, N. Vartiainen, S. Nieminen, and M. Tervaniemi, "A functional mri study of happy and sad emotions in music with and without lyrics," *Frontiers in Psychology*, vol. 2, 2011.
- [14] S. Saarikallio, C. Gold, and K. McFerran, "Development and validation of the healthy-unhealthy music scale," *Child and Adolescent Mental Health*, vol. 20, 05 2015.
- [15] B. Singh, K. Vaswani, S. Paruchuri, S. Saarikallio, P. Kumaraguru, and V. Alluri, "'help! i need some music!': Analysing music discourse depression on reddit," *PLoS ONE*, 2023.
- [16] R. Kessler, G. Andrews, L. Colpe, H. EE, D. Mroczek, S.-L. Normand, E. Walters, and A. Zaslavsky, "Short screening scales to monitor population prevalences and trends in non-specific psychological distress," *Psychological medicine*, vol. 32, pp. 959–76, 09 2002.
- [17] J. Russell, "A circumplex model of affect," *Journal of Personality and Social Psychology*, vol. 39, pp. 1161–1178, 12 1980.
- [18] Y. Agrawal, R. G. R. Shanker, and V. Alluri, *Transformer-Based Approach Towards Music Emotion Recognition from Lyrics*. Springer International Publishing, 2021, p. 167–175. [Online]. Available: http://dx.doi.org/10.1007/978-3-030-72240-1_12
- [19] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," 2020.
- [20] E. Çano and M. Morisio, "Moodylyrics: A sentiment annotated lyrics dataset," in *Proceedings of the 2017 international conference on intelligent systems, metaheuristics & swarm intelligence*, 2017, pp. 118–124.
- [21] R. Malheiro, R. Panda, P. Gomes, and R. P. Paiva, "Emotionally-relevant features for classification and regression of music lyrics," *IEEE Transactions on Affective Computing*, vol. 9, pp. 240–254, 2018. [Online]. Available: <https://api.semanticscholar.org/CorpusID:21791683>

A STEM-AGNOSTIC SINGLE-DECODER SYSTEM FOR MUSIC SOURCE SEPARATION BEYOND FOUR STEMS

Karn N. Watcharasupat Alexander Lerch

Music Informatics Group, Georgia Institute of Technology, Atlanta, GA, USA
{kwatcharasupat, alexander.lerch}@gatech.edu

ABSTRACT

Despite significant recent progress across multiple sub-tasks of audio source separation, few music source separation systems support separation beyond the four-stem vocals, drums, bass, and other (VDBO) setup. Of the very few current systems that support source separation beyond this setup, most continue to rely on an inflexible decoder setup that can only support a fixed pre-defined set of stems. Increasing stem support in these inflexible systems correspondingly requires increasing computational complexity, rendering extensions of these systems computationally infeasible for long-tail instruments. We propose Banquet, a system that allows source separation of multiple stems using just one decoder. A bandsplit source separation model is extended to work in a query-based setup in tandem with a music instrument recognition PaSST model. On the MoisesDB dataset, Banquet — at only 24.9 M trainable parameters — performed on par with or better than the significantly more complex 6-stem Hybrid Transformer Demucs. The query-based setup allows for the separation of narrow instrument classes such as clean acoustic guitars, and can be successfully applied to the extraction of less common stems such as reeds and organs.

1. INTRODUCTION

Music Source Separation (MSS) is the task of separating a musical audio mixture into its constituent components, commonly referred to as stems. The releases of DSD100 [1] and MUSDB18 [2, 3], both being four-stem MSS datasets, have defined a de-facto standard, with nearly every major work since relying on the four-stem *vocals, bass, drum*, and *others* (VDBO) setup [4–19]. While this has significantly improved the comparability and reproducibility of the task, it has also disproportionately favored the VDBO setup. Very few works have tackled MSS beyond the VDBO setup, each relying on datasets with significant limitations: Wang et al. [20] relied on MedleyDB [21, 22], whose stem ontology is somewhat unfriendly to source separation, Manilow et al. [23] relied on the syn-

thetically generated Slakh dataset [24], and others relied on proprietary data inaccessible to other research groups [11, 18], limiting reproducibility. The recently released MoisesDB [25], a multitrack source separation dataset, attempts to address these limitations, particularly in terms of stem availability and taxonomy. This aims at broadening the task beyond VDBO based on publicly available data. However, to the best of our knowledge, while MoisesDB was used in the 2023 Sound Demixing Challenge (SDX) [26], no published system has utilized MoisesDB for source separation beyond VDBO yet.

In this work, we propose Banquet,¹ a query-based source separation model that can separate an arbitrary number of stems using just one set of stem-agnostic encoder and decoder, and a pre-trained feature extractor [27]. Our model was adapted from the cinematic audio source separation Bandit model [28], which was in turn adapted from the music source separation Bandsplit RNN model [17]. Bandit significantly reduces the complexity of Bandsplit RNN by adopting a common-encoder approach with stem-specific decoders. In this work, we take the complexity reduction further by switching to a query-based setup, using only one decoder shared amongst all possible stems. Performance evaluation on MoisesDB demonstrated separation performance above oracle for drum and bass, state-of-the-art for guitar and piano, and at least 7.4 dB SNR for vocals. Our system additionally provided support for fine-level stem extraction currently available only in a few MSS systems.

2. RELATED WORK

Nearly every major MSS works since 2017 have relied on the VDBO setup. Early systems [4, 6, 29], including OpenUnmix [8], were usually Time-Frequency (TF) masking models with LSTM forming the core of the systems, with some experimenting with densely-connected convolutional systems [5, 12]. Beginning with Wave-U-Net [7], the U-Net architecture became a popular choice for MSS, with notable models such as Demucs [9, 10, 14, 18], Spleeter [11], ByteSep [13], and KUIELab-MDX-Net [15] all being some variations of a U-Net. More recently, Bandsplit RNN [17] became one of the few state-of-the-art systems to not rely on a U-Net setup. This was followed by the Bandsplit RoPE Transformer model [19] topping the



© K. N. Watcharasupat, and A. Lerch. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).
Attribution: K. N. Watcharasupat, and A. Lerch, “A Stem-Agnostic Single-Decoder System for Music Source Separation Beyond Four Stems”, in *Proc. of the 25th Int. Society for Music Information Retrieval Conf.*, San Francisco, United States, 2024.

¹ Banquet is a portmanteau of **Q**uery-based **B**andit. Code available at github.com/kwatcharasupat/query-bandit. Last accessed 24 July 2024.

leaderboard of SDX 2023 [26]. Of existing open-source systems, very few offer separation functionality beyond the VDBO setup. Spleeter [11] supports 5-stem separation with VDBO and piano. HT-Demucs [18] supports a 6-stem setup with VDBO, piano, and guitar.

2.1 Conditional source separation

The systems mentioned above were mostly designed with either stem-specific models, stem-specific decoders, or a shared decoder with predetermined outputs. As a result, these systems are not particularly amenable to the addition of new stems, especially if these new stems have limited data availability. Below we review some of the common approaches for conditional source separation that may be useful for extending existing systems beyond VDBO.

Meseguer-Brocal and Peeters [30] were likely amongst the first to attempt a conditioned U-Net for source separation using a single decoder. They used multiple feature-wise linear modulation (FiLM) [31] layers within the encoder to perform MSS in a VDBO setup. Slizovskaia et al. [32] used a similar setup with FiLMs either throughout the encoder, at the bottleneck layer, or at the final decoder layer. The systems in [32] were tested on the 13-instrument URMP dataset [33], with up to 4 active instruments in any recording, but all performed poorly in terms of mean signal-to-distortion ratio (SDR). Lin et al. [34] proposed a joint separation-transcription U-Net system, which performed well for string and brass instruments in URMP, but struggled on woodwind instruments. The system in [34] used FiLMs throughout the encoder with a query embedding from another convolutional model, and across all skip connections with transcription embeddings.

Lee et al. [35] proposed a U-Net with two methods of less aggressive conditioning with examples beyond VDBO, but only provided objective results for a VDBO setup on MUSDB18. Wang et al. [20] also proposed a U-Net, with FiLM conditioning only at the bottleneck layer. The system in [20] was able to support a substantial number of stems beyond VDBO with the caveat that its reported performance is significantly below contemporary models for VDBO stems. Gfeller et al. [36] utilized a FiLM-conditioned wave-to-wave U-Net to perform one-shot conditional audio filtering. Similar approaches were also adopted in Choi et al. [37] and Jeong et al. [38] for MSS, in Chen et al. [39] for source activity-queried separation, in Kong et al. [40] for universal source separation, and in Liu et al. [41, 42] for language-queried source separation. These works [36–42] applied FiLM or generalizations thereof to nearly every single layer of the network, significantly increasing the computational complexity of the system. We surmise that the apparent need for multiple conditioning in a U-Net is probably due to the nature of its information flow [43], which may require a significant number of information streams to be conditioned to achieve acceptable performance.

In a different direction, source separation systems relying on audio embedding “distances” have also been developed, notably with Le Roux et al. in [23, 44, 45]. In 2018,

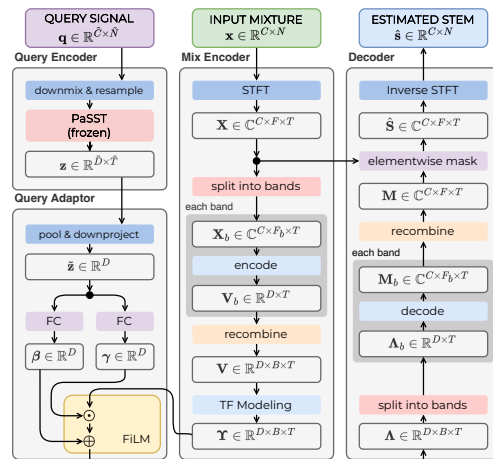


Figure 1. Overview of the Banquet System.

Kumar et al. [46] presented an early work using Euclidean audio embedding distance from a “query” embedding to inform music source separation. A similar system using a Gaussian mixture posterior in lieu of standard distance was proposed in [44]. Hierarchical masking [23] was later utilized to allow the extraction of stems at multiple levels of specificity. More recently, source separation systems with audio embedding in a low-dimensional hyperbolic space have been developed to allow music [45] and speech [47] source separation with some degrees of control on the specificity of the extraction. Uniquely, Samuel et al. [48] proposed a network-generating network approach for instrument-conditioned source separation.

3. PROPOSED SYSTEM

The overview of the proposed Banquet system is shown in Fig. 1. The system is a single-encoder single-decoder adaptation of Bandit [28], that takes in a mixture signal x and a query signal q , and extracts a stem estimate \hat{s} from the mixture signal of the “same” stem type as the query signal using a complex-valued TF mask. This is done by (i) encoding the mixture into a subband-level time-varying embedding tensor Υ , (ii) encoding the query into a single-vector representation \tilde{z} , (iii) adapting the mixture embedding, conditioned on the query, into a stem-specific embedding $\Lambda = \mathcal{Q}(\Upsilon; \tilde{z})$, then (iv) decoding the Λ to a TF mask M that is used to obtain the source estimate.

3.1 Bandit encoder

The encoder module of the system used in this work is the *musical* variant of the Bandit encoder, with $B = 64$ bands. Specifically, given an input mixture $x \in \mathbb{R}^{C \times N}$ with C channels and N samples, a short-time Fourier transform (STFT) of x is computed to obtain $X \in \mathbb{C}^{C \times F \times T}$ with a frame size of $N_{FFT} = 2(F - 1) = 2048$ and 75% overlap. The STFT is then split into overlapping subbands as detailed in [28]. Each of the subbands is then viewed as a real-valued tensor in $\mathbb{R}^{2CF \times T}$, passed through a layer norm and an affine transformation with $D = 128$ neurons to obtain $V_b \in \mathbb{R}^{D \times T}$. These tensors are then stacked to

obtain $\mathbf{V} \in \mathbb{R}^{D \times B \times T}$. TF modeling is then applied on \mathbf{V} to obtain \mathbf{Y} using 8 pairs of residual gated recurrent units (GRUs), the first of each pair operating along the time axis and the second along the band axis.

Note that this TF modeling is the only part of the model that is recurrent across either the time or the subband axes. The rest of the encoder and the decoder operate in a subband-wise manner identically for any time frame.

3.2 Query encoding

To obtain the query embedding, a PaSST model [27] trained on the OpenMIC-2018 dataset [49] is used. The 20 instruments in OpenMIC span all coarse-level classes of MoisesDB, except *other*. For compatibility, each query signal is downmixed to mono and downsampled to 32 kHz before being fed to PaSST. Although the query feature extractor could, in theory, be jointly trained with the rest of the system, preliminary experiments showed that this can result in considerable instability during training, especially if the query feature extractor is not at least pretrained. Due to the size and complexity of PaSST, the query feature extractor is fully frozen in this work. The embedding from the PaSST variant used is a time series with a feature dimension of $\tilde{D} = 784$. The embedding is averaged over time and linearly down-projected to obtain $\tilde{\mathbf{z}} \in \mathbb{R}^D$.

3.3 Query-based adaptation

In the original Bandit system [28], each stem was estimated through a dedicated decoder. As a result, \mathbf{Y} typically contains information from all stems, with most of the “separation” occurring within each of the decoders. This is evident in the fact that the encoder of a Bandit system trained on the cinematic audio Divide and Remaster (DnR) dataset [?] could be successfully used in a 4-stem MSS on the MUSDB18-HQ dataset [2] with separation quality on par with Open-Unmix [28].

In this work, only a single decoder is responsible for mask estimation for any stem. As a result, the query-based adaptation $\mathcal{Q}: (\mathbb{R}^{D \times B \times T}, \mathbb{R}^D) \mapsto \mathbb{R}^{D \times B \times T}$ has an important role in filtering out irrelevant information from \mathbf{Y} , or at least “hinting” to the decoder the nature of the target stem. A single FiLM layer is used to map from the mixture embedding to the stem-specific embedding, that is,

$$\mathbf{A}[d, b, t] = \gamma[d] \cdot \mathbf{Y}[d, b, t] + \beta[d], \quad \forall d, b, t, \quad (1)$$

where modulating variables $\gamma, \beta \in \mathbb{R}^D$ are obtained from a two-layer nonlinear affine map of $\tilde{\mathbf{z}}$. This is similar to the conditioning method used in [20].

Crucially, note that the modulating variables are not subband-specific. Due to the nature of the TF modeling module within the encoder, features of \mathbf{Y} are already aligned across subbands and time frames. Moreover, BSRNN-like models only contain one stream of information flow, with a clear bottleneck, thus lending itself to the global conditioning mechanism significantly more than, for example, U-Net-style models in [20, 41, 42].

The use of embedding-based query, as opposed to one-hot class-based query, provides significant practical flexibility in adding new instruments as data become available or in adjusting the level of specificity in the querying, as these can be done via finetuning with no architectural changes to the model. Moreover, class-based query can be emulated in an embedding-based system but not vice versa.

3.4 Bandit decoder

The decoder used is identical in structure to that in [28]. The major difference is that there is only one stem-agnostic decoder. Given a conditioned embedding tensor \mathbf{A} , the embedding tensor is split into subband-level representation $\mathbf{A}_b = \mathbf{A}[:, b, :]$. Each \mathbf{A}_b is passed through a layer norm and a gated linear unit (GLU) to obtain a real-valued tensor $\mathbb{R}^{2CF_b \times T}$ which is then viewed as a complex-valued tensor $\mathbf{M}_b \in \mathbb{C}^{C \times F_b \times T}$. Frequency-domain overlap-add is then applied to obtain the full-band mask using

$$\mathbf{M}[c, f, t] = \sum_{b=0}^{B-1} \frac{\mathbf{W}[b, f] \cdot \mathbf{M}_b[c, f - \min \mathfrak{F}_b, t]}{\sum_{k=0}^{B-1} \mathbf{W}[k, f]} \quad (2)$$

Finally, the source estimates are then obtained using elementwise masking $\hat{\mathbf{S}} = \mathbf{X} \circ \mathbf{M}$.

3.5 Loss function

The loss function used in this work is the multichannel version of the L1SNR loss proposed in [28]. The contribution for each sample of the loss function is given by

$$\mathcal{L}(\hat{\mathbf{s}}; \mathbf{s}) = \mathcal{D}(\hat{\mathbf{s}}; \mathbf{s}) + \mathcal{D}(\Re \hat{\mathbf{S}}; \Re \mathbf{S}) + \mathcal{D}(\Im \hat{\mathbf{S}}; \Im \mathbf{S}), \quad (3)$$

$$\mathcal{D}(\hat{\mathbf{y}}; \mathbf{y}) = 10 \log_{10} \frac{\|\text{vec}(\hat{\mathbf{y}} - \mathbf{y})\|_1 + \epsilon}{\|\text{vec}(\mathbf{y})\|_1 + \epsilon}, \quad (4)$$

where $\hat{\mathbf{s}} = \text{iSTFT}(\hat{\mathbf{S}})$, \mathbf{s} and \mathbf{S} are defined similarly for the ground truth, $\text{vec}(\cdot)$ is the vectorization operator, and $\epsilon = 10^{-3}$ for stability.

4. DATA AND EXPERIMENTAL SETUP

This work utilizes the MoisesDB dataset [25], which consists of 240 songs from 47 artists, in stereo format at 44.1 kHz. MoisesDB defined their stem ontology with more than 30 fine-level classes, which are then grouped into 11 coarse-level classes [25, Table 2]. Due to the lack of official splits for MoisesDB, we performed a five-fold split² on the dataset stratified by genres. The first three splits are used as the training set, the fourth as the validation set, and the last as the test set.

4.1 Query extraction

For each possible stem of each song, a 10-second chunk of the clean audio of the same stem is extracted as the query signal. This is done by computing a time series of onset strength for each stem and then aggregating the mean onset

² The splits are available in the repository. Note that not all stems contain a sufficient number of data points to be split into a five-fold validation setup. As a result, some stems are only present in a subset of folds.

strength for each 10-second sliding window with a hop size of 512 samples. The 10-second window with the strongest average onset is taken as the query signal. A t-SNE plot of the query embedding is shown in Fig. 2. While clusters can be clearly seen amongst related stems, it can also be seen that there are varying degrees of non-separability of the embedding between fine-level stems.

4.2 Training

Each model was trained using an NVIDIA H100 GPU (80 GB) for up to 150 epochs, unless otherwise stated. A training epoch consists of 8192 mixture-query pairs, with a batch size of 4. We used Adam optimizer with an initial learning rate of 10^{-3} and a decay factor of 0.98 per epoch.

In the default sampling strategy, a random song is chosen, a random trainable stem for that song is chosen as the target stem, then a random chunk of 6 s is chosen. If the current target chunk has an RMS below -36 dBFS, a new random chunk is chosen for up to 10 more trials. Otherwise, the threshold is dropped to -48 dBFS for another 10 trials. If a suitable chunk is still not found, the next random chunk is chosen regardless of RMS. A pre-extracted query of the same stem is then randomly chosen from the available pool of songs, including the song of the mixture.

4.3 Testing and inference

During testing and inference, each track is split into 6-s segments with a hop size of 0.5 s, as per [17]. The estimated stems were then reconstructed into a full track using time-domain overlap-add with a Hann window. The Banquet models are tested in two scenarios: one using a query from a different song, and another using a query from the same song (SSQ). In different-song querying, the query song for each stem is randomly chosen from another song within the test split that contains the stem. When possible, the query song is chosen so that it is from the same genre as the mixture song but from a different artist. Otherwise, a song from any genre with a different artist is chosen.

4.4 Evaluation metric

In this work, we report the full-track multichannel signal-to-noise ratio (SNR)³ as the main metric. Specifically, for a test signal \hat{s} and a reference signal s , both in $\mathbb{R}^{C \times N}$, the SNR is computed by

$$\text{SNR}(\hat{y}; y) = 10 \log_{10} \left(\frac{\|s\|_F^2}{\|\hat{s} - s\|_F^2} \right). \quad (5)$$

5. RESULTS AND DISCUSSION

In this section, we provide the results and discussion of our experiments. Section 5.1 discusses pretraining of the Bandit/Banquet encoder. Section 5.2 trials the use of the query-based setup on a subset of vocals, drums, and bass

³ Signal-to-interference ratio (SIR) and signal-to-artifact ratio (SAR) were not computed as the number of the constituent stems can be large, making the required subspace projection intractable and/or unreliable. It is also unclear if coarse-level ground truth or fine-level ground truth should be used for such a projection. See [50–52] for background.

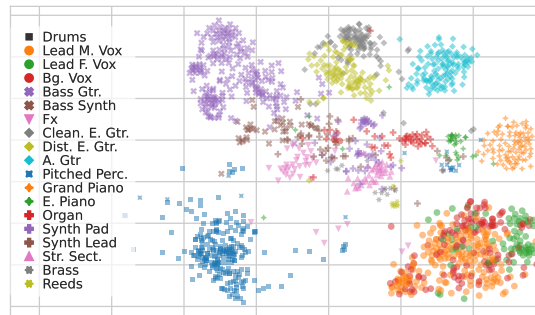


Figure 2. A t-SNE plot of the PaSST embeddings of the query signals. Stems from the same coarse-level grouping, as defined by [25], share the same symbol.

stems. Section 5.3 extends the system to include fine-level stems from *guitar* and *piano* families. Finally, Section 5.4 attempts to perform extraction on all possible fine-level stems with sufficient data.

5.1 Encoder pretraining

Preliminary experiments indicated that encoder pretraining is an important step to stabilize the training of the query-based model, especially as the number of query stems grows. The encoder pretraining is done with a common-encoder multi-decoder setup similar to [28] with a VDBO setup for 100 epochs. The VDBO decoders were discarded and the encoder was used for subsequent experiments. The performance of the pretrained model is shown in Table 1, with performance above oracle ideal ratio mask (IRM) for drums and bass, and on par with HT-Demucs for vocals.⁴

5.2 Learning to separate from queries

As a first step to verify the query-based ability of the model, a Banquet model is trained to extract only *lead female singer*, *lead male singer*, *drums*, and *bass* stems, referred to as the Q:VDB setup. We experimented with training from scratch, using a frozen pretrained encoder (FE), and using a trainable pretrained encoder (TE). While the frozen-encoder setup did not demonstrate any sign of overfitting during the training, the trainable-encoder system demonstrated (very slight) overfitting. As a result, an additional setup with data augmentation (DA) was attempted with the trainable encoder setup, using simple stem-wise within-song random gain (up to ± 6 dB), random time shifting, polarity inversion, and channel swapping.

The results are shown in Table 2. All three variants with pretrained encoder provided better performance than the model trained from scratch, except for drums in the trainable-encoder model without DA being 0.1 dB lower. Thus, for all subsequent experiments, the encoder is always pretrained. Without DA, there was no clear benefit to unfreezing the encoder. However, in a trainable en-

⁴ All coarse-level results for oracle methods, HT-Demucs, and Spleeter were recomputed only on the test set using song-wise results from github.com/moises-ai/moises-db. The song-wise results were missing for five of the songs (as of 6 April 2024), two of these belong in the test set, thus the aggregates were computed over 46 songs instead of 48 songs.

Table 1. Median SNR of the models trained on the VDBO setup, evaluated on the test set of MoisesDB.

Model	Median SNR (dB):	Vocals	Drums	Bass	Other
Bandit [28]		9.1	9.9	10.6	6.4
HT-Demucs [18]		9.1	11.0	12.2	7.3
Spleeter [11]		7.4	6.6	6.8	5.0
Oracle IRM		10.3	9.2	8.8	7.6

Table 2. Median SNR of Banquet models on the Q:VDB setup, evaluated with different-song queries.⁶

Pretrained Enc.	FE	DA	Female Vox	Male Vox	Drums	Bass
N	N	N	8.3	7.2	9.4	9.4
Y	Y	N	9.8	7.6	9.9	10.2
Y	N	N	9.8	8.0	9.3	9.8
Y	N	Y	10.2	8.0	10.1	10.8

coder system with DA, slight to moderate improvements were observed across all but the male vocal stem. Note, however, that allowing full-model training significantly increases the number of trainable parameters from 13.5 M to 24.9 M thus the computational cost and training time also increases accordingly. The performances of the drums and bass stems are on par or better than the dedicated-stem setup in Table 1. Generally, the models perform better on female vocals than on male vocals.

5.3 Extending to guitar and piano

Amongst systems that tackled MSS beyond four stems, the next two stems beyond VDBO are usually guitar and piano, due to their high prevalence within pop/rock music. The set of possible queries is thus extended from Q:VDB to also include *acoustic guitar*, *clean electric guitar*, *distorted electric guitar*, *grand piano*, and *electric piano*. This is referred to as the Q:VDBGP setup. Due to the significantly lower number of available training data for guitar and piano stems, we also experimented with a balanced sampling (BS) strategy. In this strategy, a random stem is first chosen as the target stem, then a random song containing that stem is chosen. The remainder of the sampling process is the same as the default. This strategy ensures that every stem has a similar number of training pairs, but distorts the “natural” distribution of stem occurrences.

For comparability with existing systems, the inference outputs of fine-level stems in this setup were added together to form their respective coarse-level predictions.⁷ Coarse-level results are shown in Table 3. Fine-level results for trainable-encoder models are shown in Table 4.

At the coarse level, most variants of Banquet continue to perform above the oracle IRM for drums and bass. With the default-sampling trainable encoder systems, the Banquet performed better than HT-Demucs on guitar and piano. Without DA, balanced sampling generally did not lead to consistent improvements for guitar and piano. With balanced sampling and DA on a trainable-encoder model,

⁶ Median results for the same-song query and different-song query are within 0.2 dB of each other.

⁷ The ground truth signals for are the full coarse-level tracks, e.g. *vocals* ground truth include contributions from *background vocals* even if we do not have *background vocals* in the predictions.

Table 3. Coarse-level performance of the Banquet models with different-song queries on the Q:VDBGP setup

Model	FE	DA	BS	Vox	Lead Vox	Drums	Bass	Guitar	Piano
Banquet	Y	N	N	8.0	7.9	9.8	10.5	2.3	0.8
			Y	7.9	7.7	9.6	10.5	2.2	0.9
	N	N	N	7.4	8.0	9.6	10.6	3.0	2.3
			Y	7.6	7.7	9.3	10.2	2.9	2.5
	Y	N		7.8	7.9	10.1	10.9	3.2	2.2
		Y		7.6	7.9	9.5	11.0	3.3	2.5
HT-Demucs (VDBGPO)				8.9	—	11.6	12.4	2.4	1.7
Spleeter (VDBPO)				7.0	—	6.9	6.7	—	0.7
Oracle IRM				10.0	—	9.6	7.8	5.2	5.0

Bold: best Banquet model **and/or** best non-oracle model.

however, slight gains in median SNRs of guitar and piano were observed, albeit at the cost of vocals and drum SNRs.

At the fine level, the model performance follows a similar trend to that of the coarse level. Drums and bass continue to perform above the oracle IRM, while both lead vocals performed close to the IRM. Guitar and piano performances are still well below IRM. Interestingly, it appears that querying with excerpts from the same or different track did not affect the model performance for most stems except for electric piano. This is likely due to both the small sample size of electric piano limiting generalizability, and the highly diverse set of possible timbres thus the intertwined nature of both the query embedding and the target audio with other keyboard instruments. The ability of the model to query with stems from different tracks is a double-edged sword, however, since this also means that the model is somewhat insensitive to fine differences in timbre between different renditions of the “same” instruments. This could potentially limit its usefulness when applied to a scenario where multiple target stems have very similar timbres.

5.4 Extending beyond guitar and piano

The results for the Q:VDBGP setup demonstrated that the model is able to learn to extract 5 additional stems. In this experiment, we extend the set of possible queries to include all remaining stems with at least one data point per fold: *effects*, *pitched percussion*, *organs & electronic organs*, *synth pad*, *synth lead*, *string section*, *brass*, and *reeds*. Additionally, *bass* is now broken up into *bass guitar* and *bass synth*. This is referred to as the Q:ALL setup. Although these are all fine-level stems as defined by MoisesDB, some of these classes are more specific than others. For example, *brass* is a fine-level stem despite possibly including trumpets, trombones, horns, and tuba. The experimental setups are similar to that of Setup B.⁸

The same-song query results⁹ for the models trained in

⁸ BS and DA models for Q:ALL were significantly more unstable during training than for the Q:VDBGP setup, despite being identical architecturally. When this happens, we discard the collapsed model and restart the training from scratch until we have a model that completes the entire training run with nonsilent output for most stems. No TE+DA+BS system was stable enough to finish the training run without collapse.

⁹ Note that when the FE and the TE+DA systems have SNR concentrated at 0 dB for the long-tail stems, these are indicators of the model outputting very soft, practically silent output. In general, a model yielding negative SNR for a particular stem might be more desirable than a model that has collapsed for a particular stem.

Table 4. Model performance on the Q:VDBGP setup fine-level stems.

FE	DA	BS	SSQ	Female Vox			Male Vox			Drums			Bass			Acoust. Gtr.			Clean E. Gtr.			Dist. E. Gtr.			Grand Piano			E. Piano		
				Q1	Q2	Q3	Q1	Q2	Q3	Q1	Q2	Q3	Q1	Q2	Q3	Q1	Q2	Q3	Q1	Q2	Q3	Q1	Q2	Q3	Q1	Q2	Q3	Q1	Q2	Q3
N	N	N	N	5.5	9.6	13.2	6.7	7.9	10.0	8.0	9.6	11.6	7.9	9.9	12.0	0.9	1.8	3.6	0.2	0.7	2.4	0.9	2.4	5.3	0.7	2.3	2.9	0.0	0.6	0.7
			Y	5.6	9.6	13.2	6.7	7.9	10.0	8.0	9.6	11.6	7.9	9.9	12.0	0.9	1.8	3.7	0.2	0.9	2.6	0.9	2.4	5.3	0.7	2.2	3.0	0.0	0.8	1.5
Y	N	N	N	6.1	9.6	13.1	6.8	7.7	9.7	7.8	9.3	11.3	7.6	10.0	11.5	0.8	1.8	3.6	0.2	0.8	2.5	1.0	2.5	5.4	0.8	2.5	3.1	-0.1	0.7	0.8
			Y	6.1	9.6	13.1	6.8	7.7	9.7	7.8	9.3	11.3	7.6	10.0	11.5	0.8	1.8	3.7	0.0	0.9	2.7	1.2	2.5	5.4	0.8	2.5	3.1	-0.6	0.8	1.8
Y	N	N	N	5.5	10.1	13.0	6.9	7.9	10.2	8.5	10.1	12.3	8.4	10.7	13.2	1.2	1.7	4.5	0.2	0.9	3.0	0.9	2.8	4.7	0.8	2.8	3.2	0.1	0.5	0.9
			Y	5.5	10.1	13.1	6.9	7.9	10.2	8.5	10.1	12.3	8.4	10.7	13.2	1.2	1.7	4.6	0.2	1.1	2.7	0.9	2.8	4.7	0.8	2.4	3.1	-0.1	0.6	0.9
Y	N	N	N	5.5	10.1	13.5	6.5	7.8	10.0	8.3	9.5	11.8	8.4	10.3	12.1	1.1	1.7	3.9	0.0	0.4	2.7	0.9	3.0	4.9	0.8	2.6	3.2	0.2	0.5	0.9
			Y	5.5	10.1	13.5	6.5	7.8	10.0	8.3	9.5	11.8	8.4	10.3	12.1	1.0	1.7	3.9	0.3	0.6	2.7	0.6	3.0	4.8	0.8	2.5	3.2	0.6	0.9	2.1

FE: frozen encoder, DA: data augmentation, BS: balanced sampling, SSQ: same-song query, Q1: lower quartile, Q2: median, Q3: upper quartile

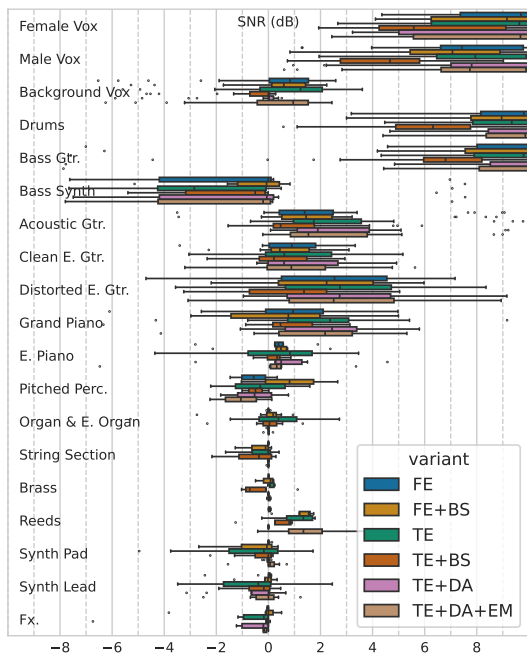


Figure 3. Performance of the Banquet models with same-song queries on Q:ALL fine-level stems

Q:ALL are shown in Fig.3. The performances of the model trained on Q:ALL on the stems from the Q:VDBGP setup are similar to those in Table 4, with the exception of the significant drop in performance for the balanced-sampled trainable-encoder model. Amongst the newly added stems, there are significant variations in performance, but they are all still very weak in terms of SNR, with no sample performing above 5 dB SNR. For organs, background vocals, and both synth stems, the trainable-encoder model yielded the better upper quartile and maximum performance, but is also very unreliable. Unfortunately, balanced sampling on a trainable encoder model only worsened the performance. DA on a trainable-encoder model with default sampling slightly improved the lower quartile performance, but is also accompanied by lower maximum and upper quartile performance. Frozen-encoder system collapsed for most long-tail stems in default sampling, but balanced sampling interestingly was more stable and performed the best for bass synth, pitched percussion, reeds, and brass. Evidently, the classical tradeoffs are at play here; allowing the model more flexibility with a trainable encoder also comes with a higher risk of model collapse or unreliable

performance. More surprisingly, the fact that even a frozen encoder trained on a VDBO setup was able to function at all beyond Q:VDB indicates that the embedding space of a Bandit encoder already contains information that is partially generalizable beyond VDBO, as also observed in [28].

The results of the long-tail stems are somewhat unsurprising given that the genre distribution in MoisesDB skewed heavily toward pop, rock, and singer-songwriter. In addition to the low track counts, these long-tail instruments also tend to have infrequent active segments and relatively softer levels within a song. In fact, of the long-tail stems, reeds and pitched percussion are the only ones with median RMS above -35 dBFS. Analysis of the SNR distribution shows that the model performance is quite correlated to the track-level RMS of the target signal (Spearman’s ρ between 0.78 and 0.81). This is likely due to a combination of low data availability and the inherent difficulty associated with cleanly extracting these “supporting” stems when there are significant spectral overlaps from more prominent co-occurring stems. In light of the recently published analysis in [53], we may have been too conservative with our DA setup. In particular, we made a conscious choice to only perform gain augmentation close to the original levels, instead of significantly amplifying softer stems. Whether the latter may improve the result at all will have to be addressed in future work. Moreover, given that [34] saw partial success with the predominantly classical instrumentation of URMP, there may also be an opportunity for a much more aggressive cross-dataset DA.

6. CONCLUSION

In this work, Banquet, a stem-agnostic single-decoder query-based source separation system was proposed to address MSS beyond the VDBO stems. At 24.9 M trainable parameters, this highly modularized model with a single stream of information flow provided strong performance for vocals, drums, and bass; outperformed significantly more complex HT-Demucs on guitar and piano; and provided a proof-of-concept for extractions of additional long-tail and/or fine-grained stems at no additional complexity. While there remains room for improvements for long-tail stems with low data availability, this work demonstrated the opportunity for further research on single-decoder systems toward supporting a large and diverse set of stems.

7. ACKNOWLEDGMENTS

This work was supported by a Cyber-Infrastructure Resource Award from the Institute for Data Engineering and Science (IDEaS), Georgia Institute of Technology.

K. N. Watcharasupat was separately supported by the American Association of University Women (AAUW) International Fellowship, and the IEEE Signal Processing Society Scholarship Program.

The authors would like to thank Yiwei Ding and Chih-Wei Wu for their assistance with the project.

8. ETHICS STATEMENT

Machine learning-based systems are inherently data-dependent. Our models rely both directly and indirectly on datasets with inherent imbalance, not only in terms of instrument and genre distribution, but also in terms of cultural origins. As a result, our system inevitably inherit some bias and may not perform on music that have not been well represented in the training data. The authors acknowledge this important limitation and are committed to continue exploring approaches to correct these biases, both in terms of data acquisition and algorithmic development.

9. REFERENCES

- [1] A. Liutkus, F.-R. Stöter, Z. Rafii, D. Kitamura, B. Rivet, N. Ito, N. Ono, and J. Fontecave, “The 2016 Signal Separation Evaluation Campaign,” in *Proceedings of the 13th International Conference on Latent Variable Analysis and Signal Separation*. Grenoble, France: Springer International Publishing, 2017, pp. 323–332.
- [2] Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimitakis, and R. Bittner, “The MUSDB18 corpus for music separation,” Dec. 2017.
- [3] —, “MUSDB18-HQ - an uncompressed version of MUSDB18,” Aug. 2019.
- [4] S. Uhlich, M. Porcu, F. Giron, M. Enenkl, T. Kemp, N. Takahashi, and Y. Mitsufuji, “Improving music source separation based on deep neural networks through data augmentation and network blending,” in *Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing*. New Orleans, LA: IEEE, Mar. 2017, pp. 261–265.
- [5] N. Takahashi and Y. Mitsufuji, “Multi-Scale multi-band denisenets for audio source separation,” in *Proceedings of the 2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*. New Paltz, NY: IEEE, Oct. 2017, pp. 21–25.
- [6] F. R. Stöter, A. Liutkus, and N. Ito, “The 2018 Signal Separation Evaluation Campaign,” in *Proceedings of the 14th International Conference on Latent Variable Analysis and Signal Separation*. Guildford, United Kingdom: Springer International Publishing, 2018, pp. 293–305.
- [7] D. Stoller, S. Ewert, and S. Dixon, “Wave-U-Net: A Multi-Scale Neural Network for End-to-End Audio Source Separation,” in *Proceedings of the 19th International Society for Music Information Retrieval Conference*. Paris, France: ISMIR, 2018, pp. 334–340.
- [8] F.-R. Stöter, S. Uhlich, A. Liutkus, and Y. Mitsufuji, “Open-Unmix - A Reference Implementation for Music Source Separation,” *Journal of Open Source Software*, vol. 4, no. 41, p. 1667, Sep. 2019.
- [9] A. Défossez, N. Usunier, L. Bottou, and F. Bach, “Demucs: Deep Extractor for Music Sources with extra unlabeled data remixed,” 2019.
- [10] —, “Music Source Separation in the Waveform Domain,” Nov. 2019.
- [11] R. Hennequin, A. Khlif, F. Voituret, and M. Moussallam, “Spleeter: A fast and efficient music source separation tool with pre-trained models,” *Journal of Open Source Software*, vol. 5, no. 50, p. 2154, Jun. 2020.
- [12] N. Takahashi and Y. Mitsufuji, “D3Net: Densely connected multidilated DenseNet for music source separation,” Mar. 2021.
- [13] Q. Kong, Y. Cao, H. Liu, K. Choi, and Y. Wang, “Decoupling Magnitude and Phase Estimation with Deep ResUNet for Music Source Separation,” in *Proceedings of the 22nd International Society for Music Information Retrieval Conference*, Online, 2021, pp. 342–349.
- [14] A. Défossez, “Hybrid Spectrogram and Waveform Source Separation,” in *Proceedings of the 2021 Music Demixing Workshop at the 22nd International Society for Music Information Retrieval Conference*. Online: ISMIR, Nov. 2021.
- [15] M. Kim, W. Choi, J. Chung, D. Lee, and S. Jung, “KUIELab-MDX-Net: A Two-Stream Neural Network for Music Demixing,” in *Proceedings of the 2021 Music Demixing Workshop at the 22nd International Society for Music Information Retrieval Conference*. Online: ISMIR, Nov. 2021.
- [16] Y. Mitsufuji, G. Fabbro, S. Uhlich, F.-R. Stöter, A. Défossez, M. Kim, W. Choi, C.-Y. Yu, and K.-W. Cheuk, “Music Demixing Challenge 2021,” *Frontiers in Signal Processing*, vol. 1, p. 808395, Jan. 2022.
- [17] Y. Luo and J. Yu, “Music Source Separation With Band-Split RNN,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 1893–1901, 2023.
- [18] S. Rouard, F. Massa, and A. Défossez, “Hybrid Transformers for Music Source Separation,” in *Proceedings of the 2023 IEEE International Conference on Acoustics, Speech and Signal Processing*. Rhodes Island, Greece: IEEE, May 2023.
- [19] W.-T. Lu, J.-C. Wang, Q. Kong, and Y.-N. Hung, “Music Source Separation with Band-Split RoPE Transformer,” in *Proceedings of the 2024 IEEE International Conference on Acoustics, Speech and Signal Processing*. Seoul, Korea, Republic of: IEEE, Sep. 2023, pp. 481–485.
- [20] Y. Wang, D. Stoller, R. M. Bittner, and J. Pablo Bello, “Few-Shot Musical Source Separation,” in *Proceedings of the 2022 IEEE International Conference on Acoustics, Speech and Signal Processing*. Singapore, Singapore: IEEE, May 2022, pp. 121–125.
- [21] R. Bittner, J. Salamon, M. Tierney, M. Mauch, C. Cannam, and J. Bello, “MedleyDB: A Multitrack Dataset for Annotation-Intensive MIR Research,” in *Proceedings of the 15th Conference of the International Society for Music Information Retrieval*. Taipei, Taiwan: ISMIR, 2014, pp. 155–160.
- [22] R. M. Bittner, J. Wilkins, H. Yip, and J. P. Bello, “MedleyDB 2.0 : New Data and a System for Sustainable Data Collection,” in *Extended Abstracts for the Late-Breaking Demo Session of the 17th International Society for Music Information Retrieval Conference*. New York City, USA: ISMIR, 2016.
- [23] E. Manilow, G. Wichern, and J. Le Roux, “Hierarchical Musical Instrument Separation,” in *Proceedings of the 21st International Society for Music Information Retrieval Conference*. Montréal, Canada: ISMIR, 2020, pp. 376–383.

- [24] E. Manilow, G. Wichern, P. Seetharaman, and J. Le Roux, "Cutting Music Source Separation Some Slakh: A Dataset to Study the Impact of Training Data Quality and Quantity," in *Proceedings of the 2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*. New Paltz, NY, USA: IEEE, Oct. 2019, pp. 45–49.
- [25] I. Pereira, F. Araújo, F. Korzeniowski, and R. Vogl, "MoisesDB: A Dataset for Source Separation Beyond 4-Stems," in *Proceedings of the 24th International Society for Music Information Retrieval Conference*, Milan, Italy, 2023, pp. 619–626.
- [26] G. Fabbro, S. Uhlich, C.-H. Lai, W. Choi, M. Martínez-Ramírez, W. Liao, I. Gadelha, G. Ramos, E. Hsu, H. Rodrigues, F.-R. Stöter, A. Défossez, Y. Luo, J. Yu, D. Chakraborty, S. Mohanty, R. Solovyev, A. Stempkovskiy, T. Habruseva, N. Goswami, T. Harada, M. Kim, J. Hyung Lee, Y. Dong, X. Zhang, J. Liu, and Y. Mitsufuji, "The Sound Demixing Challenge 2023 – Music Demixing Track," *Transactions of the International Society for Music Information Retrieval*, vol. 7, no. 1, pp. 63–84, Apr. 2024.
- [27] K. Koutini, J. Schlüter, H. Eghbal-zadeh, and G. Widmer, "Efficient Training of Audio Transformers with Patchout," in *Proceedings of the 23rd Annual Conference of the International Speech Communication Association*. Incheon, Korea: ISCA, Sep. 2022, pp. 2753–2757.
- [28] K. N. Watcharasupat, C.-W. Wu, Y. Ding, I. Orife, A. J. Hipple, P. A. Williams, S. Kramer, A. Lerch, and W. Wolcott, "A Generalized Bandsplit Neural Network for Cinematic Audio Source separation," *IEEE Open Journal of Signal Processing*, vol. 5, pp. 73–81, 2023.
- [29] N. Takahashi, N. Goswami, and Y. Mitsufuji, "MMDenseLSTM: An Efficient Combination of Convolutional and Recurrent Neural Networks for Audio Source Separation," in *Proceedings of the 16th International Workshop on Acoustic Signal Enhancement*. Tokyo, Japan: IEEE, Sep. 2018, pp. 106–110.
- [30] G. Meseguer-Brocal and G. Peeters, "Conditioned-U-Net: Introducing a Control Mechanism in the U-Net for Multiple Source Separations," in *Proceedings of the 20th International Society for Music Information Retrieval Conference*. Delft, Netherlands: ISMIR, Nov. 2019, pp. 159–165.
- [31] E. Perez, F. Strub, H. de Vries, V. Dumoulin, and A. Courville, "FiLM: Visual Reasoning with a General Conditioning Layer," in *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*. New Orleans, LA, USA: AAAI, Dec. 2017.
- [32] O. Slizovskaia, G. Haro, and E. Gómez, "Conditioned Source Separation for Music Instrument Performances," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 2083–2095, 2021.
- [33] B. Li, X. Liu, K. Dinesh, Z. Duan, and G. Sharma, "Creating a Multitrack Classical Music Performance Dataset for Multimodal Music Analysis: Challenges, Insights, and Applications," *IEEE Transactions on Multimedia*, vol. 21, no. 2, pp. 522–535, Feb. 2019.
- [34] L. Lin, Q. Kong, J. Jiang, and G. Xia, "A Unified Model for Zero-shot Music Source Separation, Transcription and Synthesis," in *Proceedings of the 22nd International Society for Music Information Retrieval Conference*. Online: ISMIR, Aug. 2021, pp. 381–388.
- [35] J. H. Lee, H.-S. Choi, and K. Lee, "Audio query-based music source separation," in *Proceedings of the 20th International Society for Music Information Retrieval Conference*. Delft, Netherlands: ISMIR, 2019.
- [36] B. Gfeller, D. Roblek, and M. Tagliasacchi, "One-Shot Conditional Audio Filtering of Arbitrary Sounds," in *Proceedings of the 2021 IEEE International Conference on Acoustics, Speech and Signal Processing*. Toronto, Canada: IEEE, Jun. 2021, pp. 501–505.
- [37] W. Choi, M. Kim, J. Chung, and S. Jung, "LaSAFT: Latent Source Attentive Frequency Transformation For Conditioned Source Separation," in *Proceedings of the 2021 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, Jun. 2021, pp. 171–175.
- [38] Y.-S. Jeong, J. Kim, W. Choi, J. Chung, and S. Jung, "LightSAFT: Lightweight Latent Source Aware Frequency Transform for Source Separation," in *Proceedings of the 2021 Music Demixing Workshop at the 22nd International Society for Music Information Retrieval Conference*. Online: ISMIR, 2021.
- [39] K. Chen, X. Du, B. Zhu, Z. Ma, T. Berg-Kirkpatrick, and S. Dubnov, "Zero-shot Audio Source Separation through Query-based Learning from Weakly-labeled Data," in *Proceedings of the 36th AAAI Conference on Artificial Intelligence*. Online: AAAI, Feb. 2022.
- [40] Q. Kong, K. Chen, H. Liu, X. Du, T. Berg-Kirkpatrick, S. Dubnov, and M. D. Plumbley, "Universal Source Separation with Weakly Labelled Data," May 2023.
- [41] X. Liu, H. Liu, Q. Kong, X. Mei, J. Zhao, Q. Huang, M. D. Plumbley, and W. Wang, "Separate What You Describe: Language-Queried Audio Source Separation," in *Proceedings of the 23rd Annual Conference of the International Speech Communication Association*. Incheon, Korea: ISCA, Sep. 2022, pp. 1801–1805.
- [42] X. Liu, Q. Kong, Y. Zhao, H. Liu, Y. Yuan, Y. Liu, R. Xia, Y. Wang, M. D. Plumbley, and W. Wang, "Separate Anything You Describe," Oct. 2023.
- [43] S. Lee and I. V. Bajic, "Information Flow Through U-Nets," in *Proceedings of the 18th IEEE International Symposium on Biomedical Imaging*. Nice, France: IEEE, Apr. 2021, pp. 812–816.
- [44] P. Seetharaman, G. Wichern, S. Venkataramani, and J. Le Roux, "Class-conditional Embeddings for Music Source Separation," in *Proceedings of the 2019 IEEE International Conference on Acoustics, Speech and Signal Processing*. Brighton, United Kingdom: IEEE, May 2019, pp. 301–305.
- [45] D. Petermann, G. Wichern, A. Subramanian, and J. L. Roux, "Hyperbolic Audio Source Separation," in *Proceedings of the 2023 IEEE International Conference on Acoustics, Speech and Signal Processing*. Rhodes Island, Greece: IEEE, Jun. 2023.
- [46] R. Kumar, Y. Luo, and N. Mesgarani, "Music Source Activity Detection and Separation Using Deep Attractor Network," in *Proceedings of the 19th Annual Conference of the International Speech Communication Association*. Hyderabad, India: ISCA, Sep. 2018, pp. 347–351.
- [47] D. Petermann and M. Kim, "Hyperbolic Distance-Based Speech Separation," in *Proceedings of the 2024 IEEE International Conference on Acoustics, Speech and Signal Processing*. Seoul, Korea, Republic of: IEEE, Apr. 2024, pp. 1191–1195.
- [48] D. Samuel, A. Ganeshan, and J. Naradowsky, "Meta-Learning Extractors for Music Source Separation," in *Proceedings of the 2020 IEEE International Conference on Acoustics, Speech and Signal Processing*. Barcelona, Spain: IEEE, May 2020, pp. 816–820.

- [49] E. J. Humphrey, S. Durand, and B. McFee, "OpenMIC-2018: An open dataset for multiple instrument recognition," in *Proceedings of the 19th International Society for Music Information Retrieval Conference*. Paris, France: ISMIR, 2018, pp. 438–444.
- [50] E. Vincent, R. Gribonval, and C. Févotte, "Performance measurement in blind audio source separation," *IEEE Transactions on Audio Speech and Language Processing*, vol. 14, no. 4, pp. 1462–1469, 2006.
- [51] J. Le Roux, S. Wisdom, H. Erdogan, and J. R. Hershey, "SDR - Half-baked or Well Done?" in *Proceedings of the 2019 IEEE International Conference on Acoustics, Speech and Signal Processing*. Brighton, United Kingdom: IEEE, 2019, pp. 626–630.
- [52] R. Scheibler, "SDR - Medium Rare With Fast Computations," in *Proceedings of the 2022 IEEE International Conference on Acoustics, Speech and Signal Processing*. Singapore, Singapore: IEEE, 2022, pp. 701–705.
- [53] C.-B. Jeon, G. Wichern, F. G. Germain, and J. Le Roux, "Why does music source separation benefit from cacophony?" in *Proceedings of the 1st Workshop on Explainable Machine Learning for Speech and Audio at the 2024 IEEE International Conference on Acoustics, Speech, and Signal Processing*. Seoul, Korea, Republic of: IEEE, Feb. 2024.

IN-DEPTH PERFORMANCE ANALYSIS OF THE ADTOF-BASED ALGORITHM FOR AUTOMATIC DRUM TRANSCRIPTION

Mickaël Zehren
Umeå Universitet
mzehren@cs.umu.se

Marco Alunno
Universidad EAFIT Medellín
malunno@eafit.edu.co

Paolo Bientinesi
Umeå Universitet
pauldj@cs.umu.se

ABSTRACT

The importance of automatic drum transcription lies in the potential to extract useful information from a musical track; however, the low reliability of the models for this task represents a limiting factor. Indeed, even though in the recent literature the quality of the generated transcription has improved thanks to the curation of large training datasets via crowdsourcing, there is still a large margin of improvement for this task to be considered solved. Aiming to steer the development of future models, we identify the most common errors from training and testing on the aforementioned crowdsourced datasets. We perform this study in three steps: First, we detail the quality of the transcription for each class of interest; second, we employ a new metric and a pseudo confusion matrix to quantify different mistakes in the estimations; last, we compute the agreement between different annotators of the same track to estimate the accuracy of the ground-truth. Our findings are twofold: On the one hand, we observe that the previously reported issue that less represented instruments (e.g., toms) are less reliably transcribed is mostly solved now. On the other hand, cymbal instruments have unprecedented relative low performance. We provide intuitive explanations as to why cymbal instruments are difficult to transcribe and we identify that they represent the main source of disagreement among annotators.

1. INTRODUCTION

Automatic Music Transcription (AMT) is a particularly important task in music information retrieval because it provides access to many high-level features of a musical track, such as its structure, melody, and rhythm. A subtask of AMT is automatic drum transcription in the presence of melodic instruments (DTM), which focuses on the estimation of the onsets of drum sounds and the identification of what drum instruments play them. In this article, we focus on DTM and specifically on the transcription of drum and cymbal sounds.

Recently, Zehren et al. presented a DTM algorithm, which we refer to as “ADTOF-based” algorithm, based on supervised learning from abundant crowdsourced annotations [1]. Thanks to the size and diversity of the datasets, the algorithm surpasses the accuracy of the previous state-of-the-art [2]. However, the resulting models are still not perfect, as their estimations contain mistakes. In this work, we carefully investigate these state-of-the-art algorithms, aiming to identify the most common sources of errors.

We evaluated the models in two distinct conditions: (i) when the training and the testing take place on different datasets (out-of-domain), and (ii) when they take place on the same dataset (on-domain). In the first case, the model is not expected to achieve perfect accuracy because of generalization errors that can be attributed to differences between testing and training data. In the second case, testing on-domain, the errors are more concerning as they suggest flaws in the algorithm; in fact, if the dataset were large enough, the model would be expected to learn the data distribution and therefore achieve nearly perfect accuracy. Thus, in this study we focus specifically on the most common errors that arise in the latter case. This was done in three steps, as described in the following.

First, in order to identify the most difficult instruments to transcribe, we independently evaluated the performance of the models on the different instrument classes. When trained and evaluated on (a different split of) the crowdsourced datasets, we observed that the models can reliably transcribe those instruments that play less often, something that in previous studies was arguably problematic to achieve. On the flip side, we also observed that the models do not transcribe cymbals as precisely as drums.

Second, to understand why cymbals are problematic, we employed both a new metric, which we named “octave F-measure”, and a pseudo confusion matrix. Through the new metric, we identified that the models often mistook the beat subdivision at which cymbals are played. Specifically, the rhythm estimated is often half or double the speed of the ground truth (e.g., eighth notes are estimated instead of quarter notes). Through the pseudo confusion matrix, we showed that different kinds of cymbals are hard to discern.

Finally, we assessed how much the quality of crowdsourced annotations affected the evaluated performance of the models. Due to discrepancies in the labels, some of the correct estimations from the models could have been mistakenly reported as errors. To estimate the accuracy of the ground truth itself, we quantified the agreement among

different annotators of the same tracks. Any difference in the annotations of two or more annotators indicates that at least one of them made a mistake; this, in turn, leads to a harsher evaluation of the models than needed.

The remainder of this article is organized as follows: Previous works on the evaluation of DTM algorithms is presented in Sec. 2; in Sec. 3 the transcription accuracy of each class is evaluated, and in Sec. 4 different sources of errors are quantified; finally, the accuracy of the annotations is estimated in Sec. 5, conclusions are drawn in Sec. 6.

2. RELATED WORKS

Automatic drum transcription has evolved from a single, complex task into a series of intermediary steps of increasing difficulty. This evolution facilitated the development of new and more efficient algorithms [3]. Previously, the transcription was limited to simplified audio tracks or constrained vocabulary sizes. However, recent progress made through approaches based on supervised deep learning (DL) has been so significant that it becomes realistic to tackle such a complex task as the non-simplified DTM. The development of DL algorithms focused on two aspects: First, better and more complex architectures are exploited to improve the capabilities of the models, most recently with the introduction of the self-attention mechanism by Vaswani et al. [4] which has been adapted for DTM (e.g., [1, 5]). Second, better training procedures are employed to tune the models, e.g., with the creation of new datasets [1, 2, 6–9].

The de facto method to measure the accuracy of these DTM algorithms, as suggested by the Music Information Retrieval Evaluation eXchange (Mirex) [10], is the F-measure (also known as F1 score). This metric is computed with the harmonic mean between precision and recall of the drum onsets: An onset is considered correct when its estimation is within a small distance from the ground truth. A distance between 20 ms to 50 ms is what is generally used, but it can be tuned depending on the precision of the ground truth [1, 11]. Moreover, the F-measure can be computed at different levels of granularity: from a single class and track to the overall result for a whole dataset. To average multiple tracks and classes, the F-measure can be either computed as the mean value (mean F-measure) or by joining tracks and classes as if they were part of the same file and instrument (sum F-measure). In this study, we rely on the latter because it is more robust to rare edge cases (e.g., a track or class with very few onsets) [12, p.23]. However, since the F-measure gives the same importance to all onsets regardless of their position (i.e., strong or weak beats) or dynamics (loudness), this metric does not necessarily capture the opinion of human listeners [6].

Besides the F-measure, other tools are also used to assess a transcription. For example, Callender et al. used listening tests “where raters compared synthesized transcriptions to original recordings” to estimate the perceived quality of the transcriptions [6]. Vogl et al. relied on confusion matrices adapted to multi-label classification to identify the

errors performed by their model [2]. Ishizuka et al. proposed a “tatum-level error rate based on the Levenshtein distance” [5].

Besides questions related to metric issues, the results of an evaluation are also heavily impacted by the datasets used for testing. There are a handful of datasets suitable for DTM which we group into the following three categories. A thorough description of the datasets is provided by Zehren et al. [13].

- Small but accurate datasets, which have been mostly annotated by hand by their creators, such as RBMA [14], ENST [15], or MDB [16].
- Large but synthetic datasets, synthesized from MIDI files to generate the input audio, such as TMIDT [2].
- Large but inaccurate datasets, which have been annotated by a crowd of people and refined algorithmically, such as ADTOF-RGW [17], ADTOF-YT [1], or A2MD [9].

To choose which datasets to use for testing, we singled out two criteria: First, the characteristics of the datasets (their data distribution) constitute the distribution in which the model is evaluated and should ideally be representative of a real-world situation. For example, testing can be done on different musical genres [1,2], real-world or synthesized audio [5], or different mixtures of instruments (e.g., audio containing four or five sound sources) [18,19]. Second, the test dataset may be part of the training dataset or be a new one, never used during training. The latter is known as an out-of-domain evaluation and, although more challenging, gives a better approximation of the true performances of the model (i.e., its generalization capabilities) [1, 20].

3. CLASS-SPECIFIC RESULTS

In this section, we compare the F-measures for different classes (set of instruments), to identify the most difficult instruments to transcribe for a model when trained in different ways. For this purpose, we selected the “Frame self-att” deep-learning architecture, as it has been recently employed for drum transcription [1], and compared three existing training procedures: 1) training on TMIDT with refinement on ENST, MDB, and RBMA [2]; 2) training on ADTOF-RGW [17]; and 3) training on ADTOF-RGW and ADTOF-YT [1].¹

We evaluated the resulting models on a set of five datasets (RBMA, ENST, MDB, ADTOF-RGW, and ADTOF-YT), to be representative of a real-world situation and to include both on-domain and out-of-domain evaluations. These datasets were carefully mapped to a common vocabulary containing five classes: bass drum (BD), snare drum (SD), toms (TT), open and closed hi-hat (HH), and other cymbals (CY). For the sake of brevity, in Fig. 1 we only present the results of the tests on ENST and ADTOF-YT, as these are representative of the tests on all five datasets.

¹ The models are available at github.com/MZehren/ADTOF

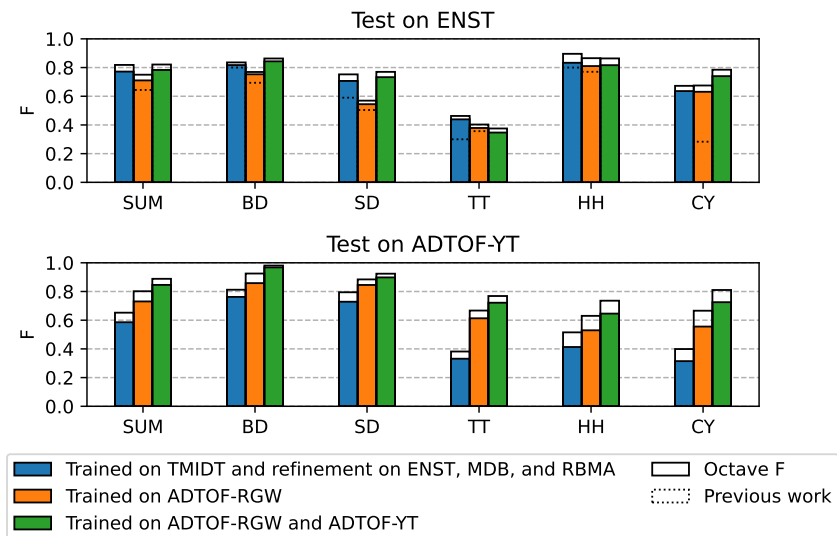


Figure 1: F-measure for the individual classes when testing on ENST (top) and ADTOF-YT (bottom).

First, we analyzed the results on ENST (top) to assess if and to what extent our results are close to those of the original authors of the three existing training procedures. Notably, our reproduction of pre-training on TMIDT with refinement on the non-crowdsourced datasets by Vogl et al. (blue bars) is slightly ahead when compared to their original work (whose results are indicated by the dotted lines inside the blue bars) [2, p.6]. Due to the fact that the evaluation was performed on a vocabulary larger than what we used in our test, a comparison for all instruments was not possible. The little improvement of our model in transcribing the instruments that can be directly compared (BD, SD, TT, and HH) is an indication that we were successful in reproducing the original algorithm. The reproduction of training on only ADTOF-RGW (orange bars) is also slightly better than the results reported [17, p.823] (indicated by the dotted lines inside the orange bars). We attribute this improvement to adopting a more random sampling procedure, something that helped train the models; indeed, compared to the previous work where consecutive (back to back) sequences were drawn between multiple occurrences of the same track, we sampled randomly the datasets (i.e., random track and position, without replacement), thus creating a more homogeneous training. Finally, although the reproduction of training on ADTOF-RGW and ADTOF-YT (green bars) cannot be compared on the class-specific results since they were not previously reported, our model achieved virtually the same sum F-Measure. Namely, when testing on ENST, the model trained on the two ADTOF datasets matches the performance of the model trained on ENST. Thus, ADTOF-based training, as it allows generalization towards ENST, does not overfit models.

Second, we analyzed the results achieved on ADTOF-YT (bottom) to highlight the potential of this dataset. The model achieved a very high F-measure on ADTOF-YT when training on both ADTOF datasets [1] (green bars),

and almost a perfect score for BD, which is surprising considering that this dataset includes the fastest tempi and the densest sequences of onsets, which intuitively are features that hinder transcription. However, we noted that such a high performance is achieved only when ADTOF-YT is part of the training data, which means that the other datasets generalize poorly to it. The fact that only ADTOF-YT attains such a high accuracy both on-domain and out-of-domain may be explained by its large size and the homogeneity of its acoustic and drum patterns (due to the bias toward the metal music genre).

Third, although training and testing on ADTOF-YT yields the highest on-domain performance, we observed that the model has an atypical distribution of performance. In contrast to the usual result where the less represented instruments are less reliably transcribed (e.g., TT when training and testing on ENST), which is due to a lack of training examples, here, the model performs worse on frequently playing instruments. In fact, most of the mistakes of the model concern the transcription of cymbals (HH and CY).

4. ERRORS IN THE ESTIMATIONS

To identify why the transcription of cymbals is prone to mistakes, we quantified the errors made by the model when training and testing on ADTOF-YT. Note that this part of the study is not interested in the generalization capabilities of the model, but in assessing how well it can learn the target data distribution when training on it. We analyze the errors of the model with two tools: First, we approximated the number of errors due to quiet notes, also known as ghost notes, in the dataset with the octave F-measure. Second, we quantified the confusion between the instruments with a pseudo confusion matrix.

4.1 Octave F-measure

We attribute the low performance for cymbals, after conducting a preliminary inspection of the estimations, to both a specific characteristic of their timbre —long sustain that may mask the next onset and the presence of many quiet notes.² Both features make cymbals very challenging to transcribe and their transcription suffers from false negatives and false positives.

As a first step toward solving this problem, we created a new metric meant to quantify how often the presence of quiet notes leads to transcription mistakes. Unfortunately, because ADTOF-YT does not contain reliable velocity information in the annotations, we could not estimate the presence of quiet notes through velocity. Therefore, we started with an assumption from the expert knowledge according to which, in many music genres, cymbals are commonly played in alternation between loud (accentuated) and quiet notes.³ This insight is in agreement with our observation that errors in the estimations are often rhythms that are half or double the speed of the ground truth, so that the algorithm would transcribe a sequence of quarter notes where it should be an eighth note or vice versa. Assuming that this mismatch is due to quiet notes, we created the octave F-measure to allow rhythms that are exactly half or double the speed of the annotations (white bars in Fig. 1). A parallel can be drawn with tempo estimation that uses the “accuracy2” metric which is defined to accept estimations that have a double or triple relationship with ground truth, disregarding ipso facto the so-called octave tempo errors [21]. The octave F-measure gives us an upper bound of the performance of the models if these mistakes were not present in the estimation and the ground truth, and helps us quantify the issues yet to be solved in the algorithms.

When looking at the octave F-measure on ADTOF-YT, we confirm the presence of undetected annotations exactly at the middle point between two estimations, and the presence of extra estimations exactly at the middle point between two annotations. This phenomenon is more common in cymbals than in any other instrument of the drum kit and it is observed in most of the datasets. In other words, the models are mistaking the beat subdivision at which cymbals occur, a problem we attribute to their specific timbre and alternation between loud and quiet notes.

4.2 Confusion Between Classes

To identify typical errors made by the model on ADTOF-YT, we employed the pseudo confusion matrix represented in Fig. 2. Compared to a standard confusion matrix, ours differs in two aspects: First, since in AMT any time position may contain multiple labels—different instruments play simultaneously—the possible sets of labels, instead of each single class, are uniquely listed in the rows and

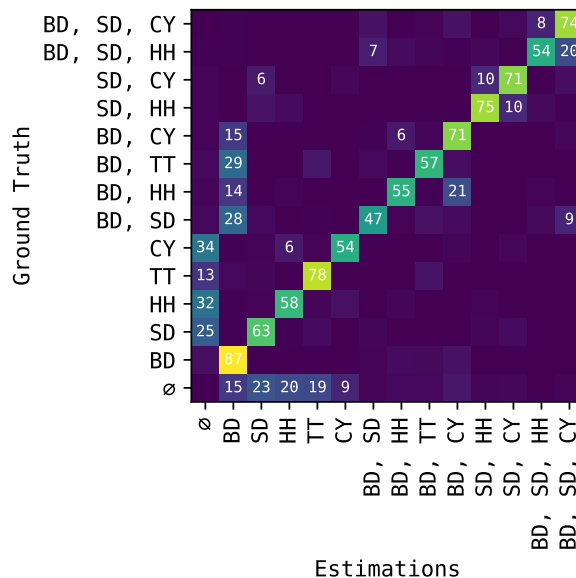


Figure 2: Pseudo confusion matrix on ADTOF-YT. The numbers represent a percentage of the ground truth.

columns of the matrix.⁴ As uniquely identifying sets of labels leads to a large matrix (2⁵ columns and rows with five classes), we truncated the figure for readability to show only the most frequent columns and rows. Second, to remove the imbalance between classes, we normalized the rows (i.e., the rows sum up to 100%). Thus, rather than displaying the count of each set of labels, we represented their proportion relative to the number of occurrences of the ground truth.

We categorize the errors (i.e., the cells outside of the diagonal) in three types following Vogl’s approach [2]: i) confusion, when the onset is detected but the label is wrong (false positives with false negatives); ii) masking, when an onset is missing, presumably because of another one, correctly detected, hides it (false negatives with true positives); iii) excitement where an extra onset is detected, presumably because another one, correctly detected, generates excitement (false positives with true positives). Additionally, another cause of mistakes, which we do not consider in this study, might be related to the low number of occurrences of some combinations of labels (e.g., BD and SD played at the same time), which makes them more difficult to estimate for the model. In Fig. 2, we identified three trends.

First, the left-most column shows that CY, HH, and SD are missing ≈ 30% of the time when they play alone; at the same time, the bottom row highlights that they are incorrectly estimated 10 – 20% of the time when there should be no instrument playing. Surprisingly, this common issue cannot be categorized as due to confusion, masking, or

² SD also contains many quiet notes, but not to the same extent as HH and CY in this dataset.

³ An illustration of this phenomenon can be viewed in the ENST dataset: Notice how every second HH onsets sounds quieter in the example video "Drummer 3, Angle 1" <https://perso.telecom-paristech.fr/grichard/ENST-drums/>

⁴ In practice, since onsets that slightly deviate from the correct position are considered simultaneous, the confusion matrix was created by using an agglomerative clustering that group onsets with a tolerance of 50 ms. As a side effect, we do not count the true negatives (i.e., positions without onsets: ∅ in the ground truth and estimation).

excitement. Instead, we attribute these two phenomena to the presence of quiet notes for those three instruments, as already commented in Section 4.1.

Second, when looking at the intersection of the rows containing HH with the equivalent columns containing CY, such as the cell in the row “BD, HH” and column “BD, CY”, we notice that HH is often confused with CY. Similarly for the intersection of the rows containing CY with the equivalent columns containing HH, we observe that CY is often confused with HH. Again, this highlights that similar-sounding instruments are misinterpreted, as already identified by Vogl [2]. However, looking specifically at the rows “CY” and “HH”, we notice that CY and HH are less often confused with each other when they occur in isolation. Thus, we conclude that confusion is exacerbated by the presence of other instruments, likely because of their masking effects.

Lastly, the second column highlights that TT and SD are both missed $\approx 30\%$ of the time when they appear with BD. Presumably because BD’s wide spectral range masks the other instruments’ spectra, this illustrates that masking seems to be very prevalent on ADTOF-YT. Excitation, on the other hand, is not a common issue compared to masking or confusion. We only notice the presence of an extra CY onset 9% of the times that “BD, SD” occurs.

In addition to the Octave F-measure that illustrates how the model misjudges the beat subdivision at which cymbals are played, the confusion matrix shows that the model does not differentiate well the cymbals. However, one might wonder why these issues are prevalent on ADTOF-YT and not the other datasets.

5. ANNOTATIONS ACCURACY

To understand why the model is prone to make mistakes specifically with ADTOF-YT, we took a closer look at the accuracy of its annotations. Both the datasets ADTOF-RGW and ADTOF-YT are crowdsourced; since human annotations are not perfect and annotators do not always agree with each other, we expect mistakes in the datasets. While a cleansing/cleaning procedure was employed to improve the time position of the annotations and to remove label ambiguity [1, p.784], it is not realistic to expect that all mistakes will be corrected. Although DL is generally robust to label noise [22], incorrect labeling might affect the models during training and testing, especially with crowdsourced datasets that likely contain more mistakes than non-crowdsourced ones. Specifically during testing, any error in the annotations is indistinguishable from wrong estimations of the models and impacts their evaluation. Therefore, by assessing the accuracy of the annotations, it is possible to estimate an upper bound of the performance of the models tested on the dataset. As this bound corresponds to the score achieved by a perfect classifier, it can show how far the current models are from this ideal.

To estimate the annotations’ accuracy on a dataset and create a ground truth of high confidence, it is common to compare labels provided by independent annotators on the same data and measure the confidence of the annotations,

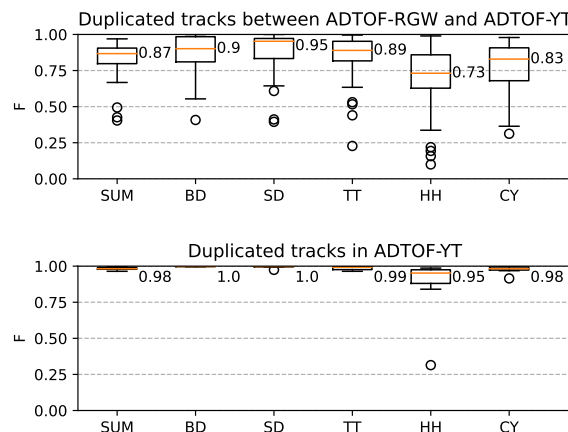


Figure 3: Box plots representing the distribution of the F-measure on tracks present both in ADTOF-RGW and ADTOF-YT (top) and on duplicated tracks in ADTOF-YT (bottom).

for example by grouping multiple independent annotations into a single set (e.g., [23, p.7], [24], [25, p.255]). Further, by comparing this ground truth with a new (group of) annotator(s), one can estimate either the ground truth accuracy or the human-level accuracy depending on how much one trusts the reference group (e.g., [23, p.7 and 31], [26]). In our context, similarly to what Flexer and Grill [27] did in their work, we aim to estimate an upper limit of the score achievable on the datasets by assessing the agreement among human annotators.

To do so, we rely on the tracks that appear multiple times and are annotated by different persons in the datasets. After aligning two instances of the same track according to their annotations, we were able to compute the agreement between the annotators the same way we evaluate any algorithmic estimation: By taking either set of annotations as the reference and the other one as the estimation, we can then compute the F-measure. Note that the results do not depend on which annotator is used as the reference: By switching the annotator used as the reference, precision and recall are also switched, without impacting the F-measure. The distribution of the F-measure for all the tracks found both in ADTOF-RGW and ADTOF-YT (34 couples, 4h28min) and duplicated in ADTOF-YT (7 couples, 34min) is shown in Fig. 3. There are no duplicated tracks within ADTOF-RGW.

On the one hand, the annotations in duplicated tracks of ADTOF-YT are almost identical, whereas they differ between ADTOF-RGW and ADTOF-YT. This is an indication that the annotators of ADTOF-YT agree more often with themselves than with the annotators of ADTOF-RGW. In turn, this is a sign that the annotations of ADTOF-YT are very accurate. If that is the case, then the models we evaluated on ADTOF-YT are far from perfect, as they do not achieve results close to the inter-rater agreement. However, we acknowledge that this trend is only supported by seven couples of tracks and further investigation is required to claim that this dataset contains so few errors.

On the other hand, the median agreement between annotators of ADTOF-RGW and ADTOF-YT is very similar to the best model’s sum F measure on ADOTF-YT (0.87 for the annotators Fig. 3 compared to 0.85 for the model Fig. 1). This suggests that the model performs as well as the annotators of ADTOF-RGW on ADTOF-YT. Moreover, this trend holds for the majority of the classes. Most notably, both the annotators and the model manifest difficulties with the discrimination between HH and CY (lowest agreement and performance). Similarly to the models in the previous sections, we attribute these human errors both to the fact that one instrument is mistaken for the other because of their similar timbre, and to the use of different rhythms because of the presence of quiet notes. See Fig. 4 (top) showing the disagreement between two annotators as an illustration of both phenomena. Although it is not clear if these discrepancies are part of ADTOF-RGW, ADTOF-YT, or both, they impact negatively the measure of the model performance. However, we noticed that the agreement between annotators is much lower than the performance of the model on BD (0.90 for the annotators compared to 0.97 for the model). This is due to the presence of simplified annotations in ADTOF-RGW. As represented in Fig. 4 (bottom), these simplifications are meant to ease the gameplay when a double bass drum technique is required (i.e., bass drum notes played with both feet) by omitting the notes played by the left foot. Despite such simplifications, the model still manages to achieve a high F-measure when testing on ADTOF-YT, which does not contain simplified annotations.

Although data is not enough to determine accurately an upper limit to the performance of the models, we believe that the agreement we measured among annotators is a reasonably good guess. Because it is not possible to know which of the annotators made a mistake (possibly both), the discrepancies between them do not always impact the measure of the model’s performance, making this estimation pessimistic.⁵ However, considering that the best performance we achieve on ADTOF-YT is close to the agreement among humans, it is intuitive that any improvement of the model beyond this point will not be easily measurable. In other words, this model is not far from a perfect classifier on this dataset.

6. CONCLUSIONS

In this work, we analyzed the performance of a state-of-the-art model for automatic drum transcription [1]. First, through the F-measure for the individual classes, we identified that ADTOF-YT is the only dataset able to train a model to such a high level of accuracy on its data distribution. In this context, when training and testing on ADTOF-YT, the transcription is: almost perfect for the bass drum (BD), better than previous methods for the sparse class of tom-toms (TT), but less reliable for cymbals. Second, to understand why cymbals are more difficult to transcribe,

⁵ In the hypothetical scenario where each disagreement is caused by only one of the annotator making a mistake, the discrepancies will affect the evaluation only 50% of the cases.

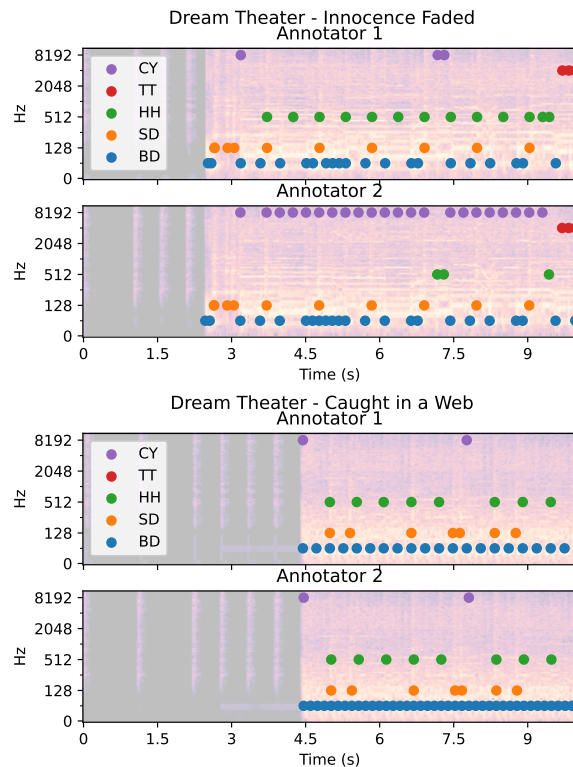


Figure 4: Spectrograms and annotations from ADTOF-RGW and ADTOF-YT for the first 10s of two tracks. Notice the confusion and the use of different subdivisions between CY and HH (top), as well as the simplification of the BD for fast rhythms (bottom).

we used a new metric we named Octave F-measure as well as a pseudo confusion matrix. We then concluded that what hinders the cymbals’ transcription is their typical accentuated-note/quiet-note pattern and their similar timbre to each other. Last, because the test data has been annotated by many people with different levels of expertise, we aimed to quantify the errors due to discrepancies in the ground truth rather than to mistakes made by the model. By estimating the accuracy of the annotations through the agreement between multiple annotators of the same tracks, we identified that the human-level accuracy is on par with the performance of the model. Thus, it is not clear whether the differences between the estimations and annotations originate from the model or the annotators, even though their causes are the same.

With this study, we quantified the main difficulties faced by the model or the annotators. The errors caused by the cymbals could be the focus of future research in ADT, which we believe could be tackled in one of two ways: Either existing annotations could be verified, possibly via a semi-automatic method relying on the estimation of a pre-trained model to detect likely errors, or complementary training data could be generated, possibly in a synthetic way, to ensure a perfect ground truth.

7. ACKNOWLEDGMENTS

The computations were enabled by resources provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS), partially funded by the Swedish Research Council through grant agreement no. 2022-06725.

8. REFERENCES

- [1] M. Zehren, M. Alunno, and P. Bientinesi, “High-Quality and Reproducible Automatic Drum Transcription from Crowdsourced Data,” *Signals*, vol. 4, no. 4, pp. 768–787, Nov. 2023. [Online]. Available: <https://www.mdpi.com/2624-6120/4/4/42>
- [2] R. Vogl, G. Widmer, and P. Knees, “Towards multi-instrument drum transcription,” in *21th International Conference on Digital Audio Effects (DAFx-18)*, Jun. 2018. [Online]. Available: <http://arxiv.org/abs/1806.06676>
- [3] C.-W. Wu, C. Dittmar, C. Southall, R. Vogl, G. Widmer, J. Hockman, M. Muller, and A. Lerch, “A Review of Automatic Drum Transcription,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 9, pp. 1457–1483, Sep. 2018. [Online]. Available: <https://ieeexplore.ieee.org/document/8350302/>
- [4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention Is All You Need,” *arXiv:1706.03762 [cs]*, Dec. 2017, arXiv: 1706.03762. [Online]. Available: <http://arxiv.org/abs/1706.03762>
- [5] R. Ishizuka, R. Nishikimi, and K. Yoshii, “Global Structure-Aware Drum Transcription Based on Self-Attention Mechanisms,” *Signals*, vol. 2, no. 3, pp. 508–526, 2021. [Online]. Available: <https://www.mdpi.com/2624-6120/2/3/31>
- [6] L. Callender, C. Hawthorne, and J. Engel, “Improving Perceptual Quality of Drum Transcription with the Expanded Groove MIDI Dataset,” *arXiv:2004.00188 [cs]*, May 2020, arXiv: 2004.00188. [Online]. Available: <http://arxiv.org/abs/2004.00188>
- [7] M. Cartwright and J. P. Bello, “Increasing Drum Transcription Vocabulary Using Data Synthesis,” in *21th International Conference on Digital Audio Effects (DAFx-18)*, 2018, pp. 72–79.
- [8] E. Manilow, G. Wichern, P. Seetharaman, and J. Le Roux, “Cutting music source separation some Slakh: A dataset to study the impact of training data quality and quantity,” in *IEEE workshop on applications of signal processing to audio and acoustics (WASPAA)*. IEEE, 2019.
- [9] I.-C. Wei, C.-W. Wu, and L. Su, “Improving Automatic Drum Transcription Using Large-Scale Audio-to-Midi Aligned Data,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Toronto, ON, Canada: IEEE, Jun. 2021, pp. 246–250. [Online]. Available: <https://ieeexplore.ieee.org/document/9414409/>
- [10] “MIREX 2021: Drum Transcription,” 2021. [Online]. Available: https://www.music-ir.org/mirex/wiki/2021:Drum_Transcription#Evaluation
- [11] J. Gardner, I. Simon, E. Manilow, C. Hawthorne, and J. Engel, “MT3: Multi-Task Multitrack Music Transcription,” in *International Conference on Learning Representations*, 2021, p. 21, arXiv: 2111.03017. [Online]. Available: <https://openreview.net/pdf?id=iMSjopcOn0p>
- [12] R. Vogl, “Deep Learning Methods for Drum Transcription and Drum Pattern Generation,” Ph.D. dissertation, Institute of Computational Perception, Linz, Austria, Nov. 2018.
- [13] M. Zehren, M. Alunno, and P. Bientinesi, “Analyzing and reducing the synthetic-to-real transfer gap in Music Information Retrieval: the task of automatic drum transcription,” Jul. 2024, arXiv:2407.19823 [cs, eess]. [Online]. Available: <http://arxiv.org/abs/2407.19823>
- [14] R. Vogl, M. Dorfer, G. Widmer, and P. Knees, “Drum transcription via joint beat and drum modeling using convolutional recurrent neural networks,” in *18th International Society for Music Information Retrieval Conference*, Suzhou, China, 2017, pp. 150–157.
- [15] O. Gillet and G. Richard, “ENST-Drums: an extensive audio-visual database for drum signals processing,” in *7th International Society for Music Information Retrieval Conference (ISMIR)*, Victoria, Canada, 2006, pp. 156–159.
- [16] C. Southall, C.-W. Wu, A. Lerch, and J. Hockman, “MDB drums- An annotated subset of MedleyDB for Automatic Drum Transcription,” Suzhou, China, 2017.
- [17] M. Zehren, M. Alunno, and P. Bientinesi, “ADTOF: A large dataset of non-synthetic music for automatic drum transcription,” in *22nd International Society for Music Information Retrieval Conference (ISMIR)*, Online, 2021, pp. 818–824.
- [18] E. Manilow, P. Seetharaman, and B. Pardo, “Simultaneous Separation and Transcription of Mixtures with Multiple Polyphonic and Percussive Instruments,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Barcelona, Spain: IEEE, May 2020, pp. 771–775. [Online]. Available: <https://ieeexplore.ieee.org/document/9054340/>
- [19] Y. Wang, J. Salamon, M. Cartwright, N. J. Bryan, and J. P. Bello, “Few-Shot Drum Transcription in Polyphonic Music,” in *21st International Society for Music Information Retrieval Conference (ISMIR)*, Montréal, 2020, pp. 117–124.

- [20] D. Hernandez, J. Kaplan, T. Henighan, and S. McCandlish, “Scaling Laws for Transfer,” Feb. 2021, arXiv:2102.01293 [cs]. [Online]. Available: <http://arxiv.org/abs/2102.01293>
- [21] H. Schreiber, J. Urbano, and M. Müller, “Music Tempo Estimation: Are We Done Yet?” *Transactions of the International Society for Music Information Retrieval*, vol. 3, no. 1, p. 111, Aug. 2020. [Online]. Available: <https://transactions.ismir.net/article/10.5334/tismir.43/>
- [22] D. Rolnick, A. Veit, S. Belongie, and N. Shavit, “Deep Learning is Robust to Massive Label Noise,” Feb. 2018, arXiv:1705.10694 [cs]. [Online]. Available: <http://arxiv.org/abs/1705.10694>
- [23] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, “ImageNet Large Scale Visual Recognition Challenge,” Jan. 2015, arXiv:1409.0575 [cs]. [Online]. Available: <http://arxiv.org/abs/1409.0575>
- [24] M. Zehren, M. Alunno, and P. Bientinesi, “M-DJCUE: A Manually Annotated Dataset of Cue Points,” in *Late Breaking/Demo at the 20th International Society for Music Information Retrieval*, Delft, The Netherlands, 2019, p. 2.
- [25] O. Nieto, G. J. Mysore, C.-i. Wang, J. B. L. Smith, J. Schlüter, T. Grill, and B. McFee, “Audio-Based Music Structure Analysis: Current Trends, Open Challenges, and Applications,” *Transactions of the International Society for Music Information Retrieval*, vol. 3, no. 1, pp. 246–263, Dec. 2020. [Online]. Available: <http://transactions.ismir.net/articles/10.5334/tismir.54/>
- [26] W. Xiong, L. Wu, F. Alleva, J. Droppo, X. Huang, and A. Stolcke, “The Microsoft 2017 Conversational Speech Recognition System,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2018, pp. 5934–5938, arXiv:1708.06073 [cs]. [Online]. Available: <http://arxiv.org/abs/1708.06073>
- [27] A. Flexer and T. Grill, “The Problem of Limited Inter-rater Agreement in Modelling Music Similarity,” *Journal of New Music Research*, vol. 45, no. 3, pp. 239–251, Jul. 2016. [Online]. Available: <https://www.tandfonline.com/doi/full/10.1080/09298215.2016.1200631>

TOWARDS MUSICALLY INFORMED EVALUATION OF PIANO TRANSCRIPTION MODELS

Patricia Hu¹ Lukáš Samuel Marták^{1,2} Carlos Cancino-Chacón¹ Gerhard Widmer^{1,2}

¹ Institute of Computational Perception, Johannes Kepler University Linz, Austria

² LIT AI Lab, Linz Institute of Technology, Austria

patricia.hu@jku.at

ABSTRACT

Automatic piano transcription models are typically evaluated using simple frame- or note-wise information retrieval (IR) metrics. Such benchmark metrics do not provide insights into the transcription quality of specific musical aspects such as articulation, dynamics, or rhythmic precision of the output, which are essential in the context of expressive performance analysis. Furthermore, in recent years, MAESTRO has become the de-facto training and evaluation dataset for such models. However, inference performance has been observed to deteriorate substantially when applied on out-of-distribution data, thereby questioning the suitability and reliability of transcribed outputs from such models for specific MIR tasks. In this work, we investigate the performance of three state-of-the-art piano transcription models in two experiments. In the first one, we propose a variety of musically informed evaluation metrics which, in contrast to the IR metrics, offer more detailed insight into the musical quality of the transcriptions. In the second experiment, we compare inference performance on real-world and perturbed audio recordings, and highlight musical dimensions which our metrics can help explain. Our experimental results highlight the weaknesses of existing piano transcription metrics and contribute to a more musically sound error analysis of transcription outputs.

1. INTRODUCTION

Automatic Music Transcription (AMT) refers to the task of converting audio signals into symbolic music representations. The target output format can be a full symbolic score including quantized rhythm, time signature and pitch spelling information, or a mid-level physical MIDI(-like) representation, describing notes in terms of their onset and offset times, pitch and velocity [1–3].

AMT methods are typically evaluated using information retrieval (IR) metrics like precision, recall and F1 score [4]. These IR metrics can be computed at the level of frames, by comparing binary piano roll-like matrices, or at

the level of note lists, by comparing notes in terms of their onset, offset, pitch and/or velocity attributes. Each error (i.e. misplaced frame or note activity) has equal weight, resulting in limited explanatory power of these metrics with respect to the underlying musical material [1, 5].

As in many other areas in MIR, the current state of the art is defined by deep neural networks [2, 3, 6, 7]. To a large extent, this progress has been enabled by the release of the MAESTRO dataset [8], which made well-aligned audio-MIDI piano performance data available on a large scale. The most up-to-date version of the MAESTRO dataset¹ contains close to 200 hours of performance data from close to 1300 recordings of Western classical piano repertoire. State-of-the-art piano transcription systems achieve beyond 90% frame-level, or 80% note-level F1 scores on its test split [2, 3, 8, 9], and have led to the release of large-scale transcribed solo piano performance datasets [10, 11].

Although these results are impressive, we believe that two important aspects have been largely overlooked: first, the validity and (lack of) explanatory power of the standard evaluation metrics with respect to musically relevant information, and second, the reliability of these transcription models on out-of-distribution data. In this work, we address the first problem by proposing a set of musically informed evaluation metrics that support a more nuanced understanding of piano transcription errors. The metrics are intended to be used in the context of computational performance studies, and therefore focus on musical dimensions that are commonly studied in the context of expressive piano performance analysis and generation. We demonstrate our metrics on a subset of the MAESTRO dataset, which we transcribe using three state-of-the-art transcription models. In particular, we contrast the performance of these models, as evaluated with the standard IR metrics, with their performance on musical dimensions such as timing, articulation and dynamics which we can evaluate using our set of musically informed metrics.

Then, to elucidate the second problem, we re-record a subset of the MAESTRO dataset on a Yamaha Disklavier grand piano and further manipulate the audio recordings by adding different levels of noise and reverberation. An analysis of the outputs of these trained transcription models on these recordings provides some detailed insights into the lack of generalization on out-of-distribution data.

¹ <https://magenta.tensorflow.org/datasets/maestro>

We make our set of metrics, data and all experimental results available at <https://github.com/CPJKU/mp-teval>.

2. RELATED WORK

This section briefly reviews the standard IR evaluation metrics along with criticism related to these, followed by a description of the benchmark datasets typically used for evaluating transcription methods.

Precision, recall and F1 score are the standard evaluation metrics used in AMT [1–4]. They can be computed either at the level of frames or at the level of notes. For frame-level evaluation, two binary piano roll matrices $M, \hat{M} \in \{0, 1\}^{P \times T}$ are compared, where $p = 1, \dots, P$ defines the pitch range and $t = 1, \dots, T$ the time step (typically with a resolution of 10ms [4]). Both M and \hat{M} are sparse matrices where 1 at a given index p, t indicates that a note with pitch p is active at time frame t .

Note-level metrics are computed by comparing lists of notes, in which each note is described by a tuple describing the onset, offset, pitch, and (where predicted) velocity. Note-based metrics can be based on onset information only, onset and offset information (i.e., note durations), or on predicted onset, offset and velocity. In onset-only note evaluation, a note is considered correct if its onset falls within a ± 50 ms threshold of its respective target onset. For onset-and-offset note-level evaluation, the note offset must fall within the greater of either an offset tolerance threshold of ± 50 ms, or a duration threshold 20% of the ground truth duration [4]. If velocity is included in the evaluation, an estimated note is considered correct if its velocity (after some normalization and rescaling operations) falls within a 0.1 tolerance threshold of the velocity of the corresponding reference note [1].

The need for better (i.e., musically or perceptually sound) transcription metrics has been expressed by various researchers before. Hawthorne et al. [1] point out that frame- and onset-only note-level evaluation does not sufficiently capture musically relevant information. Similarly, Ycart et al. [5] and Daniel et al. [12] focus on the problem of perceptual saliency of different kinds of transcription errors and each propose a new, perceptually (more) valid transcription metric. Finally, McLeod and Steedman [13] focus on the problem of audio-to-score transcription and propose a new metric that jointly evaluates voice separation, metrical alignment, note value detection and harmonic analysis along with multi-pitch detection.

With respect to training and evaluation data for solo piano transcription, until the introduction of MAESTRO [8], MAPS [14] was used as the standard dataset. Apart from size, the biggest difference between the two is the diversity of the captured recording environments: while MAESTRO exclusively contains Disklavier recordings from the Yamaha International Piano e-Competition², MAPS contains Disklavier recordings and synthesized audio simulating various recording environments. The prevailing trend

in evaluating current piano transcription models centers around the MAESTRO dataset [2, 3, 6], and most models that do include MAPS in their evaluation [1, 3] use the split proposed in [15], which only includes Disklavier recordings in the test split. Both frame- and note-level metrics are usually computed for each piece in a given test set, and their mean is subsequently reported as the inference performance for a given model and dataset/split. Frame-level metrics are typically higher than note-level ones due to common known transcription errors such as merged or segmented notes.

3. MUSICALLY INFORMED METRICS

In this section we describe our proposed metrics that are meant to capture different musical dimensions commonly studied in the context of expressive performance. Each metric compares a ground truth to a predicted MIDI performance by measuring the Pearson correlation between a performance parameter computed from the ground truth and from the predicted MIDI, respectively. We choose a correlational measure to ensure all metrics fall into the same range. The goal is to quantify dimensions of musical quality of transcriptions that are otherwise obscured by standard IR metrics. In particular, we wish to capture dimensions that are important for computational performance studies that make use of automatically transcribed piano performances.

3.1 Timing

Timing can be described as expressive deviations from the metrical grid. A common measure of expressive timing in computational performance analysis is the inter-onset-interval (IOI), that is, the amount of time passed between two consecutive notes belonging to the same stream.³

To evaluate how well a transcription preserves the micro onset deviations, we predict a monophonic melody line and the accompaniment part (i.e., all notes not belonging to the melody line) in a given MIDI using the skyline algorithm for melody identification [17].⁴

Then we compute the IOIs of these streams both on the ground truth and predicted performance, and measure their correlation. Note that for non-strictly monophonic streams (like the accompaniment part), the IOI between notes that belong to the same onset (i.e., chords) is zero. The result of this process gives us two measures, which we call *Melody IOI* and *Accompaniment IOI*.

3.2 Articulation

Articulation in expressive piano performance refers to how (adjacent) notes are played in terms of their duration, intensity, and clarity, resulting in expressive strategies such

³ We use the term *stream* as a generalization of the concept of a voice in polyphonic music [16].

⁴ The skyline algorithm has been shown to be very competitive in identifying melody lines in Western classical piano music, even when compared to more recent machine learning-based algorithms. (e.g., see Figure 5 in [18]).

² <http://piano-e-competition.com/>

as *legato*, *staccato* or *marcato*. Computationally, articulation is measured as the ratio between the time interval from the offset of the current note to the onset of the next note, and the time between the onsets of the two notes. [19–21].

We use the skyline algorithm [17] to extract monophonic melody and bass lines within a performance, and compute a sequence of KOR values, for each pair of successive note events, for both the target and the predicted performance MIDI for both streams, and their ratio. We define three metrics for capturing articulation:

1. *Melody KOR*: the correlation between the KOR sequences of the melody lines of the ground truth and the predicted performance MIDI.
2. *Bass KOR*: the correlation between the KOR sequences of the bass lines of the ground truth and the predicted performance MIDI.
3. *Ratio KOR*: for this metric, we consider the ratio of KOR sequences of the melody to the bass line. A ratio KOR greater than 1 indicates that the melody voice is played more legato than the bass voice. The Ratio KOR metric is computed as correlation of this ratio between the ground truth and the predicted performance MIDI.

3.3 Harmony

Aspects such as harmonic tension have been shown to be determining factors for various performance decisions (particularly relating to expressive tempo and dynamics [22, 23]). To quantify how well harmonic tension is preserved in a transcription, we use two features proposed by Herremans and Chew [24] based on Chew’s spiral array model [25]. This model is a three dimensional representation of pitch classes, chords and keys constructed in such a way that spatial proximity represents close tonal relationships.⁵ We use two metrics to capture the preservation of harmonic tension:

1. *Cloud Diameter*: this metric measures the maximal tonal distance as the maximum dispersion between notes in a musical segment
2. *Cloud Momentum*: this metric captures the harmonic movement in a segment as the tonal distance between consecutive sections.

For both metrics, we compute the respective feature on overlapping windows for both the ground truth and transcribed MIDI, and measure their correlation.

3.4 Dynamics

For comparing the performance of transcription models regarding expressive dynamics, we use the loudness ratio of the melody and bass lines as a proxy to identify how well a transcription preserves the dynamics of the performance.

⁵ We chose this model for its simplicity and music-theoretical grounding. Note that these features were designed for Western tonal music and may be less effective in capturing tension in other types of music.

composer	pieces	performances	duration (min)
Bach	1	7	23.36
Beethoven	5	28	285.54
Chopin	4	15	150.28
Debussy	2	3	32.06
Glinka	1	2	10.35
Haydn	3	9	90.23
Liszt	3	12	58.98
Mozart	2	4	29.02
Rachmaninoff	2	3	11.87
Schubert	3	17	107.27
Scriabin	1	5	55.05
Total	27	105	854.01

Table 1: Overview of chosen composers, pieces, and performances in the MAESTRO subset in our evaluation set.

We estimate the loudness as the “energy” of a stream (i.e., melody or bass line), which is computed using the MIDI velocity following a model proposed by Dannenberg [26]. The loudness ratio is then computed as follows (cf. Equation 8 in [26]):

$$R(t) = \log \left(\frac{m \cdot \text{vel}_{mel}(t) + b}{m \cdot \text{vel}_{bass}(t) + b} \right) \quad (1)$$

where $\text{vel}_{mel}(t)$ and $\text{vel}_{bass}(t)$ are the MIDI velocities of the melody and bass lines at time t , respectively, and m and b are constant parameters that depend on the dynamic range of the audio signal. We compute the loudness ratio for both the ground truth and estimated performance MIDI, and compute the correlation between these ratios as our metric for dynamics.⁶

4. DATA

For our experiments, we create an evaluation set with three subsets:

1. *MAESTRO*: We select audio recordings from the MAESTRO dataset, covering a diverse range of musical repertoire, composers, and performers, using all (train, validation, and test) splits as provided by the authors [3]. This choice tests whether the split category affects model generalization.⁷ An overview of the selected subset is shown in Table 1.
2. *Disklavier*: We re-record our MAESTRO subset on a Yamaha Disklavier Enspire ST C1X using the Focusrite Scarlett 18i8 and a pair of AKG P420 microphones in a moderately bright, fully carpeted room with asymmetric geometry and low background noise level.
3. *revnoise*: To simulate more challenging real-world environments, we further add perturbations using

⁶ Dannenberg’s model was chosen for its simplicity, relying only on MIDI velocity and dynamic range (note that parameters m and b cancel each other out when computing the correlation of the loudness ratio).

⁷ The official MAESTRO splits [8] ensure a unique piece-to-split mapping.

different levels of reverberation and noise (see Section 6) on selected recordings from both the *MAESTRO* and *Disklavier* subsets.

We compare three state-of-the-art piano transcription models: Onsets and Frames [8] and the Transformer transcription model [3] by Google/Magenta (which we will refer to as OaF and T5 respectively, in the following), and the high-resolution onset and offset regression model by Bytedance [2] (referred to as Kong).

We transcribe all the recordings in our MAESTRO, *Disklavier*, and *revnoise* subsets using the (officially provided) trained models, and these transcriptions then form our evaluation set. Note that all audio recordings from the *Disklavier* and *revnoise* subsets are only used for testing; we use the MAESTRO-trained models as they are provided by the respective authors via their repositories.

5. DEMONSTRATION OF MUSICALLY INFORMED METRICS

We now discuss the experimental results obtained with the three systems and explain the relation between our metrics and the standard IR metrics as computed on transcriptions of the MAESTRO subset of our evaluation set. We focus in particular on the musical dimensions that can be better understood through our metrics. For the standard evaluation, we include the frame-level score, and all note-level F1 scores other than the onset-only one as it does not capture offset and velocity information. We compute the note-level metrics with the official `mir_eval` python implementation [27].

We start our discussion with Table 2, which summarizes the evaluation results per model and metric. For comparative reasons, we also include a perceptually informed piano transcription metric, PEAMT [5].

Generally, it can be observed that the Kong model performs the best across most metrics. This implies that most of our metrics, overall, correlate with the performance ranking as measured on the standard metrics. Furthermore,

Metric	OaF	Kong	T5
Frame F1	0.8710	0.9138	0.7048
Note Offset F1	0.6167	0.8736	0.6358
Note Offset Velocity F1	0.5917	0.8587	0.6309
Melody IOI	0.2377	0.5481	0.2217
Accompaniment IOI	0.2168	0.3679	0.4329
Melody KOR	0.4057	0.7415	0.2825
Bass KOR	0.2638	0.6967	0.2672
Ratio KOR	0.4247	0.6938	0.3094
Cloud Diameter	0.7240	0.8301	0.7472
Cloud Momentum	0.2461	0.2250	0.1671
Dynamics	0.5501	0.6503	0.6355
PEAMT [5]	0.6570	0.6241	0.5789

Table 2: Model performance measured by standard metrics, our musically informed metrics, and PEAMT [5] on the MAESTRO subset of our evaluation set

it can be seen that the two Magenta models perform considerably different when measured against frame-level F1 score, yet this difference becomes less pronounced when evaluated on note-level metrics, which would suggest superior performance of the T5 model. Comparing these results to the model performance as evaluated on our set of metrics, however, reveals that while both models perform similarly on onset time prediction, the T5 model is worse at adequately capturing note durations, particularly in lower voices/frequency ranges, but superior in estimating MIDI velocity and the overall loudness ratio between voices than the OaF model. Lastly, we can observe that the perceptually informed PEAMT metric correlates most with the frame-level and harmony metric *Cloud Momentum*, which might suggest (if PEAMT is indeed a veridical listening model) that listeners place relatively high importance on harmonic context.

We continue our discussion in Figure 1, which compares the note-offset F1 score per composer (averaged over pieces and performers) and model to our musically in-

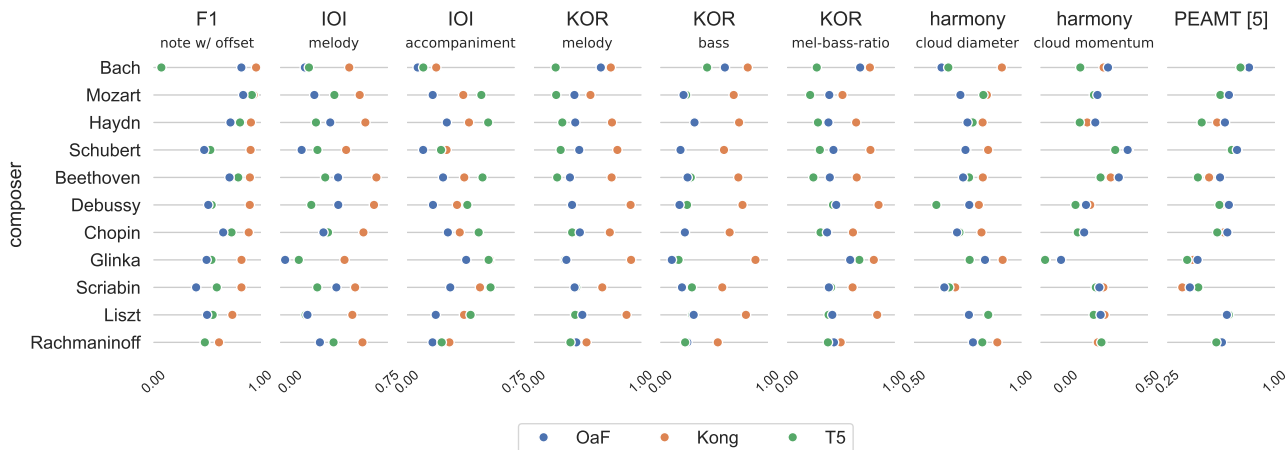


Figure 1: Model performance comparison as evaluated on note-offset F1 score and our proposed musical metrics, by composer.

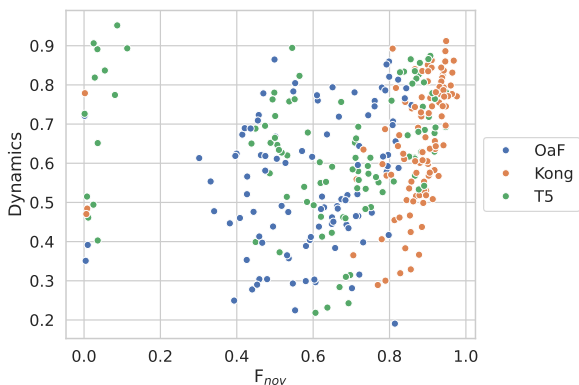


Figure 2: Relationship between model performance evaluated on note-offset-velocity F1 score and our proposed dynamics measure.

formed timing, articulation and harmony metrics. We can see again that the Kong model performs best across all composers and metrics except for the *Accompaniment IOI* timing and the *Cloud Momentum* harmony metrics. The fact that it performs better on the *Bass KOR* articulation metric, but poorly on those timing accompaniment and harmony metrics might suggest that this model detects many out-of-key extra notes that both erroneously influence the IOI sequence on the accompaniment part, and the estimation of the tonal context. Interestingly, we can also observe, as a general trend, that the F1 score somewhat deteriorates with increasing virtuoso and challenging musical repertoire. This general deterioration with increasing musical difficulty is not reflected correspondingly by our metrics, which show more variation with respect to different composers and aspects of the underlying music: While the F1 score for Bach, Mozart, Haydn and Schubert all suggest a near-perfect transcription, our metrics indicate more diverse results, and e.g., suggest poor(er) accuracy (and therefore reliability in a performance study context) in the *Melody IOI* and *Melody KOR* metrics. Another illustrative example can be found in the case of Chopin: here again the F1 score (particularly of the Kong model) would suggest a highly accurate transcription output, while our metrics reveal that the expressive dimension of articulation is not well captured. Lastly, we can observe again that

the PEAMT and harmony metric *Cloud Momentum* show a similar trend for most composers, suggesting a greater weight of the harmonic context in that trained metric.

We conclude our discussion by examining the dynamics aspect. Figure 2 illustrates the relationship between the note-offset-velocity F1 score and our proposed *Dynamics* metric. While both metrics show a weak correlation (Pearson $r = 0.21$), the figure also indicates that our metric evaluates dynamics in a more differentiated way and leads to a wider range of evaluation results than the standard metric. Note that our metric only evaluates the dynamics aspect, in particular how well the overall balance in loudness between different voice streams is preserved in a transcription. It does not account for onset, offset and pitch information, which also explains the results in the very left part of the figure that score low on F1 score but high on our metric.

6. OUT-OF-DISTRIBUTION INFERENCE

In this section we illustrate the problem of out-of-distribution performance of the models analysed. We believe that this is an important aspect to emphasize, as transcription models are ultimately intended to be (and have been) used on real-world audio performances [10, 11].

We approach the problem in two stages: First, we elucidate the problem by performing a short evaluation of the three analysed models on real-world recordings using only the standard IR metrics. Second, we simulate more challenging real-world environments using different levels of noise and reverberation, and evaluate the analysed models again using the standard and our proposed metrics, where we highlight how our musically informed metrics can reveal aspects that the standard metrics would otherwise have missed.

6.1 Generalization on real-world recordings

Table 3 shows the mean frame- and note-level F1 scores per model and per piano/acoustic recording environment on the three splits of the MAESTRO dataset (as they are officially defined).⁸

⁸ We note that for the two Magenta models, OaF and T5, our evaluation results do not come close to the reported ones in [8] and [3]. The

split	model/audio	frame			note _{off}			note _{off-vel}		
		OaF	Kong	T5	OaF	Kong	T5	OaF	Kong	T5
train	MAESTRO	0.8807	0.9207	0.7262	0.6183	0.8899	0.6350	0.5929	0.8756	0.6308
	Disklavier	0.8185	0.8508	0.6157	0.5269	0.7384	0.5132	0.4853	0.6660	0.4663
validation	MAESTRO	0.8404	0.8936	0.6546	0.6492	0.8617	0.7117	0.6236	0.8471	0.7063
	Disklavier	0.7696	0.8678	0.6093	0.5539	0.8142	0.6570	0.5161	0.7480	0.6031
test	MAESTRO	0.8527	0.9002	0.6552	0.5931	0.8215	0.5968	0.5695	0.8041	0.5896
	Disklavier	0.8049	0.8530	0.6048	0.4989	0.6979	0.5135	0.4607	0.6304	0.4624

Table 3: Frame-, note-offset, and note-offset-velocity F1 score results computed on our evaluation set, grouped per data set split, evaluated model and piano / audio environment.

We group the results per split to test whether the analysed models would perform worse on out-of-distribution recordings of performances (pieces) from the test set compared to those from the train/validation sets. Next we conduct a Kruskal Wallis ANOVA [28] to test for differences between frame-, note-offset and note-offset-velocity F1 scores, grouping the evaluation scores each by *split* and by *audio environment* and comparing each model separately. For each ANOVA we use a significance threshold of $\alpha = 0.05$. The ANOVA on the *audio environment* dimension show a statistically significant difference ($p < 0.05$) between the MAESTRO and Disklavier audio recordings for all three analysed models.

The ANOVA on the *split* dimension yields more differentiated results: For all models, the frame-level F1 scores are significantly different, and there are no statistically significant differences in the note-offset-velocity F1 scores. For the model by Kong, the note-offset-score is significantly different depending on the split, whereas the two Magenta models show no significant differences. This suggests that the most musically meaningful metric from the current set of standard metrics [1] does not sufficiently capture overfitting tendencies.

6.2 Evaluation on perturbed audio recordings

Similar as in Section 5, we again compare our musically informed metrics to the standard IR and the PEAMT metrics, however, this time on a set of more challenging audio recordings. To this end, we choose six (MAESTRO and *Disklavier*) audio recordings which we artificially perturb by introducing reverberation and synthetic noise. We use three Impulse-Response filters, modelling short, medium and long reverberation times ($RT_{60}@1\text{kHz} \in \{0.19, 1.85, 10.5\}$ seconds) and sourced from the OpenAIR⁹ database. We further add white noise into the recordings at three different Signal-to-Noise Ratio levels ($SNR_{dB} \in \{24, 12, 6\}$). Following a factorial design with these two independent variables, each with four levels, we first perturb the audio recordings on all conditions, and transcribe these recordings using all three analysed models. Following this procedure, we obtain 284 transcribed MIDI performances.¹⁰

Each grid cell in Figure 3 compares the mean note-offset F1 scores per model to the *Melody IOI* timing metric and *Cloud Momentum* harmony metric, where grid rows represent increasing reverb levels, and grid columns represent increasing noise levels. Generally, it can be observed that the performance range of models as measured by the F1 score is notably reduced compared to our metrics, indicating that our metrics possess higher discriminative capacity than the standard ones.

As expected, the inference performance of all three

differences are particularly pronounced in the note-level F1 scores.

⁹ <https://www.openair.hosted.york.ac.uk>

¹⁰ Note that 6 pieces x 4 noise levels x 4 reverberation levels x 3 models yield 288 transcriptions, but 4 recordings (each at the two higher most of either reverberation and/or noise levels) resulted in empty transcriptions (zero predicted note events) by the T5 model, and are therefore excluded from the evaluation.

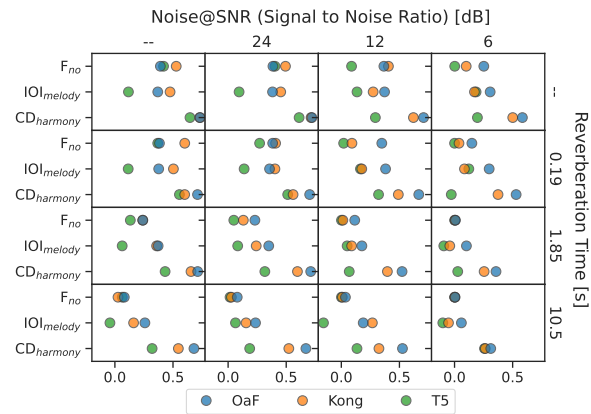


Figure 3: Performance degradation measured by note-offset F1, *Melody IOI* and *Cloud Momentum* metrics.

analysed models deteriorates with increasing noise and reverberation levels, though the deterioration is less pronounced on the noise than on the reverberation axis. Furthermore, analysing the results on the timing metric *Melody IOI* suggests that the model by Kong predicts onset times worse with increasing noise levels, while the onset times prediction by OaF seems to be more resistant to this form of perturbation. Finally, the results measured on the harmony metric *Cloud Momentum* suggest that the overall harmonic context is relatively well preserved at higher perturbation levels by the OaF and Kong models, and less so by the T5 model.

7. CONCLUSION

In this study, we investigated two aspects that are commonly neglected in the evaluation of transcription models: (i) limited explanatory power of the standard IR evaluation metrics with respect to the underlying musical material, and (ii) poor inference on out-of-distribution data. We study both problems in the context of solo piano transcription, and, in addressing the first aspect, propose a set of musically informed metrics designed to capture more musically relevant information, particularly for the context of computational studies of expressive performance.

We demonstrated our metrics on transcriptions obtained by three state-of-the-art piano transcription models on a subset of the MAESTRO dataset, the de-facto standard train and test set for current transcription models, and highlighted musical dimensions for which they provide more informative value than the standard information retrieval metrics. We have further illustrated the lack of generalization with respect to the acoustic environment, both on real-world and perturbed audio recordings.

Future work in this direction may include an extension and further validation of our new musically informed metrics, in order to capture additional qualities of expressive performance, potentially by making use of score alignment information. Additionally, a listening study with human experts could help further investigate the perceptual validity of our proposed metrics.

8. ACKNOWLEDGMENTS

This work receives funding from the European Research Council (ERC), under the European Union’s Horizon 2020 research and innovation programme, grant agreement No. 101019375 (*Whither Music?*). The LIT AI Lab is supported by the Federal State of Upper Austria.

9. REFERENCES

- [1] C. Hawthorne, E. Elsen, J. Song, A. Roberts, I. Simon, C. Raffel, J. Engel, S. Oore, and D. Eck, “Onsets and frames: Dual-objective piano transcription,” in *Proceedings of the 19th International Society for Music Information Retrieval Conference, ISMIR 2018, Paris, France, September 23-27, 2018*, 2018, pp. 50–57.
- [2] Q. Kong, B. Li, X. Song, Y. Wan, and Y. Wang, “High-resolution piano transcription with pedals by regressing onset and offset times,” *IEEE/ACM Transactions on Audio Speech and Language Processing*, vol. 29, pp. 3707–3717, 2021.
- [3] C. Hawthorne, I. Simon, R. Swavely, E. Manilow, and J. Engel, “Sequence-to-sequence piano transcription with transformers,” in *Proceedings of the 22nd International Society for Music Information Retrieval Conference, ISMIR 2021, Online, November 7-12, 2021*, 2021, pp. 246–253.
- [4] M. Bay, A. F. Ehmann, and J. S. Downie, “Evaluation of multiple-f₀ estimation and tracking systems.” in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2009, pp. 315–320.
- [5] A. Ycart, L. Liu, E. Benetos, and M. Pearce, “Investigating the perceptual validity of evaluation metrics for automatic piano music transcription,” *Transactions of the International Society for Music Information Retrieval (TISMIR)*, vol. 3, no. 1, pp. 68–81, 2020.
- [6] W.-T. Lu, J.-C. Wang, and Y.-N. Hung, “Multi-track Music Transcription with a Time-Frequency Perceiver,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2023, pp. 1–5.
- [7] B. Maman and A. H. Bermanno, “Unaligned Supervision for Automatic Music Transcription in The Wild,” in *International Conference on Machine Learning (ICML)*. PMLR, 2022, pp. 14 918–14 934.
- [8] C. Hawthorne, A. Stasyuk, A. Roberts, I. Simon, C. A. Huang, S. Dieleman, E. Elsen, J. H. Engel, and D. Eck, “Enabling Factorized Piano Music Modeling and Generation with the MAESTRO Dataset,” in *International Conference on Learning Representations, (ICLR)*, 2019.
- [9] J. Gardner, I. Simon, E. Manilow, C. Hawthorne, and J. Engel, “MT3: Multi-task multitrack music transcription,” in *International Conference on Learning Representations (ICLR)*, 2022.
- [10] Q. Kong, B. Li, J. Chen, and Y. Wang, “GiantMIDI-piano: A large-scale MIDI dataset for classical piano music,” *Transactions of the International Society for Music Information Retrieval (TISMIR)*, vol. 5, no. 1, pp. 87–98, 2022.
- [11] H. Zhang, J. Tang, S. Rafee, S. Dixon, and G. Fazekas, “ATEPP: A dataset of automatically transcribed expressive piano performance,” in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2023, pp. 446–453.
- [12] A. Daniel, V. Emiya, and B. David, “Perceptually-based evaluation of the errors usually made when automatically transcribing music,” in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2008, pp. 550–556.
- [13] A. McLeod and M. Steedman, “Evaluating Automatic Polyphonic Music Transcription,” in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2018, pp. 42–49.
- [14] V. Emiya, N. Bertin, B. David, and R. Badeau, “MAPS - A piano database for multipitch estimation and automatic transcription of music,” INRIA, Research Report, 2010. [Online]. Available: <https://inria.hal.science/inria-00544155>
- [15] S. Sigtia, E. Benetos, and S. Dixon, “An end-to-end neural network for polyphonic piano music transcription,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 5, pp. 927–939, 2016.
- [16] D. Temperley, “A Unified Probabilistic Model for Polyphonic Music Analysis,” *Journal of New Music Research*, vol. 38, no. 1, pp. 3–18, 2009. [Online]. Available: <https://doi.org/10.1080/09298210902928495>
- [17] A. Uitdenbogerd and J. Zobel, “Melodic matching techniques for large music databases,” in *Proceedings of the ACM international conference on Multimedia*, 1999, pp. 57–66. [Online]. Available: <https://dl.acm.org/doi/pdf/10.1145/319463.319470>
- [18] F. Simonetta, C. Cancino-Chacón, S. Ntalampiras, and G. Widmer, “A convolutional approach to melody line identification in symbolic scores,” in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2019, pp. 924–931.
- [19] R. Bresin and G. Umberto Battel, “Articulation strategies in expressive piano performance analysis of legato, staccato, and repeated notes in performances of

- the Andante movement of Mozart's Sonata in G Major (K 545)," *Journal of New Music Research*, vol. 29, no. 3, pp. 211–224, 2000.
- [20] B. H. Repp, "Acoustics, perception, and production of legato articulation on a computer-controlled grand piano," *The Journal of the Acoustical Society of America*, vol. 102, no. 3, pp. 1878–1890, 1995.
- [21] C. Drake and C. Palmer, "Accent structures in music performance," *Music Perception*, vol. 10, no. 3, pp. 343–378, 1993.
- [22] C. Cancino-Chacón and M. Grachten, "A computational study of the role of tonal tension in expressive piano performance," 2018. [Online]. Available: <https://arxiv.org/pdf/1807.01080>
- [23] D. Herremans and E. Chew, "Towards emotion based music generation: A tonal tension model based on the spiral array," in *The Annual Meeting of the Cognitive Science Society*, 2019, pp. 52–53. [Online]. Available: <https://hal.science/hal-03277753/document>
- [24] D. Herremans, E. Chew *et al.*, "Tension ribbons: Quantifying and visualising tonal tension." in *Second International Conference on Technologies for Music Notation and Representation (TENOR)*, 2016. [Online]. Available: <https://hal.science/hal-03165896/document>
- [25] E. Chew, "Playing with the edge: Tipping points and the role of tonality," *Music Perception: An Interdisciplinary Journal*, vol. 33, no. 3, pp. 344–366, 2016. [Online]. Available: <https://hal.science/hal-03787367/document>
- [26] R. B. Dannenberg, "The Interpretation of MIDI Velocity," in *International Computer Music Conference (ICMC)*, 2006, pp. 193–196.
- [27] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, and D. P. Ellis, "MIR_EVAL: A Transparent Implementation of Common MIR Metrics." in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2014.
- [28] A. Vargha and H. D. Delaney, "The Kruskal-Wallis Test and Stochastic Homogeneity," *Journal of Educational and Behavioral Statistics*, vol. 23, no. 2, pp. 170–192, 1998.

USING ITEM RESPONSE THEORY TO AGGREGATE MUSIC ANNOTATION RESULTS OF MULTIPLE ANNOTATORS

Tomoyasu Nakano Masataka Goto

National Institute of Advanced Industrial Science and Technology (AIST), Japan

{t.nakano, m.goto}@aist.go.jp

ABSTRACT

Human music annotation is one of the most important tasks in music information retrieval (MIR) research. Results of labeling, tagging, assessment, and evaluation can be used as training data for machine learning models that estimate them automatically. For such machine learning purposes, a single target (*e.g.*, song) is usually annotated by multiple human annotators, and the results are aggregated by majority voting or averaging. Majority voting, however, requires the number of annotators to be an odd number, which is not always possible. And averaging is sensitive to differences in the judgmental characteristics of each annotator and cannot be used for ordinal scales. This paper therefore proposes that the *item response theory (IRT)* be used to aggregate the music annotation results of multiple annotators. IRT-based models can jointly estimate annotators' characteristics and latent scores (*i.e.*, aggregations of annotation results) of the targets, and they are also applicable to ordinal scales. We evaluated the IRT-based models in two actual cases of music annotation — semantic tagging of music and Likert scale-based evaluation of singing skill — and compared those models with their simplified models that do not consider the characteristics of each annotator.

1. INTRODUCTION

Various annotations of music, such as song structure, beat timing, emotion, genre, singing phoneme, tempo, F0, singing skill, and preference, play essential roles in music information retrieval (MIR). The results of these annotations can be used not only for training machine learning models, such as deep learning models, but also for analyzing music characteristics. The results of annotations by different annotators, however, are not necessarily the same due to the ambiguity in music interpretation as well as to differences in annotators' characteristics that are individual biases stemming from factors like the experience, ability, and situation of each annotator.

Therefore, in music annotation, multiple annotators are usually assigned to the same target (*e.g.*, a song or part

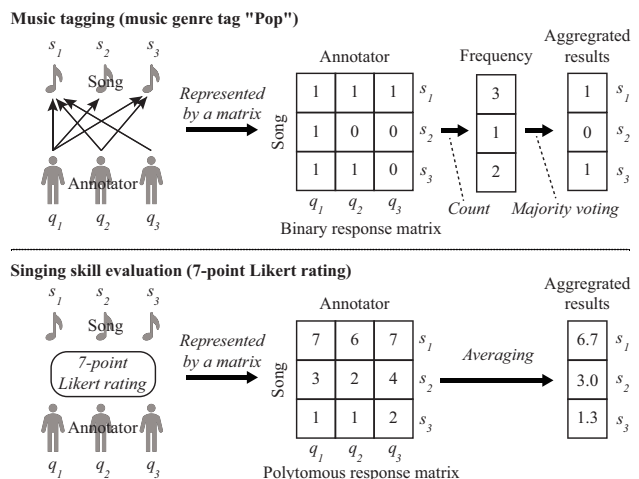


Figure 1. Examples of music annotation results of three annotators. Results of music tagging (binary rating) are aggregated by majority voting. Results of singing skill evaluation (7-point Likert rating) are aggregated by averaging.

of a song). Many studies have used multiple annotators in annotations such as singing semantic tag [1], singing ability/quality [2, 3], absolute valence-arousal annotation [4], relative valence-arousal annotation [5], song structure [6, 7], beat timing [8, 9], music semantic tag [10], and musical concept [11].

Figure 1 shows two annotation examples by multiple annotators. The first example, of music tagging, shows that annotation results of three annotators are aggregated using *majority voting*. Each annotator judges whether or not the semantic tag (music genre tag) “Pop” is applicable to each of the three target songs. The second example, of singing skill evaluation, shows that annotation results of three annotators are aggregated using *averaging*. Each annotator assigns a 7-point Likert rating to assess the singing skill in each of the three target songs. Multiple music annotation results are thus usually aggregated by two methods, majority voting [1, 5, 8, 10] and averaging [2–4].

The majority voting method requires an odd number of annotators, which is not possible in all situations. For example, if equal numbers of male and female annotators are required, the total number of annotators will be even. The binarization caused by majority voting lose information, and the averaging method cannot be used for ordinal scale values. Moreover, the two aggregation methods cannot take into account the differences in annotators' characteristics. One example of differences in annotators' char-

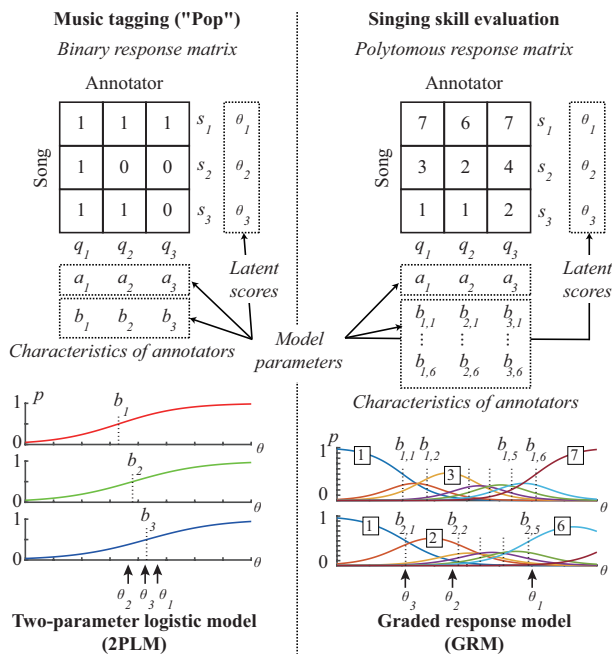


Figure 2. Examples of aggregation with two different IRT-based models, the 2PLM and the GRM. Latent scores θ can be used as aggregated results.

acteristics is that there are differences in the threshold for determining whether to tag a song in music tagging, and another is that the level of proficiency considered deserving of a perfect score in singing skill evaluation can vary depending on the annotator.

We therefore propose an aggregation method based on the *item response theory* (IRT) [12, 13] for music annotations. The IRT can take into account the differences in annotators' characteristics and aggregate annotations into latent continuous values. The first advantage of IRT is that it can be used with any number of annotators. There is no need for the number of annotators to be odd, as in majority voting, since it can estimate latent annotation scores (*i.e.*, aggregated results) for each piece of music as continuous values. The second advantage is that, when used for ordinal-scale ratings like Likert scales, it can estimate, for each annotator, different actual intervals between integer values of the rating scale.

Figure 2 shows examples of annotation aggregation by using IRT-based models. Although detailed definitions of the variables are given later in Section 3, θ_i is the latent score of a song s_i for that tag. b_j represents the annotator q_j 's characteristic, meaning the rating threshold. In the binary rating example on the left side of Figure 2, three item response functions of music tagging based on the parameters of annotators q_1 , q_2 , and q_3 are shown at the bottom of the figure. Since the song s_3 was tagged by the annotator q_2 but not by the annotator q_3 , the latent score θ_3 was higher than b_2 and lower than b_3 . On the other hand, in the singing skill evaluation example (*i.e.*, graded/polytomous rating) on the right side of Figure 2, seven functions for two annotators q_1 and q_2 are shown at the bottom of the figure as the probability that the annotators assigned each rating point based on a 7-point Likert-based rating. In this

example, a song s_2 with the latent score of θ_2 has a probability of being given scores of 3 and 2 by the annotator q_1 and q_2 , respectively. Seven such functions for each annotator represent each of these characteristics.

To show the usefulness of these IRT-based aggregation methods, we focus on two annotation tasks: music tagging as an example of binary rating and singing skill evaluation as an example of Likert scale rating. To aggregate the multiple annotation results, we use the two-parameter logistic model (2PLM) [13] and the graded response model (GRM) [14] as well-known IRT-based models. These are simple and basic models that assume unidimensionality of latent scores. These models, however, have more parameters (*e.g.*, rating thresholds and intervals) than majority voting and averaging, and cannot be properly estimated when the number of data is small [15]. This paper therefore proposes simplified versions of these models, which do not take into account the differences in annotators' characteristics, and then compares them and evaluates which model is more appropriate according to the information criterion.

2. RELATED WORK

This section describes previous research on music annotation by multiple annotators and the aggregation of their results. In addition, this section also describes applications of IRT to annotation cases.

2.1 Music Annotation Results Aggregation

There have been many cases of multiple annotators annotating the same songs in music annotation. Studies on annotators' agreement have been conducted for music genre classification [16, 17], music emotion recognition [5, 18, 19], music similarity [20], chord [21], and semantic tagging [1, 10]. The degree of inter-annotator agreement can be measured by Krippendorff's α , which is usually much smaller than 1.0 (perfect agreement) in music annotation [1, 5, 18, 19, 21], meaning that there are disagreements. Since it is only useful for evaluating agreement, not for aggregating multiple annotations, other methods such as majority voting are needed [1, 5, 10]. Even though the numbers of annotators (*i.e.*, frequencies) before majority voting were used to show the appropriateness of annotations [1, 22], they were not utilized as training data for machine learning despite their potential utility.

Music tagging or labeling is the task of binary annotation, whether tags and labels are assigned or not. Kim *et al.* [1] assigned three annotators for semantic tagging of singing voices and aggregated the results by majority voting. On the other hand, non-binary values have also been tagged. Turnbull *et al.* [10] asked annotators to vote on a 3-point scale of -1 (negative), 0 (unsure), and 1 (positive) whether the tag indicated the song. To aggregate the votes, the negative votes were subtracted from the positive votes, and the result was divided by the number of annotators.

As a polytomous annotation of ordinal scales by multiple annotators, Bogdanov *et al.* [5] performed relative annotation by three annotators and aggregated the results by

majority voting. Gupta *et al.* [2] and Sun *et al.* [3] aggregated singing quality scores on a 5-point Likert scale by averaging them. Yang *et al.* [4] assigned more than 10 annotators per song to label valence-arousal values on an 11-point scale and aggregated the results by averaging.

To overcome the limitations discussed in Section 1, we propose to use IRT for music annotation, which to the best of our knowledge has not been reported.

2.2 IRT Applications to Annotation

Although IRT has not been used in the MIR field, it has been used in the research field of natural language processing (NLP) [23]. Lalor *et al.* [24] proposed a method to generate a gold standard using IRT’s 3PLM to account for differences in item difficulty in the NLP test set. Martínez-Plumed *et al.* [25] also proposed a method to evaluate the estimation results of multiple machine learning models using 3PLM, taking into account the item difficulty of the test set. Otani *et al.* [26] proposed a framework for comparative evaluation of translation systems, utilizing an extension of the GRM. Amidei *et al.* [27] applied the IRT-based model to annotator responses and proposed a method to detect biased annotators through visualization. As a python package that can handle IRT models, `py-irt` by Lalor *et al.* [28] has been used in NLP research [29, 30].

In crowd sourcing-based annotation not limited to music, a strategy of aggregation while estimating the reliability of crowd workers has been adopted [31] and referred to as “learning from crowds” [32]. Khattak *et al.* [33] proposed and used an IRT-based model for label estimation in crowd labeling. They showed that the binarized labels based on the estimated latent scores yield better performance than conventional methods such as majority voting. Paun *et al.* [34] evaluated six Bayesian item-response models that can estimate the “true” response by aggregating multiple annotations. Several of them can estimate annotator characteristics and item difficulty. Irene Martín-Morató *et al.* [35] extended the multiple annotator competence estimation (MACE) model [36] and applied it to the sound event detection task, estimating annotator competence and excluding results from less competent annotators. Cartwright proposed a model using annotator features for crowdsourced audio quality evaluation [37].

Most closely related to this paper, Uto *et al.* [38] utilized an IRT-based model to generate training data for a deep learning model for automatic essay evaluation and to remove rater bias. This paper contributes differently from Uto *et al.* [38] not only by targeting music annotation but also by using an information criterion to compare two aggregation models and their nine simplified models.

3. IRT-BASED MUSIC ANNOTATION AGGREGATION

Item response theory (IRT) [12] is a mathematical modeling technique for testing and evaluation that was originally developed in the field of psychometrics. It models multiple *responses* (e.g., responses by multiple examinees) to

multiple *items* (e.g., questions in an exam). In our case, it models responses to multiple songs by multiple annotators. In the example in Figure 2, a probability model defines the relationship between the latent variable θ representing the latent song score and the parameters a , b representing the characteristics of the annotators. This allows, for example, music annotated with the same scores to have different latent scores θ depending on the annotators’ characteristics.

3.1 Model for binary response data

An item response model for binary response data introduces a latent score θ_i for a song i and represents the probability that the song is tagged by annotator j as follows:

$$p_{i,j}^{(2PLM)} = [1 + \exp(-a_j(\theta_i - b_j))]^{-1}, \quad (1)$$

where we used the 2PLM [13] in which the item response function is represented by a logistic function. In this equation, b_j is called *difficulty* because the tag is assigned when the score θ_i is higher than its value as shown in Figure 2. a_j is the slope of the logistic function and is called *discrimination* because it is easier to distinguish whether θ_i is higher than b_j (whether a tag is assigned) if a_j is higher.

3.2 Model for graded response (polytomous) data

The GRM [14] is a model that extends the 2PLM to response data with ordinal relationships such as those indicated by different values on a K -point Likert scale. Let $p_{i,j,k}$ be the probability that an annotator j responds to song i as category $k \in 1, \dots, K$ as follows:

$$p_{i,j,k} = p_{i,j,k-1}^{*(GRM)} - p_{i,j,k}^{*(GRM)}, \quad (2)$$

$$p_{i,j,k}^{*(GRM)} = [1 + \exp(-a_j(\theta_i - b_{j,k}))]^{-1}, \quad (3)$$

where k means the order of the categories. $p_{i,j,0}^* = 1$, and $p_{i,j,K}^* = 0$. The $b_{j,k}$ represents the difficulty in responding to categories greater than k in annotator j .

3.3 Nine originally simplified models

To evaluate usefulness of the above 2PLM and GRM in music annotation, we compare the simpler 1PLM [13] in which the parameter a_j is removed from the 2PLM (*i.e.*, the slope is not considered) as follows:

$$p_{i,j}^{(1PLM)} = [1 + \exp(-(\theta_i - b_j))]^{-1}. \quad (4)$$

Moreover, we here propose two further simpler models with reduced parameters, in which the parameters a_j and b_j are replaced by a and b (*i.e.*, the characteristics of the annotator are not considered), as follows:

$$p_{i,j}^{(2PLM')} = [1 + \exp(-a(\theta_i - b))]^{-1}, \quad (5)$$

$$p_{i,j}^{(1PLM')} = [1 + \exp(-(\theta_i - b))]^{-1}. \quad (6)$$

Regarding the GRM, we also propose the following three simplified models based on the same idea:

$$p_{i,j,k}^{*(GRM-a)} = [1 + \exp(-(\theta_i - b_{j,k}))]^{-1}, \quad (7)$$

$$p_{i,j,k}^{*(GRM')} = [1 + \exp(-a(\theta_i - b_k))]^{-1}, \quad (8)$$

$$p_{i,j,k}^{*(GRM-a')} = [1 + \exp(-(\theta_i - b_k))]^{-1}. \quad (9)$$

Although the GRM is designed for ordinal scales, we further propose four simplified models that assume that annotators’ responses are on interval scales (i.e., the intervals between the (cut) points are equally spaced) as follows.

$$p_{i,j,k}^{*(\text{GRMi})} = [1 + \exp(-a_j(\theta_i - (o_j + k'b_j)))]^{-1}, \quad (10)$$

$$p_{i,j,k}^{*(\text{GRMi-a})} = [1 + \exp(-(\theta_i - (o_j + k'b_j)))]^{-1}, \quad (11)$$

$$p_{i,j,k}^{*(\text{GRMi}') } = [1 + \exp(-a(\theta_i - (o + k'b)))]^{-1}, \quad (12)$$

$$p_{i,j,k}^{*(\text{GRMi-a}') } = [1 + \exp(-(\theta_i - (o + k'b)))]^{-1}, \quad (13)$$

where o_j and b_j denote the annotator-dependent origins and intervals, respectively, and o and b are annotator-independent origin and interval, respectively. We set $k' = k - 1$ in our current implementation.

4. EXPERIMENT

Using the IRT-based models described in the previous sections, we report the results of aggregating annotation results from multiple annotators in two real cases (Figure 1): music tagging (binary response) and singing skill evaluation based on 7-point Likert rating (polytomous response).

4.1 Aggregation of music tagging results

As an actual example of the aggregation of music annotation using the 2PLM, we targeted Japanese lyrics songs in our in-house database and music tags assigned to them.

4.1.1 Data (songs and annotations)

We prepared 120 songs with Japanese lyrics. However, as we only aim to demonstrate the effectiveness of the proposed models, any dataset of annotated songs will suffice. Annotators were six music experts whose native language was Japanese (three males, referred to as M1-M3, and three females, referred to as F1-F3). Each annotator tagged 60 songs, half of the 120 songs. To avoid gender distribution bias, the annotators were divided into two groups of three: “M1, F1, F3” (Group 1) and “M2, M3, F2” (Group 2), and the annotators in the same group tagged the same songs.

The annotators were instructed to annotate one or more of each of 15 genres, 38 subgenres, and 28 semantics. They tagged genres first, then subgenres and semantics. The 15 music genres are based on Discogs¹, which is a large open database of music genres and has been the target of research on metadata analysis [39] and music genre embedding [40, 41]. The 38 subgenres and 28 semantics (emotions, moods, and themes) were based on previous works [10, 42–46] using well-known datasets: MagnaTagATune (MTAT) [47], Million Song Dataset (MSD) [48], MTG-Jamendo [45], and CAL500exp [46]. In total, 81 tags were thus annotated.

4.1.2 Model

As described in Section 3.1, the 2PLM shown in Equation (hereafter Eqn) (1) and its simplified models (Eqns (4, 5, 6)) are used to model music tagging. For each tag t , we

¹ <https://www.discogs.com/ja/>

Table 1. Pairwise comparison of the four models. The columns represent the reference models, and the rows represent the models being compared. Bolded numbers indicate the number of tags with higher ELPD than those of the model being compared.

Model	Annotator independent		Annotator dependent	
	b Eqn (6)	a, b Eqn (5)	b_j Eqn (4)	a_j, b_j Eqn (1)
b	–	15 + 17	54 + 55	21 + 32
a, b	66 + 64	–	61 + 59	46 + 46
b_j	27 + 26	20 + 22	–	4 + 4
a_j, b_j	60 + 49	35 + 35	77 + 77	–

jointly estimate parameters, θ_i^t , a_j^t , and b_j^t using binary response data $U^t = \{u_{i,j}^t\} (i = 1 \cdots N_s^t, j = 1 \cdots N_a^t)$. Here θ_i^t represents the latent score of t for song i . a_j^t and b_j^t represent a characteristic of annotator j . N_s^t is the number of songs and N_a^t is the number of annotators.

In this paper we assume the following prior distributions for the parameters of the 2PLM.

$$\theta_i^t \sim \text{Normal}(0.0, 1.0), \quad i = 1 \cdots N_s^t, \quad (14)$$

$$a_j^t \sim \text{HalfNormal}(1.0), \quad j = 1 \cdots N_a^t, \quad (15)$$

$$b_j^t \sim \text{Normal}(0.0, 1.0), \quad j = 1 \cdots N_a^t. \quad (16)$$

Here a_j^t is not used when using the 1PLM, and a^t and b^t are used for the simplified models.

In this paper, since there is no overlap between the songs annotated by the two groups, we estimate θ_i by treating the results for “M1, F1, F3” and “M2, M3, F2” separately. Thus, the number of songs $N_s^t = 60$ and the number of annotators $N_a^t = 3$. The model parameters θ, a, b were estimated directly using the No-U-Turn Sampler (NUTS) [49], a type of Markov chain Monte Carlo (MCMC) method. We used a python package PyMC5 [50] to implement it. The number of burn-in samples was set to 5000, the number of draws to 10000, and the number of chains to 4. In other words, 40000 posterior samples were used and their posterior mean was used as the estimation result. Convergence was confirmed using the convergence diagnostic $\hat{R} < 1.01$ and effective sample size (ESS) > 400 as proposed by Vehtari *et al.* [51].

4.1.3 Results

To evaluate the proposed models, we used *expected log pointwise predictive density (ELPD)* values [52] as an information criterion. To estimate ELPD, we employed the leave-one-out (LOO) cross-validation estimate with Pareto smoothed importance sampling (PSIS) [52]. The higher the ELPD, the better the model. We conducted a pairwise comparison of the four models to evaluate the 81 tags annotated by the two groups. Table 1 shows, for each model in a column, the number of tagging evaluations that had a higher ELPD than the model in the corresponding row. For example, b denotes the model in Eqn (6). Here, the number of tags with higher ELPD is $66 + 64 = 130$ when compared to the a, b model in Eqn (5). The left side of the “+” sign indicates the number in Group 1, and the right side indicates the number in Group 2.

The results in Table 1 show that 1PLM (Eqn (4)) was

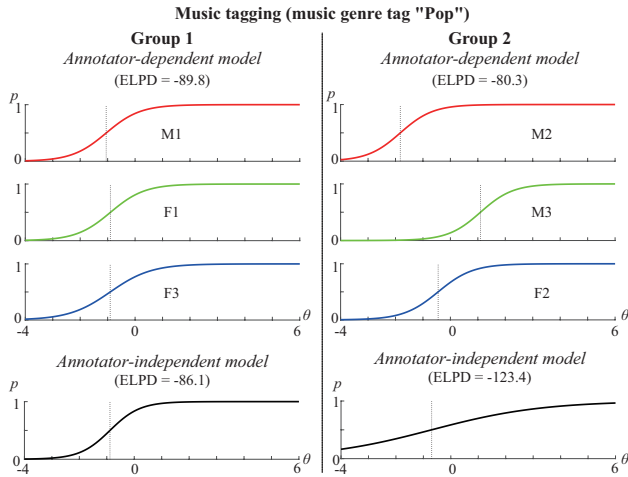


Figure 3. Examples of annotator characteristic curves for groups of three annotators each, annotating different 60 songs, for the music tag “Pop”. Parameter estimation results obtained by using annotator-dependent (Eqn (1)) and annotator-independent (Eqn (5)) models are shown.

most often the best IRT-based model for aggregating music tagging results. Table 1 also shows that in many cases the model that does not take into account the characteristics of annotators (Eqn (6)), was also better. Estimation results for the annotator-dependent and annotator-independent models are shown in Figure 3. When the annotators have different characteristics in annotating the tag “Pop” as shown in Group 2, the annotator-dependent 2PLM model has higher ELPD value to the simplified annotator-independent model as expected. Conversely, since the characteristics of Group 1 annotators are similar, the simplified annotator-independent model is superior in this case.

4.2 Aggregation of Likert scale evaluation results

As an actual example of the aggregation of music annotation using the GRM and its simplified models, we targeted the singing of Japanese lyrics in our in-house database and the results of singing skill evaluation annotated to them.

4.2.1 Data (songs and annotations)

We prepared another database comprising a total of 140 solo singing renditions with Japanese lyrics. This contains 20 songs of RWC-MDB [53], as well as 120 cover versions in which each of the 20 songs was sung by six additional singers. Ten songs were sung by male singers, while the remaining ten songs were sung by female singers. For 120 cover versions, there are a total of 40 singers, 20 male and 20 female, with a wide variety of singing experience (*i.e.*, each additional singer sung 3 songs).

These songs were annotated with detailed singing evaluations by 10 annotators who are experts for music and singing: 5 males (M4 to M8) and 5 females (F4 to F8). Singing evaluations were conducted on the singing voices mixed with the accompaniments (karaoke). Annotators conducted a 7-point evaluation from six evaluation perspectives: pitch, rhythm, pronunciation, expression, vocal projection, and overall performance. In order to control

Table 2. 7-point criteria for singing skill evaluation

Score	Criteria
7	Professional singer
6	Semi-professional (can receive a reward)
5	Amateur taking lessons to become a pro
4	Good at karaoke
3	Not so good at karaoke, but not so bad
2	Goes to karaoke, but is not very good at it
1	Poor singer and does not go to karaoke

Table 3. Results of the singing evaluation for a female singer song (evaluation perspective: overall performance). The singer ID “-” means the original singer.

ID	M4	M5	M6	M7	M8	F4	F5	F6	F7	F8
-	6	4	7	5	6	5	6	5	5	6
23	6	5	6	6	7	6	6	6	6	7
26	4	4	5	4	5	3	4	4	5	3
31	3	3	4	4	4	3	4	4	3	3
34	4	3	3	3	3	3	3	3	3	3
37	2	2	2	2	2	2	2	2	2	2
40	1	1	1	1	1	1	1	1	1	1

the evaluation criteria for each annotator, we specified the criteria shown in Table 2 and presented actual singing examples for each of the seven scores in advance.

4.2.2 Example of data

Table 3 shows the results of the 7-point evaluation of the singing skill for an example (RWC-MDB-P No.7) out of the 20 songs for “overall performance”. Although only the results of one evaluation perspective for one song are shown here, these evaluation results were actually obtained for each of the 140 songs, with the 6 different perspectives.

From Table 3 it can be seen that the evaluation scores differed among the annotators, and that there were cases where the evaluation values differed as much as 3 out of 7 points among the annotators (ID “-”). On the other hand, there were cases where all annotators had the same evaluation value of 1, as in the case of ID 40 for this song.

4.2.3 Model

As described in Section 3.2, the GRM is used to model the Likert scale in the singing skill evaluation. For each perspective p , we jointly estimate parameters, θ_i^p , a_j^p , and $b_{j,k}^p$ using polytomous response data $X^p = \{x_{i,j}^p\} (i = 1 \cdots N_s^p, j = 1 \cdots N_a^p)$, where $N_s^p = 140$ is the number of songs, $N_a^p = 10$ is the number of annotators, and $K = 7$.

In this paper we assume the following prior distributions for the parameters of the GRM.

$$\theta_i^p \sim \text{Normal}(0.0, 1.0), \quad i = 1 \cdots N_s^p, \quad (17)$$

$$a_j^p \sim \text{HalfNormal}(1.0), \quad j = 1 \cdots N_a^p, \quad (18)$$

$$b_{j,k}^p \sim \text{Normal}(\mu_k, 1.0), \quad k = 1 \cdots K - 1, \quad (19)$$

where μ_k is equally spaced from $\mu_1 = -0.1$ to $\mu_{K-1} = 0.1$. The models in Eqns (7, 9, 11, 13) do not use a_j^p , and the models without j use a^p and b_k^p .

The prior distributions in the simplified GRM-based models that assume an interval scale are as follows:

$$\theta_j^p \sim \text{Normal}(-4.0, 3.0), \quad j = 1 \cdots N_a^p, \quad (20)$$

$$b_j^p \sim \text{HalfNormal}(3.0), \quad j = 1 \cdots N_a^p. \quad (21)$$

The MCMC setting was same as in Section 4.1.2.

Table 4. PSIS-LOO estimates (values of the expected log pointwise predictive density (ELPD)). The higher, the better. The highest value in each perspective is bolded and underlined, and the second highest value is underlined.

Perspective	Annotator independent				Annotator dependent			
	$o + k'b$		b_k		$o_j + k'b_j$		$b_{j,k}$	
	$-$	a	$-$	a	$-$	a_j	$-$	a_j
	Eqn (13)	Eqn (12)	Eqn (9)	Eqn (8)	Eqn (11)	Eqn (10)	Eqn (7)	Eqn (3)
Expression	-1864.0	-1720.8	-1857.3	-1699.8	-1864.1	-1685.3	-1871.9	-1706.7
Overall performance	-1729.6	-1496.4	-1726.9	<u>-1456.4</u>	-1729.5	-1414.6	-1759.1	-1528.5
Pitch	-1871.2	-1712.3	-1853.8	-1658.8	-1870.9	-1569.6	-1796.9	-1600.8
Pronunciation	-1887.3	-1773.9	-1870.8	-1747.5	-1887.2	-1741.3	-1885.6	-1763.9
Rhythm	-1903.3	-1794.1	-1868.1	-1746.4	-1903.5	-1698.0	-1825.0	-1702.2
Vocal projection	-1828.1	-1671.3	-1807.4	-1630.1	-1828.1	-1608.6	-1832.3	-1667.7

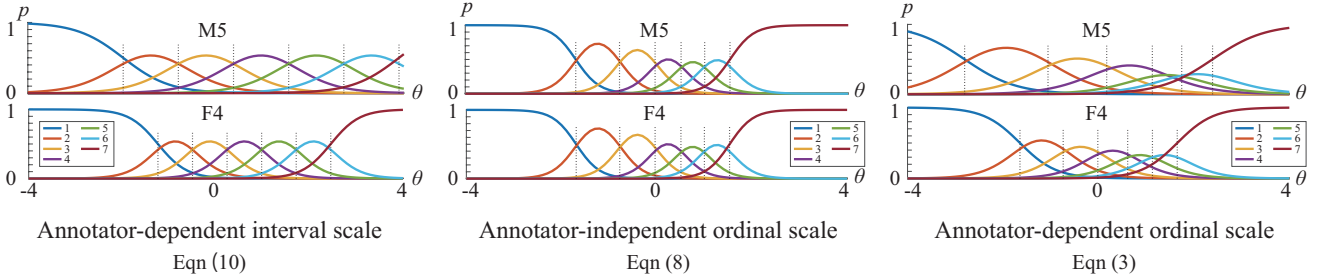


Figure 4. Item response category characteristic curves based on estimates of the parameters of annotators M5 and F4 for “overall performance”. The leftmost curves are for a simplified model (Eqn (10)) with interval scales. The center curves are for a simplified annotator-independent GRM-based model (Eqn (8)). The rightmost are for the GRM model (Eqn (3)).

4.2.4 Results

Table 4 shows the results of the model comparison. The model assuming annotator-dependent and interval measures (Eqn (10)) always performed the best. The second-best performing model was the annotator-independent one with variable intervals between cut points (Eqn (8)), or one that is the GRM (Eqn (3)).

Figure 4 visualizes the characteristics of two annotators, M5 and F4, by the three models that obtained the best evaluation results in Table 4. It can be seen that, given the same annotation data, the best simplified model in Eqn (10) estimates an equal interval scale for each annotator. While the GRM model in Eqn (3) can estimate the intervals that vary depending on both the seven categories and the two annotators, the simplified GRM-based model of Eqn (8) estimates the intervals that are shared by the ten annotators. These results suggest that evaluation scores tend to vary in intervals between annotators and/or within annotators. This means that these models potentially outperform conventional averaging-based methods, which assume annotator-independence and interval scales.

5. DISCUSSION

In the task of estimating music tags by using deep learning, binary labels are used to indicate whether the tag is assigned (1) or not (0), and are learned using the binary cross entropy loss [54]. Thus a continuous value of 0 to 1 is obtained during prediction, but the training data did not have such a continuum. In actual music tagging, however, the lack of perfect agreement among annotators means that it would be useful to represent each tag as a continuous value θ obtained by IRT when preparing the ground-truth training data for each tag. In fact, there are studies that have an-

alyzed the degree of such agreement based on the annotation results of multiple annotators in the annotation of segment boundaries of music structure in a musical piece [22].

In addition, if Likert scale-based ratings are used as machine learning data, they are typically averaged to obtain aggregated values. However, our experimental results show that these intervals can indeed differ among annotators. Thus, the proposed IRT-based aggregation has the advantage of dealing with ordinal scales.

In deep learning, there are methods to output discrete categories with ordinal relations by replacing the ordinal regression problem with binary classification subproblems and aggregating them [55, 56]. The IRT-based aggregation can replace ordinal regression as a regression problem and treat it with continuous values, which has the potential to improve machine learning performance even more.

6. CONCLUSION

This paper proposes the use of IRT for aggregating music annotation results from multiple annotators. Among the diverse types of music annotation, we targeted tagging and Likert scale-based evaluation, both of which have high practical potential. Specifically, we focused on aggregating results of music semantic tagging and singing skill evaluation using IRT’s 2PLM and GRM, respectively. We also proposed nine simplified models and verified the effectiveness of the proposed IRT-based models.

In the future, we plan to evaluate the effectiveness of IRT-based models on various datasets and annotations. Depending on the dataset, there may be new challenges to consider, such as introducing models to estimate the reliability and competence of the annotators [34–36]. Moreover, we will verify the effectiveness of using θ as training data in machine learning.

7. ACKNOWLEDGMENTS

This work was supported in part by JST CREST Grant Number JPMJCR20D4 and JSPS KAKENHI Grant Number JP21H04917, Japan.

8. REFERENCES

- [1] K. L. Kim, J. Lee, S. Kum, C. L. Park, and J. Nam, "Semantic tagging of singing voices in popular music recordings," *IEEE/ACM TASLP*, vol. 28, pp. 1656–1668, 2020.
- [2] C. Gupta, H. Li, and Y. Wang, "Perceptual evaluation of singing quality," in *Proc. APSIPA-ASC 2017*, 2017, pp. 577–586.
- [3] X. Sun, Y. Gao, H. Lin, and H. Liu, "TG-Critic: A timbre-guided model for reference-independent singing evaluation," in *Proc. IEEE ICASSP 2023*, 2023, pp. 1–5.
- [4] Y.-H. Yang, Y.-C. Lin, Y.-F. Su, and H. H. Chen, "A regression approach to music emotion recognition," *IEEE TASLP*, vol. 16, no. 2, p. 448–457, 2008.
- [5] D. Bogdanov, X. Lizarraga-Seijas, P. Alonso-Jiménez, and X. Serra, "MusAV: A dataset of relative arousal-valence annotations for validation of audio models," in *Proc. ISMIR 2022*, 2022, pp. 650–658.
- [6] J. B. L. Smith, J. A. Burgoyne, I. Fujinaga, D. D. Roure, and J. S. Downie, "Design and creation of a large-scale database of structural annotations," in *Proc. ISMIR 2011*, 2011, pp. 555–560.
- [7] O. Nieto and J. P. Bello, "Systematic exploration of computational music structure research," in *Proc. ISMIR 2016*, 2016, pp. 547–553.
- [8] M. F. McKinney, D. Moelants, M. E. P. Davies, and A. Klapuri, "Evaluation of audio beat tracking and music tempo extraction algorithms," *J. New Music Res.*, vol. 36, no. 1, pp. 1–16, 2007.
- [9] O. Nieto, M. C. McCallum, M. E. P. Davies, A. Robertson, A. M. Stark, and E. Egozy, "The harmonix set: Beats, downbeats, and functional segment annotations of western popular music," in *Proc. ISMIR 2019*, 2019, pp. 565–572.
- [10] D. Turnbull, L. Barrington, D. A. Torres, and G. R. G. Lanckriet, "Semantic annotation and retrieval of music and sound effects," *IEEE Trans. Speech Audio Process.*, vol. 16, no. 2, pp. 467–476, 2008.
- [11] Y.-H. Yang, Y.-C. Lin, A. Lee, and H. H. Chen, "Improving musical concept detection by ordinal regression and context fusion," in *Proc. ISMIR 2009*, 2009, pp. 147–152.
- [12] F. M. Lord, *Applications of Item Response Theory to Practical Testing Problems*. L. Erlbaum Associates, 1980.
- [13] R. K. Hambleton, H. Swaminathan, and H. J. Rogers, *Fundamentals of Item Response Theory*. Sage Publications, 1991.
- [14] F. Samejima, "Estimation of latent ability using a response pattern of graded scores," *Psychometrika monograph supplement*, 1969.
- [15] R. K. Hambleton and R. W. Jones, "Comparison of classical test theory and item response theory and their applications to test development," *Educational Measurement: Issues and Practice*, vol. 12, no. 3, pp. 38–47, 1993.
- [16] S. Lippens, J.-P. Martens, and T. D. Mulder, "A comparison of human and automatic musical genre classification," in *Proc. IEEE ICASSP 2004*, 2004, pp. 233–236.
- [17] K. Seyerlehner, G. Widmer, and P. Knees, "A comparison of human, automatic and collaborative music genre classification and user centric evaluation of genre classification systems," in *Proc. AMR 2010*, 2010, pp. 118–131.
- [18] M. Soleymani, A. Aljanaki, Y.-H. Yang, M. N. Caro, F. Eyben, K. Markov, B. W. Schuller, R. C. Veltkamp, F. Weninger, and F. Wiering, "Emotional analysis of music: A comparison of methods," in *Proc. ACMMM 2014*, 2014, pp. 1161–1164.
- [19] J. Fan, K. Tatar, M. Thorogood, and P. Pasquier, "Ranking-based emotion recognition for experimental music," in *Proc. ISMIR 2017*, 2017, pp. 368–375.
- [20] A. Flexer and T. Grill, "The problem of limited interrater agreement in modelling music similarity," *J. New Music Res.*, vol. 45, no. 3, pp. 239–251, 2016.
- [21] H. V. Kooops, W. B. de Haas, J. A. Burgoyne, J. Bransen, A. Kent-Muller, and A. Volk, "Annotator subjectivity in harmony annotations of popular music," *J. New Music Res.*, vol. 48, no. 3, pp. 232–252, 2019.
- [22] M. J. Bruderer, M. McKinney, and A. Kohlrausch, "Structural boundary perception in popular music," in *Proc. ISMIR 2006*, 2006, pp. 198–201.
- [23] J. P. Lalor, P. Rodriguez, J. Sedoc, and J. Hernandez-Orallo, "Item response theory for natural language processing," in *Proc. EACL 2024: Tutorial Abstracts*, 2024.
- [24] J. Lalor, H. Wu, and H. Yu, "Building an evaluation scale using item response theory," in *Proc. EMNLP 2016*, 2016, pp. 648–657.
- [25] F. Martínez-Plumed, R. B. C. Prudêncio, A. Martínez-Usó, and J. Hernández-Orallo, "Making sense of item response theory in machine learning," in *Proc. ECAI 2016*, 2016, pp. 1140–1148.

- [26] N. Otani, T. Nakazawa, D. Kawahara, and S. Kurohashi, “Irt-based aggregation model of crowdsourced pairwise comparison for evaluating machine translations,” in *Proc. EMNLP 2016*, 2016, pp. 511–520.
- [27] J. Amidei, P. Piwek, and A. Willis, “Identifying annotator bias: A new IRT-based method for bias identification,” in *Proc. COLING 2020*, 2020, pp. 4787–4797.
- [28] J. P. Lalor and P. Rodriguez, “py-irt: A scalable item response theory library for Python,” *INFORMS Journal on Computing*, 2023.
- [29] J. P. Lalor, H. Wu, and H. Yu, “Learning latent parameters without human response patterns: Item response theory with artificial crowds,” in *Proc. EMNLP 2019*, 2019.
- [30] P. Rodriguez, J. Barrow, A. M. Hoyle, J. P. Lalor, R. Jia, and J. Boyd-Graber, “Evaluation examples are not equally informative: How should that change nlp leaderboards?” in *Proc. ACL-IJCNLP 2021*, 2021, pp. 4486–4503.
- [31] P. O’Donovan, J. Libeks, A. Agarwala, and A. Hertzmann, “Exploratory font selection using crowdsourced attributes,” *ACM Transactions on Graphics (TOG)*, vol. 33, no. 4, pp. 1–9, 2014.
- [32] V. C. Raykar, S. Yu, L. H. Zhao, G. H. Valadez, C. Florin, L. Bogoni, and L. Moy, “Learning from crowds,” *Journal of Machine Learning Research*, vol. 11, pp. 1297–1322, 2010.
- [33] F. K. Khattak, A. Salleb-Aouissi, and A. Raja, “Accurate crowd-labeling using item response theory,” *Collective Intelligence*, 2016.
- [34] S. Paun, B. Carpenter, J. Chamberlain, D. Hovy, U. Kruschwitz, and M. Poesio, “Comparing bayesian models of annotation,” *Transactions of the Association for Computational Linguistics*, vol. 6, pp. 571–585, 2018.
- [35] I. Martín-Morató and A. Mesaros, “Strong labeling of sound events using crowdsourced weak labels and annotator competence estimation,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 902–914, 2023.
- [36] D. Hovy, T. Berg-Kirkpatrick, A. Vaswani, and E. H. Hovy, “Learning whom to trust with mace,” in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (HLT-NAACL)*, 2013, pp. 1120–1130.
- [37] M. Cartwright, “Supporting novice communication of audio concepts for audio production tools,” Ph.D. dissertation, Northwestern University, December 2016.
- [38] M. Uto and M. Okano, “Learning automated essay scoring models using item-response-theory-based scores to decrease effects of rater biases,” *IEEE Transactions on Learning Technologies*, vol. 14, no. 6, pp. 763–776, 2021.
- [39] D. Bogdanov and X. Serra, “Quantifying music trends and facts using editorial metadata from the discogs database,” in *Proc. ISMIR 2017*, 2017, pp. 89–95.
- [40] R. Hennequin, J. Royo-Letelier, and M. Moussallam, “Audio based disambiguation of music genre tags,” in *Proc. ISMIR 2018*, 2018, pp. 645–652.
- [41] P. Alonso-Jiménez, X. Serra, and D. Bogdanov, “Music representation learning based on editorial metadata from discogs,” in *Proc. ISMIR 2022*, 2022, pp. 825–833.
- [42] E. Law, K. West, M. I. Mandel, M. Bay, and J. S. Downie, “Evaluation of algorithms using games: The case of music tagging,” in *Proc. ISMIR 2009*, 2009, pp. 387–392.
- [43] J. Lee, J. Park, K. L. Kim, and J. Nam, “Sample-level deep convolutional neural networks for music auto-tagging using raw waveforms,” in *Proc. SMC 2017*, 2017, pp. 220–226.
- [44] J. Pons, O. Nieto, M. Prockup, E. Schmidt, A. Ehmann, and X. Serra, “End-to-end learning for music audio tagging at scale,” in *Proc. ISMIR 2018*, 2018, pp. 637–644.
- [45] D. Bogdanov, M. Won, P. Tovstogan, A. Porter, and X. Serra, “The MTG-jamendo dataset for automatic music tagging,” in *Proc. ICML 2019*, 2019.
- [46] S.-Y. Wang, J.-C. Wang, Y.-H. Yang, and H.-M. Wang, “Towards time-varying music auto-tagging based on cal500 expansion,” in *Proc. IEEE ICME 2014*, 2014, pp. 1–6.
- [47] E. Law and L. von Ahn, “Input-agreement: A new mechanism for collecting data using human computation games,” in *Proc. ACM CHI 2009*, 2009, pp. 1197–1206.
- [48] T. Bertin-Mahieux, D. P. Ellis, B. Whitman, and P. Lamere, “The million song dataset,” in *Proc. ISMIR 2011*, 2011, pp. 591–596.
- [49] M. D. Hoffman and A. Gelman, “The No-U-Turn Sampler: adaptively setting path lengths in hamiltonian monte carlo,” *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1593–1623, 2014.
- [50] O. A. Pla, V. Andréani, C. Carroll, L. Dong, C. Fonnebeck, M. Kochurov, R. Kumar, J. Lao, C. C. Luhmann, O. A. Martin, M. Osthege, R. Vieira, T. V. Wiecki, and R. Zinkov, “PyMC: A modern and comprehensive probabilistic programming framework in python,” *PeerJ Computer Science*, vol. 9, p. e1516, 2023.

- [51] A. Vehtari, A. Gelman, D. Simpson, B. Carpenter, and P.-C. Bürkner, “Rank-normalization, folding, and localization: An improved \hat{R} for assessing convergence of MCMC,” *Bayesian Analysis*, pp. 1–38, 2021.
- [52] A. Vehtari, A. Gelman, and J. Gabry, “Practical bayesian model evaluation using leave-one-out cross-validation and waic,” *Statistics and Computing*, vol. 27, no. 5, pp. 1413–1432, 2017.
- [53] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, “RWC Music Database: Popular, classical, and jazz music databases,” in *Proc. ISMIR 2002*, 2002, pp. 287–288.
- [54] M. Won, A. Ferraro, D. Bogdanov, and X. Serra, “Evaluation of CNN-based automatic music tagging models,” in *Proc. SMC 2020*, 2020.
- [55] Z. Niu, M. Zhou, L. Wang, X. Gao, and G. Hua, “Ordinal regression with multiple output CNN for age estimation,” in *Proc. CVPR 2016*, 2016, pp. 4920–4928.
- [56] S. Chen, C. Zhang, M. Dong, J. Le, and M. Rao, “Using ranking-CNN for age estimation,” in *Proc. CVPR 2017*, 2017, pp. 742–751.

JUST LABEL THE REPEATS FOR IN-THE-WILD AUDIO-TO-SCORE ALIGNMENT

Irmak Bukey
Carnegie Mellon University

Michael Feffer
Carnegie Mellon University

Chris Donahue
Carnegie Mellon University

ABSTRACT

We propose an efficient workflow for high-quality offline alignment of in-the-wild performance audio and corresponding sheet music scans (images).¹ Recent work on audio-to-score alignment extends dynamic time warping (DTW) to be theoretically able to handle *jumps* in sheet music induced by repeat signs—this method requires no human annotations, but we show that it often yields low-quality alignments. As an alternative, we propose a workflow and interface that allows users to quickly annotate jumps (by clicking on repeat signs), requiring a small amount of human supervision but yielding much higher quality alignments on average. Additionally, we refine audio and score feature representations to improve alignment quality by: (1) integrating measure detection into the score feature representation, and (2) using raw onset prediction probabilities from a music transcription model instead of piano roll. We propose an evaluation protocol for audio-to-score alignment that computes the distance between the estimated and ground truth alignment in units of measures. Under this evaluation, we find that our proposed jump annotation workflow and improved feature representations together improve alignment accuracy by 150% relative to prior work (33% \rightarrow 82%).

1. INTRODUCTION

Sheet music has been used as a primary means of communicating musical ideas for centuries. Accordingly, sheet music is a profoundly important modality for MIR, not only because of the breadth of musical knowledge and history contained within, but also because sheet music constitutes a vital interface between MIR systems and musicians. However, while multimodal MIR systems are rapidly improving at tasks like music transcription [3–6] and controllable generation [7–9], these systems typically operate on MIDI as a symbolic music format. This may be less useful to musicians, e.g., a musician might prefer transcription systems to output sheet music instead of MIDI.

¹ Video examples: <https://bit.ly/jltr-ismir2024>
Code: <https://github.com/irmakbky/jltr-alignment>
Corresponding author: Irmak Bukey <ibukey@cs.cmu.edu>

We conjecture that the scarcity of fine-grained alignment data linking sheet music to corresponding performance audio is a key bottleneck to incorporating sheet music into multimodal MIR systems. Alignments allow multimodal MIR data to be segmented into input-output chunks of tractable length for training models, and the lack of sheet music alignments may partially explain why sheet music is mostly overlooked. Moreover, alignments have practical utility outside of multimodal MIR, e.g., they may be used by musicians to practice along with pre-recorded accompaniments. Unfortunately, collecting alignments is deceptively tricky. For example, one could have a musician use a touch screen to point to the current location in sheet music while listening to a recording in real time. However, their tracking may be imprecise (due to expressive performance timing) and lack non-obvious details that are essential for segmentation (bar line locations, number of active staves).

In this work, we investigate the task of alignment of offline in-the-wild performance audio and corresponding sheet music scans (images), with a long-term goal of aligning large corpora of sheet music and performance recordings at scale. Much of the past work on audio-to-score alignment make at least one of several common assumptions that inhibit their practicality for collecting aligned data at scale: (i) the presumed availability of digital scores like MIDI or MusicXML as opposed to sheet music images [10–14], (ii) the alignment of MIDI performances or synthesized audio instead of real audio recordings [12, 15, 16], (iii) limitations in instrument diversity, commonly piano only [10, 16, 17], or (iv) dependence on time-consuming human annotation [18, 19].

Here we propose an audio-to-score alignment procedure that makes none of these assumptions, potentially offering a path forward for large-scale data collection. Most closely related to our approach is that of Shan et al. [16, 17], who examine offline alignment of in-the-wild piano sheet music images and performance recordings by aligning feature representations derived from the score and audio via MIR methods. In addition to operating on more diverse ensembles, our work has two primary distinctions: (1) we take a different approach to handling *jumps* in scores, and (2) we modify their feature representations.

A key challenge in audio-to-score alignment is handling inter-measure jumps in scores induced by repeat signs. Shan et al. [16, 17] propose extensions to DTW that are capable of automatically handling jumps. Here we propose a pragmatic alternative: a workflow and interface that allows humans to quickly annotate jumps, and a system that incor-



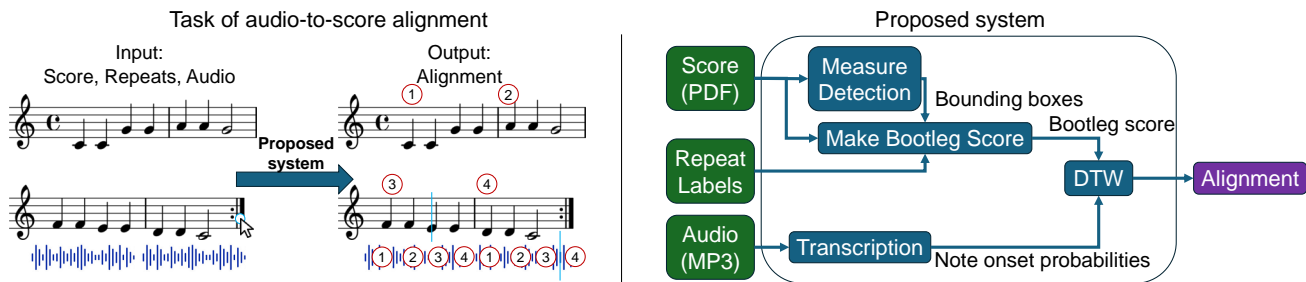


Figure 1. An overview of the task of audio-to-score alignment and our proposed approach. Given a score image (as a PDF) and corresponding performance audio (e.g., an MP3) as input, the task involves outputting an alignment between time in the recording and playheads in the score image. A key challenge in this task is handling jumps in the score, e.g., those created by repeat signs. In lieu of robust automatic methods for detecting or handling jumps, we propose a pragmatic approach of having experts simply label the repeats, which can be done quickly and greatly improves task performance. Our proposed system combines the repeat labels with score feature representations inspired by past work on bootleg scores [1]. This score representation is aligned with audio feature representations inspired by [2] using ordinary DTW.

porates these jump labels. We find that this approach can yield much higher-quality alignments than the automatic one, costing only seconds of annotator time.

We additionally extend the *bootleg score* feature representations used by Shan et al. [17], first proposed by Yang et al. [1]. Creating a bootleg score involves detecting noteheads and staff lines to produce a simple binary representation of a score that is conducive to alignment. We find that the use of measure bounding box detection as a preprocessing step improves the quality of underlying notehead and staff line detection algorithms. Additionally, motivated by findings in [2], we find that using raw onset probabilities predicted by a music transcription model as the audio feature representation produces higher quality alignments than using the MIDI transcriptions—see Figure 1 for a summary.

Motivated by our long-term goals of bringing sheet music into multimodal MIR, we also propose a new measure-aware evaluation scheme for comparing alignments. We speculate that measure-level alignment granularity is necessary for tractable training of multimodal MIR systems in the short term and that human perception of alignment quality is tied to measures. Accordingly, we prescribe new measure-aware alignment metrics for this task, such as an accuracy metric which reports the proportion of time where the estimated alignment is within a half measure radius of the ground truth alignment. On a small but diverse dataset of in-the-wild sheet music and aligned audio [18], we observe that our proposed system achieves an accuracy of 120% relative to that of Shan et al. (33% → 72%). By providing repeat labels, we improve the absolute accuracy of our system from 20% → 83% on a subset of pieces that have repeats. Our work makes the following contributions:

- A system capable of high-quality in-the-wild alignment of sheet music images and performance audio.
- A pragmatic workflow we call *Just Label The Repeats* that further improves alignment accuracy.
- An interface that enables rapid jump annotation.

2. TASK DESCRIPTION

Motivated by Thickstun et al. [20], here we formalize both the task of in-the-wild audio-to-score alignment and our proposed measure-aware evaluation. For a sheet music image with P pages (henceforth, a *score*), we define a *score playhead* (aligned position marker) as a tuple $(p, y, h, x) \in \mathcal{S} = \{0, \dots, P - 1\} \times [0, 1]^3$, where p is the page number, y and h are the offset and height of the current *system* (collection of staves) relative to the top edge and height of the page, and x is the playhead offset relative to the left edge of the page (see Figure 2). An analogous *audio playhead* is comparatively straightforward: a timestamp $t \in [0, T)$, where T is the audio length in seconds. An *alignment* is a mapping from audio to score playheads, i.e., $[0, T) \rightarrow \mathcal{S}$.

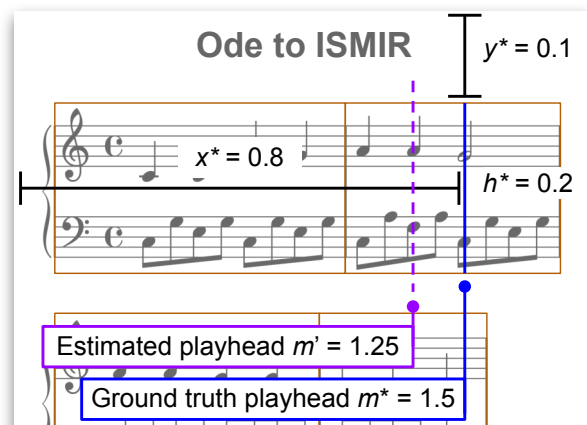


Figure 2. A score playhead (blue line), the output of an audio-to-score alignment, is characterized by its vertical offset (y), horizontal offset (x), and height (h), all relative to the page. A measure-aware alignment is indexed by m , a fractional measure, that can be converted to a score playhead by lookup and interpolation in a list of bounding boxes (brown outlines). Our measure-aware evaluation compares estimated playheads m' to ground truth m^* .

2.1 Measure-aware alignment

The above task definition is intended to be broad enough to encompass both past and future work on this task. Here we define a more specific form of alignment based around the location and ordering of *measures* in the score.

In this setting, measures are characterized by an ordered list of M bounding boxes $\mathbb{M} = [b_0, \dots, b_{M-1}]$, where $b_i = (b_i^p, b_i^y, b_i^h, b_i^x, b_i^w)$. Respectively, this tuple defines for each bounding box its page number, vertical offset, height, horizontal offset, and width. The ordering of this list is defined as the *logical* order that an expert would traverse when performing the piece—all *jumps* (repeat signs, Dal segno, etc.) are unrolled. For example, a score with 4 measures and a repeat implies that $M = 8$ and $b_i = b_{i+4}$.

Given a list of bounding boxes, a *measure-aware* score playhead can be characterized by a single continuous value $m \in [0, M)$, where $b_{\lfloor m \rfloor}$ is the bounding box of the current measure and fractional residual $m - \lfloor m \rfloor$ represents the offset from the left edge of the bounding box relative to its width. To convert a measure-aware score playhead in $[0, M)$ to an ordinary score playhead in \mathcal{S} , we define $h_{\mathbb{M}} : m \mapsto (b_{\lfloor m \rfloor}^p, b_{\lfloor m \rfloor}^y, b_{\lfloor m \rfloor}^h, b_{\lfloor m \rfloor}^x + b_{\lfloor m \rfloor}^w(m - \lfloor m \rfloor))$.

Given \mathbb{M} , a *measure-aware alignment* is a function $g : [0, T) \rightarrow [0, M)$. Because outputs of g index logical order (where jumps are unrolled), a measure-aware alignment g is a monotonically increasing function, i.e., $g(t_a) \leq g(t_b) \iff t_a \leq t_b$. Furthermore, we can compose $h_{\mathbb{M}}$ and g to induce an alignment that outputs score playheads, i.e., $h_{\mathbb{M}} \circ g : [0, T) \rightarrow \mathcal{S}$.

2.2 Measure-aware evaluation

Here we propose three measure-aware metrics for evaluating estimated alignments. **Our primary evaluation metric, MAcc, is defined as the proportion of time where the estimated score playhead is within a half measure radius of the ground truth score playhead**, which we posit is sufficiently precise for broader goals of multimodal MIR systems. ME_{err} and MD_{ev} are the mean and standard deviation (across time) of the absolute error between the estimated and ground truth playheads in units of ground truth measures.

More formally, given a ground truth measure-aware alignment g^* characterized by measures \mathbb{M}^* , and an estimated alignment g' characterized by \mathbb{M}' , we define:

$$\begin{aligned} \text{MDiff}(t) &= \text{Reindex}(g'(t), \mathbb{M}', \mathbb{M}^*) - g^*(t), \\ \text{MAcc} &\equiv \frac{1}{N} \sum_{i=0}^{N-1} \begin{cases} 1 & \text{if } |\text{MDiff}(\frac{Ti}{N})| \leq \frac{1}{2} \\ 0 & \text{otherwise.} \end{cases} \\ \text{MErr} &\equiv \frac{1}{N} \sum_{i=0}^{N-1} \left| \text{MDiff}\left(\frac{Ti}{N}\right) \right|, \\ \text{MDev} &\equiv \sqrt{\frac{1}{N} \sum_{i=0}^{N-1} \text{MDiff}\left(\frac{Ti}{N}\right)^2} \end{aligned}$$

We set $N = 100T$, i.e., we compute all metrics at a resolution of 100 comparisons per second. As an example, Figure 2 shows a single comparison where $\text{MDiff} = 0.25$.

The `Reindex` procedure ensures that the evaluation is based in units of ground truth measure indices, despite potential discrepancies between \mathbb{M}' and \mathbb{M}^* . Informally, this procedure matches each box in \mathbb{M}' to the box in \mathbb{M}^* that it is closest to in terms of Euclidean distance between mid-points, using the ordering from \mathbb{M}^* to break ties that occur for repeated measures.

3. SYSTEM DESCRIPTION

In this section, we detail our proposed system for in-the-wild audio-to-score alignment. At a high level, our method uses DTW to align piano roll-like feature representations extracted independently from the audio and score (Figure 1). Both representations are matrices where one axis corresponds to time (either in units of seconds or measures) and the other corresponds to 88 MIDI pitches from A0 to C8 (piano range). Specifically, score feature representations are matrices in $\{0, 1\}^{48M \times 88}$ (binary) where M is the number of measures in the score, and audio feature representations are matrices in $[0, 1]^{Tf_k \times 88}$ (continuous) for performance length T and frame rate $f_k = 31\text{Hz}$.

3.1 Score feature representation

Our proposed score feature representation is an extension of *bootleg scores* proposed by Yang et al. [1]. Extracting bootleg scores involves detecting noteheads and staff lines, and then combining this information into a binary matrix where a 1 encodes the presence of a notehead at a particular horizontal position on a particular staff line. Our system extends this representation in two ways: (1) we use measure detection as a preprocessing step and run notehead detection algorithms on segmented measure images instead of full pages, and (2) we translate notehead positions on staff lines into MIDI pitches before alignment.

Measure, notehead and staff line detection. We use the methods from [1] to detect noteheads and staff lines from score images. Instead of operating on full page images, we first segment pages into measures using the measure detection model from [21], and detect noteheads and staff lines on individual measure images resized to fixed dimensions. In preliminary experiments, we found that measure segmentation improved detection consistency—our intuition is that these methods are sensitive to absolute pixel sizes of noteheads and stafflines, and resizing individual measures to uniform size reduces variance in the sizes of these attributes across measures and pieces. Here we resize measures by resizing the smaller of their height and width to 900 pixels, preserving aspect ratio for the larger of the two. For each notehead, we retain its *bootleg location*, defined as its raw (pixel-wise) horizontal location within the measure, and its semantic (discrete) vertical position within the detected staff lines (e.g., an F and G natural in treble clef are one apart in their *staff positions*).

Piano rolls. We create a binary piano roll-like representation of the score using the bootleg locations of noteheads. Specifically, for each logical measure image index $k \in \{0, \dots, M-1\}$, we construct a binary matrix $S_k \in \{0, 1\}^{48 \times 88}$, where a 1 at row i and column j corresponds to a notehead with horizontal location $\frac{j}{48}$ relative to the

measure width, and its staff position converted to a pitch j . We pick a measure representation to have 48 rows to give sufficient resolution to a variety of rhythmic patterns and note durations. We concatenate $[S_0, \dots, S_{M-1}]$ together to form our final representation $S \in \{0, 1\}^{48M \times 88}$.

Converting staff position to pitch. A key obstacle is that, without key signature and clef information, the mapping from staff positions to MIDI pitches is ambiguous. If we had these attributes, we could simply look up the pitch associated with each staff line. However, we found existing OMR systems to have brittle support for detecting this information for in-the-wild sheet music—hence, our method does not assume that we have access to this information. Accordingly, we convert bootleg scores into piano rolls by simply assuming treble and bass clefs respectively when two staves are detected—if more or fewer staves are detected, we default to the treble clef—and the key of C major. Surprisingly, perhaps because of the global optimality of DTW, these assumptions lead to reasonable alignments even when they are incorrect. We note that ground truth key signature and clef information for each measure can be fed into our method to facilitate and improve this conversion, but we do not require it.

3.2 Audio feature representation

Our audio feature representation pipeline is comparatively simple. To compute it, we simply pass the audio through the Onsets and Frames piano transcription model [3]. Motivated by [2], we use the raw onset prediction probabilities from this model as our audio feature representation, which is a matrix in $[0, 1]^{Tf_k \times 88}$. Despite this transcription model being trained on piano, we find that its onset probabilities can yield reasonable alignments even for non-piano audio.

3.3 Alignment

Finally, we align the score representations in $\{0, 1\}^{48M \times 88}$ and audio representations in $\mathbb{R}^{Tf_k \times 88}$. We use the implementation of standard DTW from `librosa` [22] with default parameters: equal-weighted transitions $(1, 1)$, $(0, 1)$, and $(1, 0)$, and Euclidean distance to compute costs.

4. EXPERIMENTS

Here we detail our experiments, which center around comparing our proposed method to that of Shan et al. [17] on the MeSA-13 [18] and SMR [1] datasets using our proposed measure-level evaluation (see Section 2.2).

4.1 Datasets

We evaluate our approach and relevant baselines on two different datasets. The first is MeSA-13 [18], a dataset of 13 sheet music scans and corresponding real (i.e., not synthesized) performance audio. This dataset contains expert annotations of measure bounding boxes in logical order (\mathbb{M}^*) and the timestamp of every measure in the performance audio (we linearly interpolate between timestamps to get a continuous ground truth mapping g^*). While small, MeSA-13 has reasonable diversity in score typesetting, performance acoustics (two pieces feature instruments besides piano), and jumps (two pieces have repeats).

The second dataset is a subset of the Sheet MIDI Retrieval v1.0 (SMR) dataset [1]. The full dataset contains scanned scores from IMSLP for 100 solo piano pieces (none of which have jumps), corresponding MIDI performances synthesized as audio, and human annotations of measures per line and measure timestamps. Of notable absence are annotations of measure bounding boxes, which are required for our proposed evaluation. Accordingly, we detect measures [21] and discard pieces where the detections do not agree with annotations of measures per line—this leaves us with a subset of 60 pieces for evaluation. Henceforth, SMR refers to this subset.

4.2 Access to additional annotations

We primarily evaluate systems in an automatic setting where systems are only given the score and audio as input. Because our system can incorporate additional score annotations when available, we also evaluate in settings where our system has access to additional annotations from the ground truth, simulating workflows where experts are in the loop during alignment. Specifically, we explore settings where our method has access to ground truth repeat annotations (R), measure bounding boxes (M), and staff information (S)—clef and key signatures. We only evaluate in these settings on MeSA-13 where we have these labels.

4.3 Baselines

We compare the performance of our system, composed of our feature extraction pipeline and vanilla DTW, to that of [17]. In the latter system, the feature extraction pipeline uses bootleg scores and staff line detection on the score images to extract staff lines (referred to as segments). For audio features, a transcribed MIDI representation is obtained from the Onsets and Frames piano transcription model [3] which is then used to compute bootleg scores. Finally, Hierarchical DTW performs a segment-level alignment between score features and audio features while handling jumps and repeats that occur at segment boundaries, but not those within segments. Thus, we opt to evaluate this baseline approach at the measure level instead of the segment level (which also allows for comparison with our system) by converting the segment-level alignment to a measure-level one via an algorithm with several key steps.

We first use measure detection [21] to locate measures in each segment. Then, we map segment indices to measure indices based on the positions of detected measures. Finally, we turn the given alignment between audio timestamps and segment indices to one between audio timestamps and measure indices via linear interpolation.

We compare these two systems as proposed instead of comparing their components for two reasons. First, in [17] it is claimed the system performs segment-level alignment on pieces without repeat info; we aim to test this. Second, while Hierarchical DTW allows backwards jumps to prior segments, it only allows forward jumps to one segment past the last one seen. Using Hierarchical DTW at the measure level would limit possible forward jumps to only one measure past the last one observed, which is insufficient for realistic alignment tasks.

Dataset	Given	System	MAcc	MErr	MDev
M13	-	[17]	0.33	10.9	11.6
	-	Ours	0.72	1.9	3.7
	R [†]	Ours	0.82	0.4	0.2
	R,M	Ours	0.86	0.4	0.2
	R,M,S	Ours	0.88	0.3	0.2
M13 _R	-	[17]	0.17	23.6	10.2
	-	Ours	0.20	10.0	3.0
	R [†]	Ours	0.83	0.3	0.0
	R,M	Ours	0.93	0.2	0.0
	R,M,S	Ours	0.95	0.2	0.0
M13 _{NR}	-	[17]	0.36	8.6	10.2
	-	Ours	0.82	0.4	0.2
	R [†]	Ours	0.82	0.4	0.2
	R,M	Ours	0.85	0.4	0.3
	R,M,S	Ours	0.87	0.3	0.2
SMR	-	[17]	0.36	14.2	18.7
	-	Ours	0.82	2.9	18.8

Table 1. Evaluation on MeSA-13 (including subsets with Repeats and No Repeats) and SMR. Our method outperforms that of [17] across all datasets except the subset of MeSA-13 with no repeats. R[†] is our recommended setting where our method is given access to ground truth Repeats that require little time for humans to annotate—we observe limited gains from more time-consuming annotations of Measure bounding boxes and Staff metadata.

4.4 Results and discussion

In Table 1, we report the measure-level alignment metrics (Section 2.2) of our system in all four settings and the system of [17] across the MeSA-13 (M13) and SMR datasets. To emphasize the effect of jumps on alignment performance, we separately report performance on the subset of MeSA-13 pieces with and without repeats (M13_R and M13_{NR}, respectively).

We observe that our system outperforms that of [17] in the automatic setting across all datasets. The superior performance of our system over that of [17] is likely due to our refinements to feature representations. We also note here that the system of [17] is designed to work on line-level, therefore evaluating it using our measure-level metric yields lower accuracy than what was reported in [17].

Additionally, we observe that given repeats and ground truth measure annotations, our system’s performance improves by 22% relative (MAcc 0.72 → 0.88) on M13. However, we also observe a relative performance improvement of 14% (MAcc 0.72 → 0.82) using our system on the same dataset when we only pass in repeats. Given that repeats are much easier for humans to annotate than measure bounding boxes, we propose to have humans *just label the repeats* as a recommended tradeoff between alignment quality and annotator time. We also explore providing our system with measure-level key signature and clef

Representation	M13	SMR
Onset probabilities	0.88	0.82
Onset predictions	0.86	0.82
Frame probabilities	0.70	0.53
Frame predictions	0.66	0.51
MIDI	0.46	0.20

Table 2. Evaluation of measure-aware alignment accuracies (MAcc) achieved by different audio feature representations obtained from the Onsets and Frames piano transcription model [3] on MeSA-13 and SMR.

information, finding that this information only marginally improves performance relative to the default key and clef assumptions described in Section 3.1.

4.5 Different audio feature representations

Here we compare alternative audio feature representations by evaluating MAcc on M13 given all additional information (i.e., the R,M,S setting described in Section 4.2). While we primarily use raw onset prediction probabilities (Section 3.2), the Onsets and Frames model [3] provides other possibilities including onset predictions (thresholded probabilities), frame probabilities and predictions, and the postprocessed MIDI transcription converted to piano roll (see [3] for details). Table 2 shows that onsets consistently outperform frames as an alignment representation—our intuition is that onsets are more appropriate for our setting as the bootleg score representation does not encode note durations. Additionally, while transcribed MIDI is a common feature representation for music alignment, in our setting we find it to be the worst choice.

5. LABELING INTERFACE

Here we describe a web-based interface that we built to enable experts to quickly annotate jumps in scores (induced by repeat signs, D.S. al coda, etc.). Our experiments show large improvements in alignment quality given jump labels. Accordingly, we designed an interface to make this process efficient—experts can label jumps in a matter of seconds. We include videos demonstrating the end-to-end process and qualitative results of our proposed workflow for pieces outside our evaluation data.² In these videos, labeling jumps takes less than 6s per page on average.

Our interface features a unified workflow for jump annotation based on clicking the starting and ending measure of a jump (Figure 3). To enable this workflow, we first run measure detection [21] on the backend and visualize detected measures on the frontend as a user hovers over the score. Then, users can simply click on two different measures to set a jump—this simple unified workflow accommodates a long tail of jump glyphs. The interface also visualizes the logical order of the measures induced by the measure bounding boxes and any jumps the user has set (M from Section 2.1). Finally, a user can download the

² Video examples: <https://bit.ly/jltr-ismir2024>

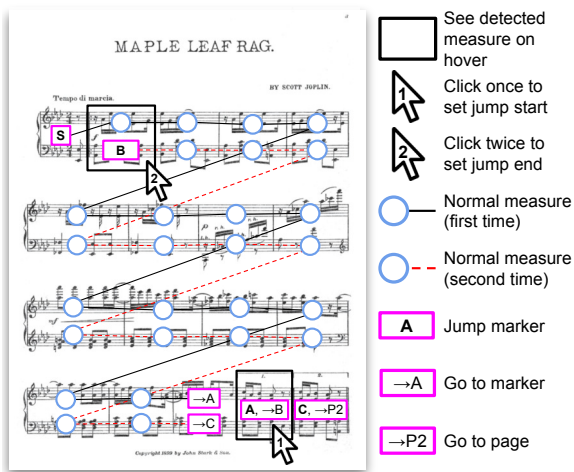


Figure 3. Illustration of our web-based interface for labeling jumps (e.g. repeats) in scores. Our interface enables rapid jump annotation (just seconds per page after training), which we find to dramatically improve alignment quality on pieces with jumps.

logical-order measures as a simple JSON file, which can then be loaded into our Python-based alignment package.

6. RELATED WORK

Our work relates to prior work in offline audio-to-score alignment, score following, and score annotation tools.

6.1 Offline alignment

Offline alignment entails building a correspondence between sheet music and performance recording of the contained music. As described previously, our work is inspired by the bootleg score line of research [1, 16, 17, 23, 24] as well as the usage of raw audio prediction scores in [2]. Our contributions stem from combining these approaches with minimal human annotations to outperform prior baselines. Other prior work in this area includes [15, 25], which use MIDI events as an intermediate representation between scores and audio, and [26–31], which use chroma vectors (where components correspond to the 12 pitch types in Western music). We perform a similar procedure to the methods used in these works except that we use bootleg scores as our intermediate representation. We diverge from [14], which uses MusicXML as an intermediate representation, and from [32], which uses LilyPond representations, but we mention them here as related approaches. We also incorporate reasoning from [20] regarding evaluation metrics.

6.2 Score following

In contrast to offline alignment, *score following* involves building a real-time alignment between sheet music and live performance audio. Initial solutions to this problem include [33, 34], with later works addressing jumps and repeats [35, 36]; for more on related work in past decades, see survey papers [37, 38]. Unlike most other research in score following which often assumes that a digital score

representation (like MIDI) is available, our emphasis is on *solely using score images* with an aim to perform alignment at scale. This said, some recent work does attempt to solve this problem in images. For instance, works such as [39–42] map audio snippets to corresponding places in score images using neural networks, but they are limited to piano music. We diverge from them by considering a range of different types of raw score images and audio (such as ones with instrumentation beyond solo piano) and leveraging bootleg scores [1, 16, 17, 23, 24] for mapping, but these are still related to our work.

6.3 Sheet music annotation interfaces

Our work also relates to past work on designing interfaces to assist in the annotation of sheet music. Most directly related is that of Feffer et al. [18] which attempts to facilitate interactive annotation of sheet music and audio alignment via a workflow based on aligning detected beat timestamps [43] to detected measures [21]. This interface was used to compile the MeSA-13 dataset of aligned audio and scores, which we use to evaluate our work, though we note that their interface required 20 hours of expert time to collect less than an hour of aligned data. In contrast, our interface is designed to be used for a matter of seconds to annotate repeats. Other interfaces focus on facilitating measure bounding box annotations [44, 45], which is complementary to our workflow that focuses on repeat annotation using predicted bounding boxes. Lastly, Soundslice [19] is a commercial product that offers an interactive alignment workflow based on stronger notions of OMR, but its implementation details are proprietary.

7. CONCLUSION

In summary, we introduce a workflow for efficiently aligning sheet music images to performance audio. The key insight we leverage is that while automated alignment algorithms are currently not robust to repeats in scores, humans can quickly label these repeats, thereby improving alignment performance. We validate this approach on a dataset of in-the-wild sheet music scans and real performance recordings, showing that we outperform existing baselines that only use automated approaches.

Given these results, one future project we aim to undertake is to collect jump annotations at scale to create large aligned datasets. We acknowledge that the datasets we used to evaluate our approach are small, but the insights gained from them can help scale up data for future evaluations. For instance, we could extend the interface from Section 5 to allow annotators to quickly audit and adjust alignments to collect more data, as in [18]. Moreover, other future work could revisit the creation of a fully automated alignment algorithm with insights from our work, namely that such an algorithm that leverages OMR to identify jumps and repeats may be more successful than one that does not. Collecting more data would therefore be helpful for developing and evaluating future approaches. Lastly, as described in the start of our paper, large aligned datasets could be used to derive multimodal MIR systems for music students and professionals alike.

8. ETHICS STATEMENT

As described previously, advancements in audio-to-score alignment can result in new multimodal datasets derived from existing repositories of sheet music and audio (such as IMSLP [46]) and new interactive music systems tailored for performance. Our motivation for pursuing this direction is to unlock multimodal MIR systems that (1) supplement music education by helping performers rehearse, (2) understand or generate sheet music to unlock seamless communication with human musicians, and (3) reduce reliance on copyrighted material for building music AI (i.e., by leveraging public domain scores and recordings).

We also recognize several potential ethical concerns stemming from our work. Firstly, our method is firmly rooted in conventions of Western music. Accordingly, downstream systems and data derived from our method may reflect a Western bias that does not generalize well to other musical traditions, especially those with different notation or tuning systems. Secondly, though our goal is to lessen the amounts of copyright infringement taking place to build generative music AI, multimodal MIR systems could be used to circumvent data protections, e.g., by transcribing copyrighted recordings as less-protected sheet music. Lastly, increased ability to understand sheet music could lead to deepfakes or misinformation, e.g., scores that could be falsely attributed to Beethoven, or ragtime recordings that could be falsely attributed to Joplin. In response to these concerns, we recommend that future work mitigate these risks by, for example, developing analogous systems capable of improving understanding of non-Western music notation. We also recommend that MIR researchers should be mindful of data protections, copyright violations, and artistic mimicry that, if subverted, could threaten the livelihood of musicians.

9. REFERENCES

- [1] D. Yang, T. Tanprasert, T. Jenrungrot, M. Shan, and T. J. Tsai, "Midi passage retrieval using cell phone pictures of sheet music," in *ISMIR*, 2019.
- [2] B. Maman and A. H. Bermano, "Unaligned supervision for automatic music transcription in the wild," in *International Conference on Machine Learning (ICML)*, 2022.
- [3] C. Hawthorne, E. Elsen, J. Song, A. Roberts, I. Simon, C. Raffel, J. Engel, S. Oore, and D. Eck, "Onsets and frames: Dual-objective piano transcription," *ISMIR*, 2018.
- [4] C. Donahue, J. Thickstun, and P. Liang, "Melody transcription via generative pre-training," in *ISMIR*, 2022.
- [5] J. Gardner, I. Simon, E. Manilow, C. Hawthorne, and J. Engel, "Mt3: Multi-task multitrack music transcription," in *International Conference on Learning Representations (ICLR)*, 2022.
- [6] R. M. Bittner, J. J. Bosch, D. Rubinstein, G. Meseguer-Brocal, and S. Ewert, "A lightweight instrument-agnostic model for polyphonic note transcription and multipitch estimation," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2022.
- [7] A. Agostinelli, T. I. Denk, Z. Borsos, J. Engel, M. Verzetti, A. Caillon, Q. Huang, A. Jansen, A. Roberts, M. Tagliasacchi *et al.*, "MusicLM: Generating music from text," *arXiv:2301.11325*, 2023.
- [8] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Défossez, "Simple and controllable music generation," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.
- [9] S.-L. Wu, C. Donahue, S. Watanabe, and N. J. Bryan, "Music controlnet: Multiple time-varying controls for music generation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2024.
- [10] J. J. Carabias-Orti, F. J. Rodriguez-Serrano, P. Vera-Candeas, N. Ruiz-Reyes, and F. J. Canadas-Quesada, "An audio to score alignment framework using spectral factorization and dynamic time warping," in *ISMIR*, 2015.
- [11] J.-L. Syue, L. Su, Y.-J. Lin, P.-C. Li, Y.-K. Lu, Y.-L. Wang, A. W. Su *et al.*, "Accurate audio-to-score alignment for expressive violin recordings," in *ISMIR*, 2017.
- [12] A. Arzt and S. Lattner, "Audio-to-score alignment using transposition-invariant features," in *ISMIR*, 2018.
- [13] T. Tanprasert, T. Jenrungrot, M. Müller, and T. Tsai, "Midi-sheet music alignment using bootleg score synthesis," in *ISMIR*, 2019.
- [14] F. Foscarin, A. Mcleod, P. Rigaux, F. Jacquemard, and M. Sakai, "Asap: a dataset of aligned scores and performances for piano transcription," in *ISMIR*, 2020.
- [15] M. Dorfer, A. Arzt, and G. Widmer, "Learning audio-sheet music correspondences for score identification and offline alignment," in *ISMIR*, 2017.
- [16] M. Shan and T. Tsai, "Improved handling of repeats and jumps in audio-sheet image synchronization," in *ISMIR*, 2020.
- [17] ———, "Automatic generation of piano score following videos." *Transactions of the International Society for Music Information Retrieval (TISMIR)*, 2021.
- [18] M. Feffer, C. Donahue, and Z. Lipton, "Assistive alignment of in-the-wild sheet music and performances," in *ISMIR Late Breaking Demo Track*, 2022.
- [19] Soundslice LLC, "Soundslice: Create living sheet music," <https://www.soundslice.com>, 2024, accessed on 2024-04-12.

- [20] J. Thickstun, J. Brennan, and H. Verma, “Rethinking evaluation methodology for audio-to-score alignment,” *arXiv preprint arXiv:2009.14374*, 2020.
- [21] S. Waloschek, A. Hadjakos, and A. Pacha, “Identification and cross-document alignment of measures in music score images,” in *ISMIR*, 2019.
- [22] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, “librosa: Audio and music signal analysis in Python,” in *SciPy*, 2015.
- [23] T. Tsai, D. Yang, M. Shan, T. Tanprasert, and T. Jen-rungrot, “Using cell phone pictures of sheet music to retrieve midi passages,” *IEEE Transactions on Multimedia*, 2020.
- [24] D. Yang, A. Goutam, K. Ji, and T. Tsai, “Large-scale multimodal piano music identification using marketplace fingerprinting,” *Algorithms*, 2022.
- [25] M. Dorfer, J. Hajic Jr, A. Arzt, H. Frostel, and G. Widmer, “Learning audio-sheet music correspondences for cross-modal retrieval and piece identification.” *Transactions of the International Society for Music Information Retrieval (TISMIR)*, 2018.
- [26] F. Kurth, M. Müller, C. Fremerey, Y.-h. Chang, and M. Clausen, “Automated synchronization of scanned sheet music with audio recordings,” in *ISMIR*, 2007.
- [27] D. Damm, C. Fremerey, F. Kurth, M. Müller, and M. Clausen, “Multimodal presentation and browsing of music,” in *Proceedings of the 10th international conference on Multimodal interfaces*, 2008.
- [28] C. Fremerey, M. Clausen, S. Ewert, and M. Müller, “Sheet music-audio identification.” in *ISMIR*, 2009.
- [29] V. Thomas, C. Fremerey, M. Müller, and M. Clausen, “Linking sheet music and audio-challenges and new approaches,” *Multimodal Music Processing*, 2012.
- [30] C. Fremerey, M. Müller, and M. Clausen, “Handling repeats and jumps in score-performance synchronization,” in *ISMIR*, 2010.
- [31] M. G. M. Gasser, A. Arzt, and G. Widmer, “Automatic alignment of music performances with structural differences,” in *ISMIR*, 2013.
- [32] L. Liu, V. Morfi, and E. Benetos, “Joint multi-pitch detection and score transcription for polyphonic piano music,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021.
- [33] R. B. Dannenberg, “An on-line algorithm for real-time accompaniment,” in *Proceedings of the International Computer Music Conference (ICMC)*, 1984.
- [34] B. Vercoe, “The synthetic performer in the context of live performance,” in *Proceedings of the International Computer Music Conference (ICMC)*, 1984.
- [35] B. Pardo and W. Birmingham, “Modeling form for on-line following of musical performances,” in *Proceedings of the National Conference on Artificial Intelligence*, 2005.
- [36] A. Arzt, G. Widmer, and S. Dixon, “Automatic page turning for musicians via real-time machine listening,” in *ECAI 2008*. IOS Press, 2008.
- [37] M. Puckette and C. Lippe, “Score following in practice,” in *Proceedings of the International Computer Music Conference (ICMC)*, 1992.
- [38] N. Orio, S. Lemouton, and D. Schwarz, “Score following: State of the art and new developments,” in *Proceedings of the New Interfaces for Musical Expression (NIME) Conference*, 2003.
- [39] M. Dorfer, A. Arzt, and G. Widmer, “Towards score following in sheet music images,” in *ISMIR*, 2016.
- [40] F. Henkel and G. Widmer, “Real-time music following in score sheet images via multi-resolution prediction,” *Frontiers in Computer Science*, 2021.
- [41] F. Henkel, R. Kelz, and G. Widmer, “Learning to read and follow music in complete score sheet images,” in *ISMIR*, 2020.
- [42] ———, “Audio-conditioned u-net for position estimation in full sheet images,” in *2nd International Workshop on Reading Music Systems*, 2019.
- [43] S. Böck, F. Korzeniowski, J. Schlüter, F. Krebs, and G. Widmer, “madmom: a new Python Audio and Music Signal Processing Library,” in *Proceedings of the 24th ACM International Conference on Multimedia*, 2016.
- [44] F. Zalkow, A. V. Corrales, T. Tsai, V. Arifi-Müller, and M. Müller, “Tools for semi-automatic bounding box annotation of musical measures in sheet music,” in *ISMIR Late Breaking Demo Track*, 2019.
- [45] E. Egozy and I. Clester, “Computer-assisted measure detection in a music score-following application,” in *Proceedings of the 4th International Workshop on Reading Music Systems, Online*, 2022.
- [46] “Imslp: Petrucci music library.” [Online]. Available: <https://imslp.org/>

INVESTIGATING TIME-LINE-BASED MUSIC TRADITIONS WITH FIELD RECORDINGS: A CASE STUDY OF CANDOMBLÉ BELL PATTERNS

Lucas S. Maia^{1,2} Richa Namballa³ Martín Rocamora^{4,5}
Magdalena Fuentes³ Carlos Guedes¹

¹ MaSC, New York University Abu Dhabi, UAE

² PEE/COPPE, Universidade Federal do Rio de Janeiro, Brazil

³ New York University, United States

⁴ Universitat Pompeu Fabra, Spain ⁵ Universidad de la República, Uruguay

{lucas.maia, richa.namballa}@nyu.edu

ABSTRACT

We introduce a series of transdisciplinary corpus studies aimed at investigating cross-cultural trends in time-line-based music traditions. Our analyses concentrate on a compilation of field recordings from the Centre de Recherche en Ethnomusicologie (CREM) sound archive. To demonstrate the value of an interdisciplinary approach combining ethnomusicology and music information research to rhythmic analysis, we propose a case study on the bell patterns used in the musical practices of Candomblé, an Afro-Brazilian religion. After removing vocals from the recordings with a deep learning source separation technique, we further process the instrumental segments using non-negative matrix factorization and select the bell components. Then, we compute a tempo-agnostic rhythmic feature from the bell track and use it to cluster the data. Finally, we use synthesized patterns from the musical literature about Candomblé as references to propagate labels to the rhythmic clusters in our data. This semi-supervised approach to pattern analysis precludes the need for downbeat and cycle annotations, making it particularly suited for extensive archive investigations. Lastly, by comparing bell patterns in Candomblé and a West African music tradition, we lay the foundation for our future cross-cultural research and observe the potential application of this methodology to other time-line-based music.

1. INTRODUCTION

Over the years, the music information retrieval/research (MIR) community has embraced more culturally-inclusive research geared towards the analysis of non-Western music [1]. Many of these studies have done extensive work to increase the representation of certain musical styles

from specific cultures [2]. However, few MIR initiatives have attempted to investigate multiple musical cultures simultaneously at a larger scale due to the level of difficulty and lack of data. These hardships reinforce a cycle of under-representation of various populations and risk further emphasizing the perspective of MIR through a Western-centric lens.

In this paper, we propose one of several studies aimed at a more global, inclusive, and transdisciplinary approach to MIR that puts humanistic (ethnomusicological and anthropological) and computational approaches in dialogue within a framework defined as Sonic Digital Humanities [3]. Our work centers on a substantial corpus of data from the Centre de Recherche en Ethnomusicologie (CREM). While we will explore only a part of this archive in this paper, we discuss its contents and their importance in Section 3. In Sections 4-7, we introduce a preliminary investigation on a subset of the archive analyzing bell patterns in Brazilian Candomblé and music from West Africa to demonstrate the potential this data has for future cross-cultural research on time-line-based acoustic traditions.

2. RELATED WORK

Ethnomusicology and MIR are often associated in a way that one is viewed as the source discipline while the other is the target [4]. However, researchers have suggested treating them as partners rather than as a hierarchy [5, 6]. With this approach, the MIR community can develop new computational methodologies which incorporate external information, such as cultural context, to better understand audio signals [1]. Specifically, we apply a framework known as the Sonic Digital Humanities (SDH) to our investigation. SDH is a branch of the Digital Humanities concerned with digital collections of music and other forms of sonic culture. It provides a space in which computational means to the analysis of sound culture may be developed and carried out in a productive dialogue with humanistic modes of data collection and critical inquiry [3].

By analyzing collections from an SDH perspective, we intend to ask questions about the cross-cultural relationships of different musical styles on a global scale. While engaging in data-driven analyses of these musical



styles, we attempt to understand (1) whether these findings support evidence collected from ethnomusicological studies about cross-cultural influences and (2) how these large-scale computational investigations can provide further insights into comparing music cross-culturally. In the case of (1), validating (or not) musicological evidence about cross-cultural relationships can improve MIR approaches to provide more reliable large-scale studies of lesser-known styles in the digital world. With (2), these methods can yield new findings of what characterizes a musical style born out of cross-cultural influence.

The first steps in expanding MIR beyond the exclusive sphere of Western music involved improving cultural representation in datasets. As a result, several culture- or style-specific corpora have been published for studies on music such as American ragtime, Beijing opera, and more [7, 8]. These advancements have improved access to a variety of data, but have yet to make a dent in the dominance of Western methodologies in MIR. A recent push by Huang et al. [9] for the MIR community to go beyond the collection of diverse data, collaborate with musicologists, and reflect on the way we engage with music serves as an appropriate objective for our studies of the CREM archive.

Despite the variety and richness of the information available in the CREM archive, very few studies have been published about this data. One MIR study utilizes the database to evaluate a proposed timbre classification method on a diverse set of musical instruments with the intent of allowing the indexing of ethnomusicological databases [10]. This sparseness of research presents major opportunities to explore the CREM archive in greater detail over a series of long-term, novel studies.

Our first investigation concerns the analysis of time lines, also known as bell patterns, in large collections of music. Time lines are short, cyclic patterns played in ostinato, often with a bell, castanet, or sticks [11, 12], that are used as a “controlling structural concept” [13, p. 1] in African music. This type of organization extends beyond geographical boundaries and can be heard in Afro-diasporic musical styles from the Caribbean or South America, for example. A key aspect of time lines is that they are qualitatively different from the concept of meter, as they originate from the movement of feet in dance [11], and denote a circularity that is characteristic of African music traditions [13]. This distinguishing factor calls for computational pattern recognition strategies that go beyond traditional methods of meter detection.

Toussaint proposed several mathematical methods, including geometric and graphical ones, for the analysis of clave-bell rhythm time lines [14]. Despite not being originally automated, they served as foundational work for future research concerning rhythmic complexity. Soon after, Toussaint continued their work on clave-bell time lines by comparing metrics for rhythmic similarity, such as Hamming distance and Euclidean interval vector distance [15]. All of the methods described in [14] and [15] require manual annotations, which are time-consuming and not scalable to a corpus as vast as the CREM archive.

Consequently, our proposed pipeline for time line pattern analysis is semi-supervised, with the majority of feature extraction and similarity computations automated. We draw inspiration from [16], who used template matching to track tempi of Afro-Cuban clave rhythms. However, their method has the drawback of requiring an exhaustive search for matching every tempo at each onset. Additionally, we consider the approach by [17], who inferred meter from Candombe recordings using rhythmic templates learned with the help of annotations. We improve upon these methodologies by using reference tracks to compute similarity measures based on [18]. The scale transform magnitudes (STM) [18] operate on the autocorrelation of the signal’s onset strength. They are robust to tempo variation, which facilitates the transfer of labels from the references to the tracks under study.

In this paper, we use Candomblé as a case study of time-line-based music. Candomblé is an Afro-Brazilian religion known for syncretically combining elements from many cultures, most notably Yoruba, Bantu, and Fon — which were brought to Brazil by enslaved West African populations [19]. Music plays a crucial role in the religious practices of Candomblé. Antiphonal songs are performed throughout the entire ceremony, accompanied by a drumming ensemble, always with the intent of allowing the participants and certain deities (*orixás*) to communicate [20]. Different rhythmic patterns, in both singing and drumming, are associated with different *orixás*. There are a few historical collections of Candomblé field recordings in the CREM archive, serving as a valuable resource of audio data for our investigation.

3. CREM-NYUAD COLLECTION

The CREM database is an extensive archive of digitized audio recordings from cultures around the world. Spanning from the beginning of the 20th century to today, the archive contains over 48,000 field recordings and more than 17,000 published commercial recordings representing over 1300 ethnic groups across 199 countries. The public has access to rich metadata cataloguing the database as well as thousands of recordings available to listen to for free on the archive’s website.¹

Through a partnership with the Centre National de la Recherche Scientifique (CNRS), New York University Abu Dhabi (NYUAD) has acquired a subset of the CREM archive for the purpose of analyzing the sound recordings. Henceforth, we call this subset the CREM-NYUAD collection. The CREM-NYUAD collection consists of 14,379 records from 129 countries, with a majority coming from Africa, Asia, and South America. In particular, Vietnam, Nepal, Madagascar, Gabon, and Algeria are among the countries with the most records in the dataset. Each item consists of audio features, such as spectrograms and tempograms, extracted during a prior collaboration between CREM and NYU. The associated metadata for each record contains basic data about the item, such as the collec-

¹ <https://archives.crem-cnrs.fr>

tion name and date, but also includes valuable information about the location, language, instrumentation, and ethnographic context of the recording. This information provides a significant advantage in our pilot study of Candomblé bell patterns, as we will see in Sections 4-7.

An important feature of our collection, which distinguishes it from many other datasets used in MIR research, is the prevalence of field recordings in the corpus. Field recordings provide important cultural context through the settings in which the traditions are recorded, such as in the *terreiros* (places of worship) of Candomblé [21]. In contrast to commercial studio tracks, field recordings are often taken in natural conditions where there are various social and environmental sounds, as well as noise, in the final recording [22, 23]. Furthermore, the time span over which these field recordings are collected often reflects the technological progress of the time period with more recent recordings producing higher quality audio. The acoustic diversity of the collection poses additional challenges to our computational methods in the form of silence, noise, and artifacts (e.g., clicks). We attempt to overcome some of these obstacles to the analysis in our pipeline, but save any audio restoration endeavors for future work.

4. BELL PATTERNS IN CANDOMBLÉ

The drumming ensemble in Candomblé typically consists of three differently-sized drums called *atabaques*, a dried gourd covered in beads known as *xequerê*, and a single or double clapperless bell called *gan* or *agogô*. Figure 1 shows ten essential Candomblé bell patterns as notated in [24]. These motifs were identified as the main patterns utilized by the Ketu nation, the largest branch of the Candomblé religion today. For example, pattern 1 represents the bell part in *vassi*, which is a common pattern in many different rhythms of Candomblé and is performed by bells with two accompanying drums. Pattern 2 is the same as the Son clave [14]. Pattern 3, known as *ijexá*, gained popularity in not only religious contexts, but also in the festive *Carnaval* parades held in Salvador, Bahia.

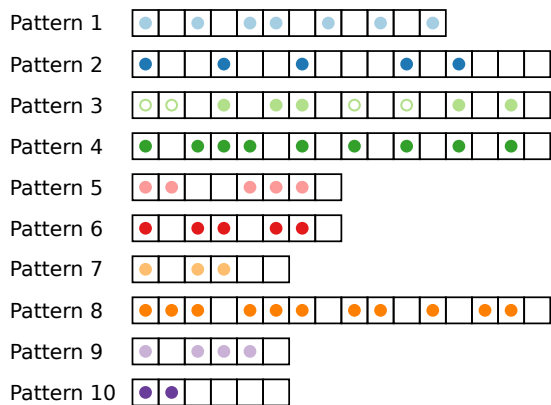


Figure 1: Examples of Candomblé bell patterns in time unit box system (TUBS) notation [24]. Open and closed dots indicate the use of high- and low-pitched bells.

5. METHODOLOGY

The process we employ to extract rhythmic features associated with bell patterns in field recordings is encapsulated in Figure 2, and explained further in this section.

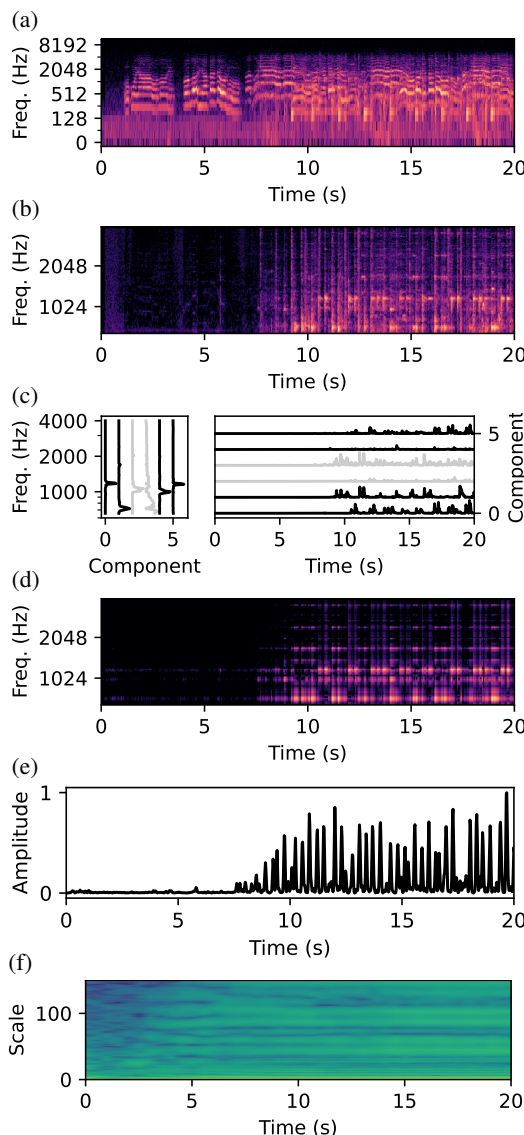


Figure 2: Workflow for extracting rhythmic features for bell patterns: (a) spectrogram of the original track (vocals and instruments); (b) the isolated instrumental part, from 650 to 4000 Hz; (c) NMF-learned templates and activations (highlighting those selected via the spectral crest); (d) reconstructed spectrogram of the bell part; (e) smooth bell activation function; and (f) frame-wise STM.

5.1 Source Separation

In order to analyze the bell patterns in greater detail, we need to isolate them from the remainder of the audio track. To do so, we first remove the vocals from each track due to their potential obstruction of the bells’ frequency band. Using the pre-trained state-of-the-art hybrid transformer source separation model, Demucs [25], we preserve the instrumentals by separating the vocal stem.

To further abstract the bell patterns from the rest of the rhythm sections, we decompose the non-vocal track with

non-negative matrix factorization (NMF) [26]. First, we resample the non-vocal signal to 8000 Hz. Then, we compute its short-time Fourier transform (STFT) with a 64-ms window and a hop size of 20 ms. We restrict the NMF analysis from 650 to 4000 Hz, which discards the main frequencies from low-pitched drums. At this time, we assume that the signal contains primarily the bell tones (sometimes in two distinct pitches, when both bells of the *agogô* are used) and noise-like components emanating from the *xéquerê* or the other drums’ attacks. For this reason, we run the NMF algorithm with $n = 6$ components. After the algorithm converges, we use the spectral crest [27] to identify sources corresponding to the bells by selecting components whose templates are more tonal in nature. The geometric mean of all crest factors serves as a threshold. Finally, we reconstruct the separated spectrogram of the bell part with the dot product of the matrices composed by the selected template–activation pairs.

5.2 Feature Extraction

The next step in our workflow is computing a rhythmic feature based on the scale transform magnitudes [18]. We first compute the time derivative of the log-compressed reconstructed source spectrogram, using a factor of compression $C = 1000$ as in [28]. All of the bins are summed up, and we apply half-wave rectification to keep only positive peaks. To smooth this accent signal, we use a lag of 3 frames in the computation of the time-difference [29] and further process the signal by convolving it with a Gaussian kernel ($\sigma = 20$ ms). Finally, we follow the procedure of [18] and determine the local autocorrelation of the accent signal with an 8-second moving rectangular window (hop size of 0.5 s). The direct scale transform [30] converts the autocorrelation at each frame into the scale domain, such that tempo is not encoded in the representation, and we keep only the first 150 scale coefficients. We discard all frames at the start of the signal whose energy lies below a threshold of -60 dB. Feature vectors can be compared using cosine similarity or Euclidean distance [18], with the former being better suited for handling changes in level between the recordings.

5.3 Label Propagation

We follow a semi-supervised procedure to classify patterns in the dataset. For this purpose, we create synthetic versions of the patterns in Figure 1 with no accents or timing deviations. We extract the rhythmic patterns of these synthesized reference tracks using the same pipeline as before. The only differences are that we use all $n = 3$ NMF components to generate a single activation and that we summarize the STM feature by taking the average along the time axis. Lastly, we propagate labels to the original (unlabeled) dataset in the following fashion:

1. For each track i in the dataset, we measure the maximum pairwise distance, σ_i , between STM frames;
2. For each frame j of track i , we find the closest data

point, r_k , from the reference set, such that the distance $d(x_{ij}, r_k)$ is minimal;

3. If $d(x_{ij}, r_k) \leq \sigma_i$, x_{ij} receives the same label as r_k , else it receives a “null” label;
4. “Winner-take-all”: we perform plurality voting among all labels for x_i where the most prevalent label is used to represent the entire track.

While the labels can be propagated within the feature space, this process can also be intuitively performed in a lower-dimension embedding space.

6. EXPERIMENTS AND RESULTS

We select a specific set of tracks from the CREM-NYUAD collection on which to run our entire pipeline. We rely on the metadata described in Section 3 to identify which files contain bell sounds. Table 1 shows the countries in West Africa (and Brazil) with these bell patterns and the number of recordings from each country.

Country	# Records
Congo-Brazzaville	98
Brazil	71
Benin	42
Angola	30
Gabon	27
Mali	24
Côte d’Ivoire	11

Table 1: Number of records with bell patterns per country.

In this initial experiment, we further restrict our scope by selecting, from those files recorded in Brazil, a set of recordings by ethnographer (and Candomblé initiate) Pierre Verger.² Moreover, with the assumption that bells mostly establish a cyclic pattern, we consider only the first 60 s of each recording. Next, we proceed with the analysis from Section 5; i.e., after our pre-processing steps, we extract the rhythmic features for all tracks in the subset and in the reference set. We then perform label propagation from the reference set to the subset. Using UMAP [31], a manifold learning technique, we present the results in Figure 3.

With regards to structure, our pipeline clearly extracts meaningful information from the rhythmic patterns in the subset, as many distinct clusters are visible. Interestingly, we observe that the reference patterns are well distributed among these clusters.

The label propagation procedure also reveals important aspects of the data distribution. For instance, we notice that the approach labels a large portion of the frames as belonging to the “pattern 1” archetype. Furthermore, note how closely patterns 7 and 9 are represented in the embedding. This proximity is easily explained by them differing on only a single beat (see again Figure 1). The families of patterns 2 and 5 also appear near each other in the manifold, but this time their pattern lengths are unequal. However, by “interpolating” pattern 5 and cyclically rotating

²https://archives.crem-cnrs.fr/archives/collections/CNRSMH_I_2007_011/

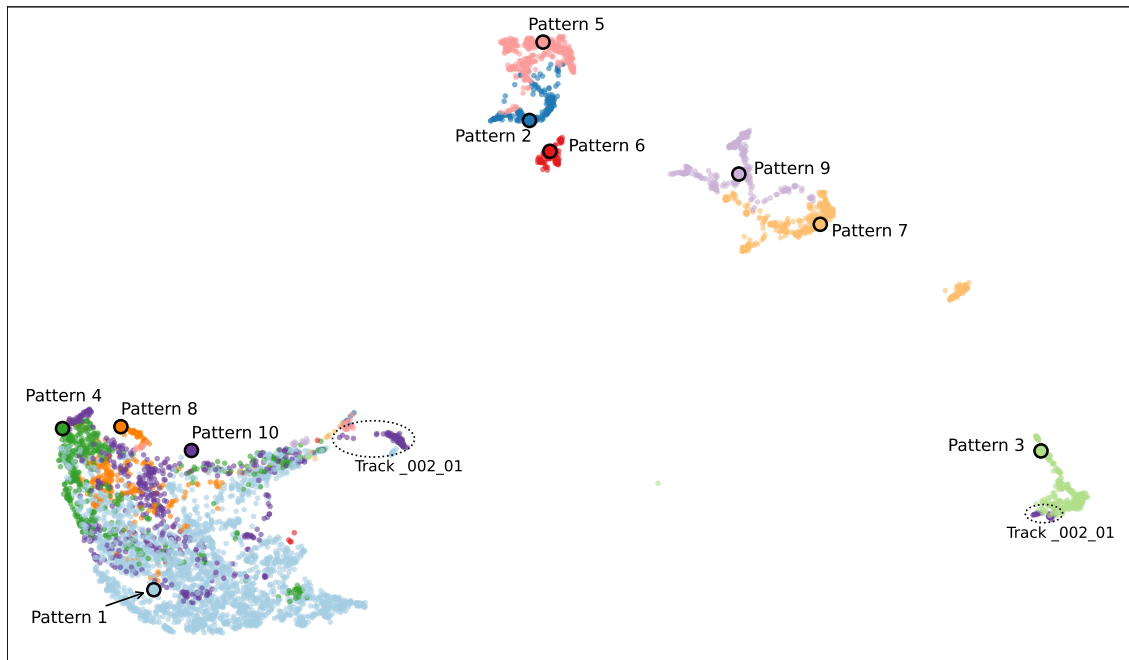


Figure 3: UMAP projection of the Candomblé bell patterns and reference patterns (cosine metric, n -neighbors = 70, min-dist = 0.1). The same color coding of Figure 1 is used here. Circled: the frames corresponding to track `_002_01`.

pattern 2 (Figure 4), we see that they are more alike than they appear at the surface level, especially considering that the STM feature is independent of tempo. A similar argument explains the close proximity of clusters for patterns 5 and 6.

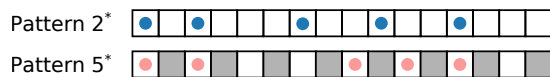


Figure 4: Interpolation and cyclical rotation of patterns 5 and 2 reinforce the similarities found in the embedding. Three of the five onsets match, and the remaining two are a slight shift away from their counterparts.

We can also assess the label propagation method’s performance by investigating some information from the metadata. For example, recordings `_002_01`, `_004_03`, `_005_04`, and `_005_05` contain the indication “*ijexá* rhythm” (“*rhythme ijexá*”) in their titles, so presumably they correspond to the same type of bell pattern as pattern 3. Consequently, in the embedding, their frames are clustered and labeled together as part of the “pattern 3” archetype. The only exception is recording `_002_01` (circled), which was incorrectly classified as pattern 10 and divided into two sections: the majority of frames are situated in the easternmost region of the large 1–4–8–10 cluster, while a handful of remaining frames are found near pattern 3. In this case, the misclassification could be attributed to the crest selection procedure’s inability to retrieve the main component of the highest-pitched bell.

Figure 5 showcases selected examples of the onset activations from the subset, juxtaposed with their corresponding reference activations. To ensure alignment, we manually adjusted the references’ timing to match the excerpts.

Differences between the references and audio realizations, such as additional or missing notes, can be ascribed to recording conditions or introduced by the pipeline. Despite these discrepancies, the workflow demonstrates robustness, as evidenced by the confirmation of the automatic classification through listening tests. Pattern variations can also originate from the player, who may miss a note or add embellishments (flams). Another type of discrepancy we have identified, resulting from small scale deviations, is illustrated with recording `_004_09`.

Lastly, we conduct another analysis which uses a larger number of recordings with bell patterns from the Republic of the Congo (see Table 1). We follow the same procedure as before, but lower the minimum frequency for the NMF decomposition from 650 to 300 Hz, since West African bells are typically larger and lower pitched. Figure 6 displays the embedding of both the Brazilian and Congolese patterns from our subset. This visualization shows similarities between some of the patterns; these potential cross-cultural intersections require further investigation. In particular, with a “plurality voting” procedure similar to our label propagation scheme, we can detect that recording `_030_03` from collection `CNRSMH_I_1974_013`³ is the most akin to the patterns of the Brazilian recordings. A short listening test confirms that the bell in this recording performs a rhythmic pattern similar to that of pattern 1 (the most common one in the Brazil subset).

7. DISCUSSION

We emphasize two important consequences of our study. Both observations emerge from our attempt to balance

³https://archives.crem-cnrs.fr/archives/items/CNRSMH_I_1974_013_030_03/

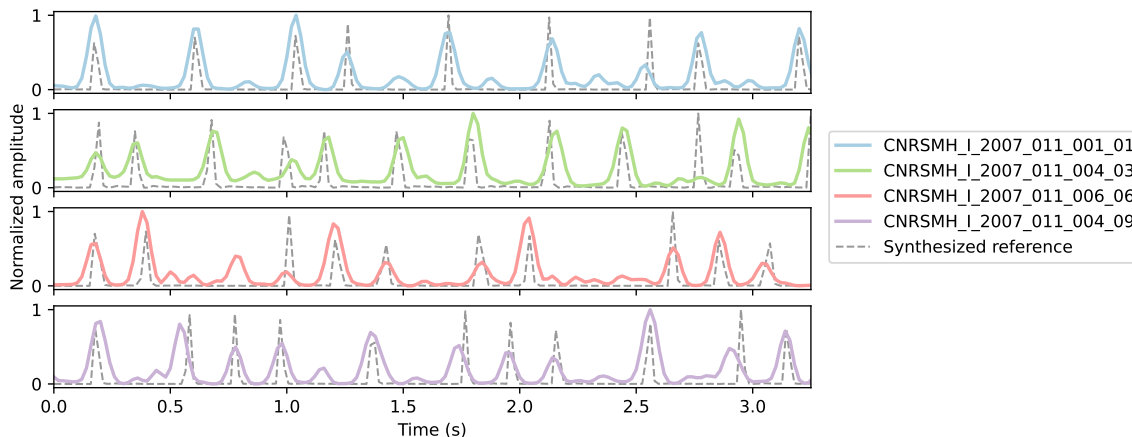


Figure 5: Excerpts of onset activations and their corresponding references. The same color coding of Figure 1 is used here.

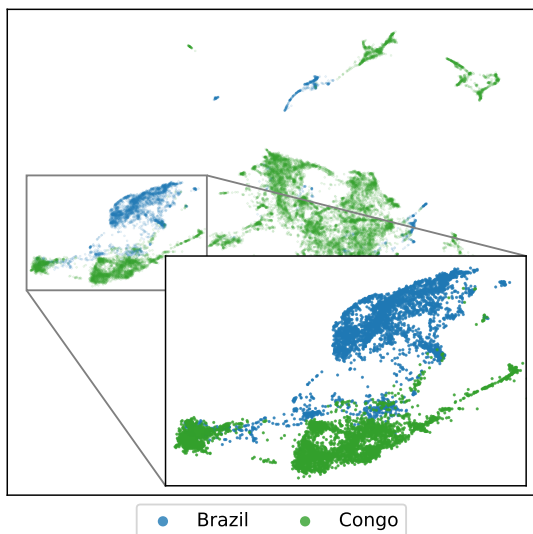


Figure 6: Embedding of rhythmic features from Brazilian and Congolese bell patterns in the subset.

data-driven methodologies with more humanistic perspectives using the SDH framework.

Firstly, the non-negative matrix factorization presents a significant bottleneck in our analysis. Ideally, we would use source separation to isolate the bells from the track directly. However, the out-of-the-box Demucs model is trained only to separate 4-6 stems of specific Western instruments. A blind application of this model often results in an unpredictable placement of the bell patterns, as they may end up in the drum stem for one track, but in the “other” category for another. This inconsistency underscores the need for a more culturally-inclusive, tailored approach, such as the few-shot source separation model proposed in [32], or even a new perspective on source separation as a task. Improvements such as these could offer a more accurate and consistent separation of the bells, enhancing the rhythmic salience.

Secondly, despite the technical challenges produced by field recordings, our feature extraction pipeline has proven to be remarkably robust. Most importantly, it respects the unique characteristics of West African and Afro-diasporic music, particularly the concept of rhythmic cycles. Time

lines are qualitatively different from the concept of meter as a temporal hierarchical grouping mechanism and serve culture-specific purposes depending on the context of their use. While they can be mapped into meters due to their cyclic or recurrent nature, they often play “against” their metrical grid [12]. Anku [13] suggests that they should be perceived as a “circular concept” rather than a linear one, allowing performers to seamlessly enter and exit the performance with little inhibition. Our pipeline, which makes no assumptions regarding the notions of meter or downbeat and uses no annotations, was designed to respect these unique characteristics. The only attribute we infer is the cyclic nature of the rhythmic patterns to ease our computational load. Our minimal suppositions demonstrate the capacity of our methodology to expand to other styles of cyclic music, beyond what is studied in this paper.

8. CONCLUSION

We presented a pilot study investigating bell patterns in Candomblé from historical field recordings in a subset of the CREM archive, the CREM-NYUAD collection. Our approach is a preliminary venture in following the Sonic Digital Humanities (SDH) framework, to address the inherent challenges and complexities of applying computational methods to musical traditions which have been underrepresented in MIR. SDH aims to combine computational and ethnographic approaches in a dialogue on the same plane, while embracing any tensions which may entail in an agonistic fashion to push the traditional boundaries of interdisciplinarity [6].

Our study was influenced by the distinct characteristics of West African and Afro-diasporic music. Without requiring meter annotations, we could detect and classify patterns from the collection using a robust and adaptable pipeline, despite encountering challenging recording conditions and unique rhythmic structures. Our code is available at <https://github.com/nyuad-masc/crem-time-lines>. Future work will focus on addressing shortcomings and further refining our methods (e.g., source separation) to improve the analysis of time-line-based music traditions and allow the study of cross-cultural influences.

9. ETHICS STATEMENT

Our highest priority in this study is to respect the cultural heritage and to protect the privacy of the communities represented in the CREM-NYUAD collection. We recognize that every culture and tradition has nuances which cannot be captured by computational methods alone and should not be subjected to reductions or generalizations. Any analyses implemented using the materials from this corpus are solely for academic purposes in an effort to increase cultural diversity by establishing a dialogue between the humanities and music information research.

10. ACKNOWLEDGEMENTS

The authors would like to thank Mark Cartwright for his prior work extracting audio features from the CREM database, which serves as a crucial foundation for this and future research. We also thank Joséphine Simonnot, whose efforts were instrumental in the curation and acquisition of the CREM-NYUAD collection.

11. REFERENCES

- [1] X. Serra, "A multicultural approach in music information research," in *Proceedings of the 12th International Society for Music Information Retrieval Conference*, Miami, USA, Oct. 2011, pp. 151–156.
- [2] —, "Creating research corpora for the computational study of music: the case of the CompMusic project," in *Proceedings of the AES 53rd International Conference on Semantic Audio*, London, UK, Jan. 2014, pp. 1–9.
- [3] A. Eisenberg and C. Guedes, "Prolegomena for sonic digital humanities," in *British Forum for Ethnomusicology & International Council for Traditional Music Ireland Joint-Annual Conference*, Cork, Ireland, Apr. 2024.
- [4] K. Neubarth, M. Bergeron, and D. Conklin, "Associations between musicology and music information retrieval," in *Proceedings of the 12th International Society for Music Information Retrieval Conference*, Miami, USA, Oct. 2011, pp. 429–434.
- [5] P. Proutskova, "Musical memory of the world-data infrastructure in ethnomusicological archives," in *Proceedings of the 8th International Conference on Music Information Retrieval*, Vienna, Austria, Sep. 2007, pp. 161–162.
- [6] G. Born, "Diversifying MIR: Knowledge and real-world challenges, and new interdisciplinary futures," *Transactions of the International Society for Music Information Retrieval*, vol. 3, no. 1, p. 193–204, 2020.
- [7] A. Volk and W. B. de Haas, "A corpus-based study on ragtime syncopation," in *Proceedings of the 14th International Society for Music Information Retrieval Conference*, Curitiba, Brazil, Nov. 2013, pp. 163–168.
- [8] R. Caro Repetto and X. Serra, "Creating a corpus of jingju (Beijing opera) music and possibilities for melodic analysis," in *Proceedings of the 15th International Society for Music Information Retrieval Conference*, Taipei, Taiwan, Oct. 2014, pp. 313–318.
- [9] R. Huang, A. Holzapfel, B. Sturm, and A.-K. Kaila, "Beyond diverse datasets: Responsible MIR, interdisciplinarity, and the fractured worlds of music," *Transactions of the International Society for Music Information Retrieval*, vol. 6, no. 1, pp. 43–59, 2023.
- [10] D. Fourer, J.-L. Rouas, P. Hanna, and M. Robine, "Automatic timbre classification of ethnomusicological audio recordings," in *Proceedings of the 15th International Society for Music Information Retrieval Conference*, Taipei, Taiwan, Oct. 2014, pp. 295–300.
- [11] K. Agawu, *The African Imagination in Music*. Oxford University Press, 2016.
- [12] —, "The metrical underpinnings of African timeline patterns," 2013. [Online]. Available: <https://www.youtube.com/watch?v=8ypTYNGLr5A>
- [13] W. Anku, "Circles and time: A theory of structural organization of rhythm in African music," *Music Theory Online*, vol. 6, no. 1, pp. 1–8, 2000.
- [14] G. T. Toussaint, "A mathematical analysis of African, Brazilian, and Cuban clave rhythms," in *Bridges: Mathematical Connections in Art, Music, and Science*, 2002, pp. 157–168.
- [15] —, "A comparison of rhythmic similarity measures," in *Proceedings of the 5th International Conference on Music Information Retrieval*, Barcelona, Spain, Oct. 2004.
- [16] M. Wright, W. A. Schloss, and G. Tzanetakis, "Analyzing Afro-Cuban rhythms using rotation-aware clave template matching with dynamic programming," in *Proceedings of the 9th International Society for Music Information Retrieval Conference*, Philadelphia, USA, Sep. 2008, pp. 647–652.
- [17] L. Nunes, M. Rocamora, L. Jure, and L. W. P. Biscaíno, "Beat and downbeat tracking based on rhythmic patterns applied to the Uruguayan Candombe drumming," in *Proceedings of the 16th International Society for Music Information Retrieval Conference*, Málaga, Spain, Oct. 2015, pp. 246–270.
- [18] A. Holzapfel and Y. Stylianou, "Scale transform in rhythmic similarity of music," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 1, pp. 1766–185, 2011.
- [19] R. Bastide, *As Américas negras: as civilizações africanas no Novo Mundo*. São Paulo: Difel/Edusp, 1974.

- [20] P. A. McGrath-Kerr, “A cannibalist’s manifesto: Candomblé rhythms for drum kit,” Ph.D. dissertation, Faculty of Fine Arts and Music, University of Melbourne, 2019.
- [21] W. W. Megenny, “Afro-Brazilian percussion instruments: Etymologies & uses,” *Revista del CESLA. International Latin American Studies Review*, no. 9, pp. 25–35, 2006.
- [22] M. Marolt, “Probabilistic segmentation and labeling of ethnomusicological field recordings,” in *Proceedings of the 10th International Society for Music Information Retrieval Conference*, Kobe, Japan, Oct. 2009, pp. 75–80.
- [23] P. Proutskova and M. A. Casey, “You call that singing? ensemble classification for multi-cultural collections of music recordings,” in *Proceedings 10th of the International Society for Music Information Retrieval Conference*, Kobe, Japan, Oct. 2009, pp. 759–764.
- [24] H. Schroy and B. Reis, “A orquestra do Candomblé da nação Ketu,” 2011. [Online]. Available: <https://www.youtube.com/watch?v=kUgdwkBD-P8>
- [25] S. Rouard, F. Massa, and A. Défossez, “Hybrid transformers for music source separation,” in *Proceedings of the 2023 IEEE International Conference on Acoustics, Speech and Signal Processing*, Rhodes Island, Greece, Jun. 2023, pp. 1–5.
- [26] D. Lee and H. S. Seung, “Algorithms for non-negative matrix factorization,” in *Advances in Neural Information Processing Systems*, vol. 13, Denver, USA, 2000, pp. 556–562.
- [27] G. Peeters, “A large set of audio features for sound description (similarity and classification) in the CUIDADO project,” IRCAM, Tech. Rep., 2004.
- [28] P. Grosche and M. Muller, “Extracting predominant local pulse information from music recordings,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 6, pp. 1688–1701, 2011.
- [29] S. Böck and G. Widmer, “Maximum filter vibrato suppression for onset detection,” in *Proceedings of the 16th International Conference on Digital Audio Effects*, Maynooth, Ireland, 2013.
- [30] W. J. Williams and E. J. Zalubas, “Helicopter transmission fault detection via time-frequency, scale and spectral methods,” *Mechanical Systems and Signal Processing*, vol. 14, no. 4, pp. 545–559, 2000.
- [31] L. McInnes, J. Healy, N. Saul, and L. Großberger, “UMAP: Uniform manifold approximation and projection,” *The Journal of Open Source Software*, vol. 3, no. 29, p. 861, 2018.
- [32] Y. Wang, D. Stoller, R. M. Bittner, and J. Pablo Bello, “Few-shot musical source separation,” in *Proceedings of the 2022 IEEE International Conference on Acoustics, Speech and Signal Processing*, Singapore, Singapore, May 2022, pp. 121–125.

Author Index

Acosta, Elizabeth 281
 Adi, Yossi 146, 264
 Affolter, Joanne 547
 Agarwal, Rajat 1046
 Ai, Hsin 555
 Akama, Taketo 240
 Alben, Noel 705
 Alfaro-Contreras, María 462, 596
 Alluri, Vinoo 1046
 Alunno, Marco 1060
 Anslow, Michael 625
 Antonisen, Silas 688
 Aouameur, Cyran 272
 Araz, Recep Oguz 478
 Argüello, Giulia 405
 Arifi-Müller, Vlora 248
 Audebert, Nicolas 580

 Barnett, Julia 360
 Barthet, Mathieu 713
 Batlle-Roca, Roser 1004
 Bemman, Brian 389
 Benetos, Emmanouil 103, 661, 669, 825
 Berendes, Hans-Ulrich 248
 Berg-Kirkpatrick, Taylor 874
 Beyer, Tim 319
 Bientinesi, Paolo 1060
 Bindi, Giovanni 1037
 Bittner, Rachel 897
 Bjare, Mathias Rose 922
 Blang, Frederik 1012
 Bogdanov, Dmitry 478, 825
 Born, Georgina 745
 Bryan, Nicholas J. 874
 Buisson, Morgan 207
 Bukey, Irmak 1085
 Burgoyne, John Ashley 397, 791
 Caillon, Antoine 1020

 Calvo-Zaragoza, Jorge 225, 462, 596, 914
 Cancino-Chacón, Carlos Eduardo 1068
 Cantil, Ben 882
 Carr, CJ 429
 Carvalho, Hugo T. 618
 Chandran, Sharat 834
 Chang, Yi-Hsin 555
 Chen, Bo-Yu 634
 Chen, Haonan 1029
 Chen, Jitong 454
 Chen, Ke 335
 Chen, Tsung-Ping 189
 Chen, Yu-Hua 446
 Cheng, Hao-Chung 311, 634
 Cheng, Yuan-Chiao 446
 Choi, Keunwoo 946
 Chowdary, Pavani B. 1046
 Christensen, Justin 389
 Cífka, Ondřej 737
 Collins, Tom 154
 Comunità, Marco 420, 661
 Condit-Schultz, Nathaniel 705, 989
 Copet, Jade 146
 Cornia, Nicholas 486

 Dai, Angela 319
 Dang, Tommy 281
 Défossez, Alexandre 146
 Delgado-Sánchez, Teresa 225
 Demerlé, Nils 721
 Demetriou, Andrew M. 137
 Dendukuri, Krishna Maneesha 1020
 Deng, Jiajun 343, 493
 Deng, Qixin 669
 Deruty, Emmanuel 78
 Desblancs, Dorian 327
 Devis, Ninon 1012
 D’Hooge, Alexandre 520
 Dixon, Simon 288, 511, 611
 Doh, Seungheon 946
 Donahue, Chris 680, 1085

- Dong, Hao-Wen 588
Doras, Guillaume 721
Du, Xingjian 70, 1029
Duan, Zhiyao 182, 973

Edwards, Andrew C. 288, 611
Ehmann, Andreas F. 295
Eremenko, Vsevolod E. 520
Esling, Philippe 721, 1037
Essid, Slim 198, 207
Evans, Nicholas 540
Evans, Zach 429

Fan, Fan 493
Fang, Kun 996
Fazekas, George 111, 563, 571, 825
Feffer, Michael 1085
Flexer, Arthur 697
Forment, Bruno 486
Foscarin, Francesco 503, 962
Freeman, Michael 680
Fu, Jie 669
Fuentes, Magdalena 1093
Fujinaga, Ichiro 996
Fukayama, Satoru 86
Fukuda, Seikoh 381
Fukuda, Yuko 381

Gagneré, Antonin 198
Gao, Chenyu 154
García, Hugo Flores 360
García-Iasci, Patricia 225
Garoufis, Christos 95
Gat, Itai 264
Genova, David 721
Georgieva, Elena 232
Géré, Léo 580
Ghinassi, Iacopo 164
Glytsos, Marios 95
Gómez, Emilia 1004

Gómez-Marín, Daniel 540
González-Barrachina, Pedro 596
Gotham, Mark R. H. 217
Goto, Masataka 128, 529, 1076
Grachten, Maarten 272
Green, Owen 745
Groves, Ryan 397
Guan, Shuen-Huei 555
Guedes, Carlos 1093
Guinot, Julien PM 571
Guo, Yike 103, 669
Guo, Zixun 611

Hadjeres, Gaëtan 625
Haki, Behzad 540
Hamasaki, Masahiro 128
Hammoudeh, Ahmad 890
Han, Danbinaerin 217
Haseeb, Muhammad Taimoor 890
Haseyama, Miki 470
Henkel, Florian 295
Hennequin, Romain 327, 954
Herremans, Dorien 858
Ho, Yu-Hsiang 446
Holzapfel, Andre 371
Hosoda, Masamichi 381
Hsiao, Wen-Yi 311
Hu, Patricia 1068
Huang, Cheng-Zhi Anna 1020
Huang, Jingyue 335
Huang, Yipeng 669

Ibnyahya, Ilias 661
Ikemiya, Yukara 420
Imort, Johannes 1012
Ive, Julia 164

Jääskeläinen, Petra 371
Jang, Jyh-Shing Roger 446
Jeong, Dasaem 217, 588
Jiang, Junyan 783

- Jiang, Yue 866
 Jordà, Sergi 540
 Ju, Yaolong 343, 493
 Jung, Jiye 799
 Jung, Jongmin 588

 Kaila, Anna-Kaisa 371
 Kalimeri, Kyriaki 164
 Karystinaios, Emmanouil 503, 651
 Kato, Jun 529
 Kaye, Robert 413
 Kim, DongMin 217
 Kim, Halla 799
 Kim, Haven 634
 Kim, Hyon 304
 Kim, Hyunjae 352, 842
 Kim, Jaehun 137, 295
 Kim, Jeounghoon 799
 Kim, Taesoo 946
 Knees, Peter 850
 Kong, Qiuqiang 1029
 Kong, Yuexuan 954
 Koops, Hendrik Vincent 807
 Kreuk, Felix 264
 Krishnan, Venkatakrisnan Vaidyanathapuram 705
 Kwon, Daeyong 946

 Lagrange, Mathieu 954
 Lan, Yun-Han 311
 Landau, Camilo 295
 Lanzendörfer, Luca A. 405
 Lattner, Stefan 111, 272, 625, 922
 Lee, Jin Ha 397, 529
 Lee, Kyung Myun 352, 842
 Lee, Sihun 217
 Leger, Rebecca 397
 Lenz, Julian 981
 Lerch, Alexander 1051
 Li, Bochen 1029
 Li, Min Susan 618
 Li, Xiaobing 642
 Li, Yizhi 669
 Liang, Huidong 70
 Liang, Jinhua 511
 Liang, Xia 70
 Liao, Wei-Hsiang 1004
 Liem, Cynthia 137
 Lin, Chenghua 669
 Lin, Hanfeng 669
 Lin, Liwei 783
 Liu, Mingyu 70
 Liu, Xubo 669
 Liu, Xunying 343
 López-Espejo, Iván 688
 Lorenzo, Betty Cortiñas 493
 Lostanlen, Vincent 954
 Loth, Jackson J. 713
 Lu, Wei-Tsung 454
 Luca, Massimiliano Di 618
 Lui, Simon 343, 493
 Luna-Barahona, Noelia N. 462

 Ma, Yinghao 103, 669
 Maia, Lucas S. 1093
 Mak, Simon 866
 Malandro, Martin E. 438
 Maman, Ben 120
 Manco, Ilaria 825, 938
 Mangal, Natasha 397
 Mani, Anirudh 981
 Manolios, Sandy 137
 Maragos, Petros 95
 Martinez-Sevilla, Juan Carlos 914
 Marták, Lukáš Samuel 1068
 Matsubara, Masaki 381
 McAuley, Julian 874
 McFee, Brian 207, 232
 Melechovsky, Jan 858
 Meseguer-Brocal, Gabriel 327, 954
 Micchi, Gianluca 807

Miner, Luke 737
Mitsufuji, Yuki 420, 1004
Morris, Lidia J. 397, 529
Motomura, Ami 381
Moussallam, Manuel 327
Müller, Meinard 120, 173, 248
Muluneh, Mequanent Argaw 729

Nair, Anish A. 705
Nakamura, Eita 503
Nakamura, Tomohiko 86
Nakano, Tomoyasu 1076
Nakatsuka, Takayuki 128
Nam, Juhan 799, 946
Namballa, Richa 1093
Narang, Jyoti 256
Nercessian, Shahan 1012
Newman, Michele 397, 529
Nguyen, Ngan V.T. 281
Ni-Hahn, Stephen 866
Nieto, Oriol 938
Niitsuma, Masahiro 381
Nistal, Javier 272
Novack, Zachary 874
Nuttall, Thomas 61

Ogata, Jun 86
Ogawa, Takahiro 470
Oh, Eun Ji 352
Ohri, Kartik 413
Özer, Yigitcan 248

Pan, Jiahao 669
Parada-Cabaleiro, Emilia 520
Pardo, Bryan 360
Park, Hannah 217
Park, Juyong 799
Park, Saebyul 799
Park, Seokbeom 842
Parker, Julian D. 429

Pasini, Marco 111, 272
Pauwels, Johan 897
Peeters, Geoffroy 198, 625
Peng, Yan-Tsung 729
Peter, Silvan 603
Plaja-Roglans, Genís 61
Pons, Jordi 429
Pouw, Charlotte 791
Preniqi, Vjosa 164

Qiu, Lin 930
Quinton, Elio 571, 807, 825

Ramos, Yannis 53
Ramoneda, Pedro 240, 520
Rao, Preeti 834
Reiss, Joshua D. 563, 661
Ren, Zeng 53
Reuben, Federico 154
Rigaux, Philippe 580
Riley, Xavier 288, 611
Riou, Alain 625
Ripollés, Pablo 232
Rizo, David 225, 462, 914
Rocamora, Martín 61, 240, 1093
Roebel, Axel 146
Rohrmeier, Martin A. 53, 547
Rolland, Jean-Baptiste 563
Roselló, Adrián 462
Rouard, Simon 146
Roy, Abhinaba 858
Roychowdhury, Sujoy 834
Rudin, Cynthia 866
Ryu, Jiwoo 588

Sailor, Malcolm 814
Saitis, Charalampos 164
Saito, Koichi 420
Sakurai, Keigo 470
Salamon, Justin 938
Sandberg, Samuel E. 295

- Sarmiento, Pedro Pereira 288, 713
 Sasao, Eri 381
 Schlüter, Jan 962
 Schreiber, Hendrik 737
 Sears, David 281
 Serra, Xavier 61, 256, 304, 478, 520, 1004
 Shankar, Adithi 61
 Shibuya, Takashi 420
 Shikarpur, Nithya Nadig 1020
 Singh, Bhavyajeet 1046
 Singh, Shubhr 661
 Smith, Jordan B. L. 1029
 Smith, Noah A. 930
 Sowula, Robert 850
 Spijkervet, Janne 1029
 Steinmetz, Christian J. 563, 661
 Strahl, Sebastian 173
 Sturm, Bob L. T. 745
 Stöter, Fabian-Robert 248, 737
 Su, Li 729
 Suda, Hitoshi 86
 Sun, Chen 680
 Sun, Maosong 642

 Takahashi, Akira 420
 Takahashi, Shusuke 420
 Tal, Or 264
 Tamer, Nazif Can 256
 Tan, Chih-Pin 555
 Taylor, Josiah 429
 Tian, Zeyue 669
 Togo, Ren 470
 Tralie, Christopher J. 882
 Tsai, Fang Duo 634

 Vanka, Soumya Sai 563
 Vasilakis, Yannis 897
 Vásquez, Marcel A. Vélez 791
 Vega, Viviana De La 256

 Wald-Fuhrmann, Melanie 745
 Wan, Yanming 930
 Wang, Ju-Chiang 454, 1029
 Wang, Lu 103
 Wang, Wenwu 669
 Wang, Yashan 642
 Wang, Yi 669
 Wang, Zijie 70
 Wang, Ziyu 996
 Watcharasupat, Karn N. 1051
 Wattenhofer, Roger 405
 Weck, Benno 825
 Wei, Megan 680
 Wei, Weixing 906
 Widmer, Gerhard 503, 603, 651, 922, 962, 1068
 Wing, Alan M. 618
 Wong, Stella 954
 Wu, Chun Yat 493
 Wu, Jui-Te 446
 Wu, Shangda 642
 Wu, Shih-Lun 634
 Wu, Yuhang 103
 Wu, Yulun 906
 Wu, Yusong 1020
 Wu, Zhiyue 103

 Xia, Guangyu 669
 Xia, Gus 783, 890, 996
 Xu, Weihai 866
 Xue, Wei 103, 669

 Yan, Yujia 973
 Yang, Guang 930
 Yang, Jing 343, 493
 Yang, Qikai 669
 Yang, Shiqi 420
 Yang, Yi-Hsuan 311, 335, 446, 555, 634
 Yeh, Yen-Tung 446
 Yin, Zirui 866
 Yip, Jason 529
 Yoshida, Shunsuke 86

Yoshii, Kazuyoshi 189, 906

Yu, Feng 642

Yu, Huiran 182

Yuan, Ruibin 103, 669

Yuan, Shanxin 661

Zehren, Mickael 1060

Zeitler, Johannes 120

Zhang, Ge 669

Zhang, Huan 511

Zhang, Muru 930

Zhang, Xinyue 103

Zhang, Yixiao 783

Zhao, Jiahao 906

Zhao, Mengjie 420

Zhong, Zhi 420

Zhou, Ziya 103

Zhu, Bilei 70

Zhu, Jinlong 470

Zhu, Rico 866

Ziv, Alon 264

Zlatintsi, Athanasia 95

Zou, Pei 70, 1029

Zuidema, Willem 791

Zukowski, Zachary 429