24th International Society for Music Information Retrieval Conference

November 5-9, 2023
Milan, Italy

Proceedings

ISMIR 2023 was organized by Politecnico di Milano, Fondazione Politecnico di Milano, the International Society for Music Information Retrieval, and a diverse international committee of organizers.

Website: https://ismir2023.ismir.net

ISMIR 2023 logo design: Fabio Antonacci
Cover page design: Luca Comanducci

Edited by:
Augusto Sarti *(Politecnico di Milano, Italy)*
Fabio Antonacci *(Politecnico di Milano, Italy)*
Mark Sandler *(Queen Mary University of London, UK)*
Paolo Bestagini *(Politecnico di Milano, Italy)*
Simon Dixon *(Queen Mary University of London, UK)*
Beici Liang *(Nomono, Norway)*
Gaël Richard *(Télécom Paris, France)*
Johan Pauwels *(Queen Mary University of London, UK)*

ISBN: 978-1-7327299-3-3

Title: Proceedings of the 24th International Society for Music Information Retrieval Conference, Milan, Italy, Nov 5-9, 2023.

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.

© 2023 International Society for Music Information Retrieval

ISBN 978-1-7327299-3-3
9 781732 729933

ISMIR Organizers

POLITECNICO
MILANO 1863

Fondazione
Politecnico
di Milano

ISMIR

ISMIR Sponsors

Platinum Sponsors

moises

Gold Sponsors

YAMAHA

musixmatch

algoriddim

deezer

Adobe

Google Research

UNIVERSAL

UNIVERSAL MUSIC GROUP

Silver Sponsors

ACRCloud

steinberg

NATIVE INSTRUMENTS

SiriusXM

WIMIR Sponsors

Patrons

deezer

Contributors

moises

Google Research

NATIVE INSTRUMENTS

SiriusXM

Supporters

steinberg

Organizing Team

Conference Chairs

General Chairs

Augusto Sarti *(Politecnico di Milano, Italy)*
Fabio Antonacci *(Politecnico di Milano, Italy)*
Mark Sandler *(Queen Mary University of London, UK)*

Scientific Program Chairs

Paolo Bestagini *(Politecnico di Milano, Italy)*
Simon Dixon *(Queen Mary University of London, UK)*
Beici Liang *(Nomono, Norway)*
Gaël Richard *(Télécom Paris, France)*

Publications Chair

Johan Pauwels *(Queen Mary University of London, UK)*

Tutorial Chairs

Johanna Devaney *(Brooklyn College, CUNY, USA)*
Emmanouil Benetos *(Queen Mary University of London, UK)*

Late-Breaking / Demo Chairs

Massimiliano Zanoni *(Politecnico di Milano, Italy)*
George Fazekas *(Queen Mary University of London, UK)*

Jam Session Chairs

François Pachet *(Spotify, France)*
Mark d'Inverno *(Goldsmiths, University of London, UK)*

Music and Tech Chairs

Matthew Yee King *(Goldsmiths, University of London, UK)*
Cristina Rottondi *(Politecnico di Torino, Italy)*

Sponsorship/Industry Chairs

Ilaria Manco *(Queen Mary University of London, UK)*
Luca Andrea Ludovico *(Università degli Studi di Milano, Italy)*

Satellite Events Chair

Federico Avanzini *(Università degli Studi di Milano, Italy)*

Women in MIR / Diversity and Inclusion Chairs

Claire Arthur *(Georgia Institute of Technology, USA)*
Xiao Hu *(University of Hong Kong, Hong Kong)*
Francesca Ronchini *(Politecnico di Milano, Italy)*

Virtual Technology, Social Media and Website Chairs

Marco Olivieri *(Politecnico di Milano, Italy)*
Luca Comanducci *(Politecnico di Milano, Italy)*
Mirco Pezzoli *(Politecnico di Milano, Italy)*
Clara Borrelli *(Apple, UK)*

Volunteering & Logistics Chair

Riccardo Giampiccolo *(Politecnico di Milano, Italy)*

Social program & Local Organization Chair

Alberto Bernardini *(Politecnico di Milano, Italy)*

Unconference Chair

Geoffroy Peeters *(Télécom Paris, France)*

Volunteers

Chiara Auriemma
Li Baichen
Olga Besedova
Lei Bian
Arunava Bose
Andrea Bosisio
Hyeyoon Cho
Karolayne Dessabato
Jacopo Fantin
Raísa Farias Silveira
Tirna Ghosh
Marcello Grati
Emanuele Greco
Julien Guinot
Dilip Harish
Thế Hoàng Nguyễn
Xenofon Karakonstantis
Hyon Kim
Gopika Krishnan
Alice Lenoci
Xinmeng Luan
Teng Ma
Francesco Maccarini
Carlo Macrí
Mayur Mankar
Noemi Mauri
Ivan Meresman Higgs
Eray Özgünay
Silvia Pasin
Riccardo Passoni
Matteo Pettenò
Adriane Replogle
Davide Rizzotti
André Santos
Surabhi Shinde
Haokun Song
Dario Sorce
Maksim Stepanov
Mattia Surricchio
Camillo Trujillo
Marco Viviani
I-Chieh Wei
Shijie Yang
Yuchen Zhang
Jian Zhou

Program Committee

Meta-Reviewers

Vinoo Alluri, International Institute of Information Technology Hyderabad
Vipul Arora, Indian Institute of Technology Kanpur
Claire Arthur, Georgia Institute of Technology
Andreas Arzt, Apple
David Bainbridge, University of Waikato
Christine Bauer, Paris Lodron University Salzburg
Juan P. Bello, New York University
Emmanouil Benetos, Queen Mary University of London
Rachel Bittner, Spotify
Dmitry Bogdanov, Universitat Pompeu Fabra
Juan J. Bosch, Spotify
John Ashley Burgoyne, University of Amsterdam
Rafael Caro Repetto, Kunstuniversität Graz
Kahyun Choi, Indiana University Bloomington
Keunwoo Choi, Gaudio Lab Inc.
Chris Donahue, Stanford University
Stephen Downie, University of Illinois Urbana-Champaign
Zhiyao Duan, Unversity of Rochester
Sebastian Ewert, Spotify
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
Nobutaka Ito, University of Tokyo
Ozgur Izmirli, Connecticut College
Katherine M. Kinnaird, Smith College
Peter Knees, Technische Universität Wien
Sri Rama Murty Kodukula, Indian Institute of Technology Hyderabad
Audrey Laplante, Université de Montréal
Jin Ha Lee, University of Washington
Alexander Lerch, Georgia Institute of Technology
Florence Levé, Université de Picardie Jules Verne
Cynthia C. S. Liem, Delft University of Technology
Brian McFee, New York University
Cory McKay, Marianopolis College
Andrew McLeod, Fraunhofer IDMT
Hema A. Murthy, Indian Institute of Technology Madras
Meinard Müller, International Audio Laboratories Erlangen
Juhan Nam, Korea Advanced Institute of Science and Technology
Oriol Nieto, Adobe Research
Mitsunori Ogihara, University of Miami
Johan Pauwels, Queen Mary University of London
Geoffroy Peeters, Télécom Paris
Jordi Pons, Stability AI
Preeti Rao, Indian Institute of Technology Bombay
Justin Salamon, Adobe Research
Xavier Serra, Universitat Pompeu Fabra
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
Douglas Turnbull, Ithaca College
George Tzanetakis, University of Victoria
Peter Van Kranenburg, Utrecht University/Meertens Institute
Gabriel Vigliensoni, Concordia University
Cheng-i Wang, AudioShake
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

Reviewers

Jakob Abeßer
Pablo Alonso-Jiménez
Lior Arbel
Stefan Balke
Berker Banar
Adrián Barahona-Ríos
Mathieu Barthet
Dogac Basaran
Gilberto Bernardes
Louis Bigo
Otso Björklund
Clara Borrelli
Paul Brossier
Dan Brown
Bryony Buck
Marcelo Caetano
Jorge Calvo-Zaragoza
Pavel Campr
Mark Cartwright
Francisco J. Castellanos
Sungkyun Chang
Bo-Yu Chen
Ke Chen
Yu-Hua Chen
Tian Cheng
Ching-Yu Chiu
Shreyan Chowdhury
Graham K. Coleman
Nathaniel Condit-Schultz
Guillem Cortès
Laura Cros Vila
Helena Cuesta
Shuqi Dai
Roger B. Dannenberg
Matthew Davies
Timothy R. de Reuse
Reinier de Valk
Alessio Degani
Christian Dittmar
Hao-Wen Dong
Simon Durand
Matthias Eichner
Vsevolod E. Eremenko
Jianyu Fan
Xavier Favory
Andres Ferraro
Flavio Figueiredo
Christoph Finkensiep
Frederic Font
Francesco Foscarin
Dominique Fourer
Klaus Frieler
Diego Furtado Silva
Nick Gang
Kaustuv Kanti Ganguli
Chenyu Gao
Roman B. Gebhardt
Elena Georgieva
François G. Germain
Riccardo Giampiccolo
Mark R. H. Gotham
Niccolo Granieri
Carlos Guedes
Chitralekha Gupta
Siddharth Gururani
Ranjani H G
Gaëtan Hadjeres
Jan Hajič jr.
Ben Hayes
Florian Henkel
Peyman Heydarian
Katharina Hoedt
Jiawen Huang
Yu-Fen Huang
Chris Hubbles
Yun-Ning Hung
Karim M. Ibrahim
Charles Inskip
Berit Janssen
Dasaem Jeong
Junyan Jiang
Yucong Jiang
Zeyu Jin
Yaolong Ju
Maximos Kaliakatsos-Papakostas
Jaehun Kim
Jong Wook Kim
Minje Kim
Phillip B. Kirlin
Qiuqiang Kong
Radha Manisha Kopparti
Amanda E. Krause
Michael Krause
Kosmas Kritsis
Frank Kurth
Taegyun Kwon
Pierre Laffitte
Stefan Lattner
Jongpil Lee
Mark Levy
David Lewis
Bochen Li
Yizhi Li
WeiHsiang Liao
Kin Wah Edward Lin
Yuan-Pin Lin
Lele Liu
Yi-Wen Liu
Antoine Liutkus
Patricio López-Serrano
Hanna Lukashevich
Akira Maezawa
Lucas S. Maia
Ethan Manilow
Sandy Manolios
Leandro Balby Marinho
Matija Marolt
Benjamin Martin
David Martins de Matos
Matthias Mauch

Rudolf Mayer
Shuxin Meng
Gianluca Micchi
Remi Mignot
Ronald Mo
Hyeongi Moon
Fabio Morreale
Alia Morsi
Manuel Moussallam
Lucas N. Ferreira
Tomoyasu Nakano
Marco Olivieri
Patricio Ovalle
Yigitcan Özer
Vinutha T. P.
Emilia Parada-Cabaleiro
So Yeon Park
Ashis Pati
Miguel Perez Fernandez
Antonio Pertusa
Matevž Pesek
Pedro D. Pestana
Silvan Peter
Genís Plaja-Roglans
Lorenzo Porcaro
Laure Prétet
Zafar Rafii
Antonio Ramires
David Rizo
Martín Rocamora
Axel Roebel
Gerard Roma
Iran R. Roman
Sebastian Rosenzweig
Joe Cheri Ross
Pedro Pereira Sarmento
Maximilian Schmitt
Hendrik Schreiber
Simon J. Schwär
Sertan Şentürk
Micael A. Silva
Federico Simonetta
Anup Singh
George Sioros
Christian J. Steinmetz
Daniel Stoller
Fabian-Robert Stöter
Vinod Subramanian
Michael Taenzer
Nazif Can Tamer
Jingjing Tang
Tiago F. Tavares
Marko Tkalcic
Timothy Tsai
Kosetsu Tsukuda
Alexandra Uitdenbogerd
Finn Upham
Jose J. Valero-Mas
Aneesh Vartakavi
Makarand Velankar
Amruta Vidwans
Venkata S. Viraraghavan
Changhong Wang
Chung-Che Wang
Ju-Chiang Wang
Jun-You Wang
Yu Wang
Ziyu Wang
Kento Watanabe
Benno Weck
I-Chieh Wei
David M. Weigl
Christopher W. White
Gordon Wichern
Scott Wisdom
Daniel Wolff
Minz Won
Kyle J. Worrall
Chih-Wei Wu
Shih-Lun Wu
Anna Xambó
Furkan Yesiler
Sangeon Yong
Minjoon Yoo
Johannes Zeitler
Huan Zhang
Yixiao Zhang
Yudong Zhao
Yi Zhong

Preface

Welcome to ISMIR 2023, the 24th International Society for Music Information Retrieval Conference. ISMIR is the world's leading research forum on processing, searching, organizing, and accessing music-related data. Our community reflects a diversity of scientific disciplines, seniority levels, professional affiliations, and cultural backgrounds. We aim to foster and stimulate this diversity, leading to better science and better music services. The organizing team, who came together from all over the world to ensure the success of this event, welcomes you to ISMIR 2023.

Scientific Program

The ISMIR 2023 scientific program comprised three keynote talks, six tutorials and 103 papers. A total of 272 abstracts were registered on the submission system, of which 229 were submitted as complete papers eligible for review. In keeping with the practices of the previous years, a two-tier double-blind review process was conducted involving a total of 211 reviewers and 63 meta-reviewers. Each paper was assigned to a single meta-reviewer and 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 4 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 metareviewers and reviewers. 104 papers were accepted (one of which was later withdrawn by the authors), giving an **acceptance rate of 45.4%** (or 38.2% if incomplete submissions are included). The SPC would like to express their thanks to the ISMIR community of reviewers and metareviewers for their wholehearted support of this critical aspect of a successful ISMIR technical program.

Table 1 summarizes the number of submitted 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.

Table 1: Papers submitted and accepted by subject area

Subject Area	**Submitted**	**Accepted**	**Accepted %**
MIR tasks	77	21	20.2%
Musical features and properties	42	17	16.3%
Knowledge-driven approaches to MIR	37	18	17.3%
Applications	35	11	10.6%
MIR fundamentals and methodology	22	10	9.6%
Evaluation, datasets, reproducibility	19	7	6.7%
Human-centered MIR	19	8	7.7%
Computational musicology	12	6	5.8%
MIR and ML for musical acoustics	5	3	2.9%
Philosophical and ethical discussions	4	3	2.9%
Total	**272**	**104**	

Table 2: Summary of publication statistics over the 24-year-history of the ISMIR conference

Year	Location	Oral	Poster	Total	Authors	Unique Authors	Authors / Paper	Unique Authors / Paper
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

Best Paper Awards

Awards for the best paper and best student paper were given at ISMIR 2023. Best paper candidates were selected from the 104 accepted papers. The SPC selected six candidate papers based on reviewers' and meta-reviewers' nominations as well as the paper review scores and comments. The final selections were made by specially appointed judges drawn from experienced MIR researchers who had no conflict of interest with any of the award candidates. In addition, judges from Universal Music Group selected one paper from a longer list of highly ranked papers on the basis of its contribution to responsible MIR research, for a special Responsible Research Award, sponsored by UMG.

The following papers were nominated for consideration for the Best Paper Awards (in order of paper number):

- *Exploring the Correspondence of Melodic Contour with Gesture in Raga Alap Singing*, Shreyas M Nadkarni, Sujoy Roychowdhury, Preeti Rao and Martin Clayton
- *CLaMP: Contrastive Language-Music Pre-training for Cross-Modal Symbolic Music Information Retrieval*, Shangda Wu, Dingyao Yu, Xu Tan and Maosong Sun
- *BPS-Motif: A Dataset for Repeated Pattern Discovery of Polyphonic Symbolic Music*, Yo-Wei Hsiao, Tzu-Yun Hung, Tsung-Ping Chen and Li Su
- *PESTO: Pitch Estimation with Self-Supervised Transposition-Equivariant Objective*, Alain Riou, Stefan Lattner, Gaëtan Hadjeres and Geoffroy Peeters
- *LP-MusicCaps: LLM-Based Pseudo Music Captioning*, Seungheon Doh, Keunwoo Choi, Jongpil Lee and Juhan Nam
- *Singing Voice Synthesis Using Differentiable LPC and Glottal-Flow-Inspired Wavetables*, Chin-Yun Yu and George Fazekas

Each of the Best Paper candidates will be invited to publish an extended version of their paper in the Transactions of the International Society for Music Information Retrieval (TISMIR), the open access journal of the Society. The Society will cover the article processing charges of these publications. The following three awards were given:

Best Paper Award *PESTO: Pitch Estimation with Self-Supervised Transposition-Equivariant Objective*, Alain Riou, Stefan Lattner, Gaëtan Hadjeres and Geoffroy Peeters

Best Student Paper Award *CLaMP: Contrastive Language-Music Pre-training for Cross-Modal Symbolic Music Information Retrieval*, Shangda Wu, Dingyao Yu, Xu Tan and Maosong Sun

UMG Award for Responsible Research *Data Collection in Music Generation Training Sets: A Critical Analysis*, Fabio Morreale, Megha Sharma and I-Chieh Wei

Diversity & Inclusion

The ISMIR 2023 conference took a broad view of Diversity, Equity, and Inclusion (DEI). We considered two "sides" of DEI: on the one hand, the diversity and equity in the musical objects we study or create, as well as the musical artists, technological approaches, and content that we work with; and the diversity, equity, and inclusion of the people doing the research in MIR. From this standpoint, the DEI Chairs, in consultation with the organizing committee, coordinated a variety of initiatives with the aim of bringing together (and widening) the range of perspectives, traditions, and people across our MIR community. Notably, thanks to the Board and the generous support of our sponsors, we were able to support an unprecedented level of financial support covering travel, accommodation, registration, and childcare costs. Waivers and fee reductions for the above-mentioned categories were prioritized for underrepresented individuals including women, ethnic minorities, members of the LGBTQIA community, and attendees from low-income countries. In addition, priority was also given to unaffiliated attendees, "new-to-ISMIR" presenters, and students. All attendees were eligible to apply for childcare grants.

Inclusion Panel The aim of the DEI panel was to foster discussion, both between panelists, and between the audience and the panelists, relating to not only the diversity of people who work in MIR, but also diversity in topics, approaches, and data. Panelists were selected based on their excellent track records of commitment to diversity in MIR. After a brief overview of their research and how it has been influenced by, or relates to, DEI, our panelists contributed to a rich discussion of both the content and the people in MIR, and the ways in which we, as a community, can improve. Discussion topics included: bias in music access, consumption, and recommender systems; barriers and issues of DEI in MIR scholarship and how to overcome them; the relation of DEI and "ethical AI"; and how to increase the number of women and minorities in our field.

Moderator: Claire Arthur, Georgia Institute of Technology
Panelists: Anja Volk, Utrecht University; Jin Ha Lee, University of Washington; Christine Bauer, University of Salzburg; Lorenzo Pocarno, European Commission Joint Research Center (Milan)

Inclusion Meetup A DEI meetup session was planned with the aim of designing a social event that would encourage the intermingling of people from different backgrounds, levels of scholarship, and communities. Several interactive activities were organized to encourage movement, mingling, discussion, and entertainment. According to those who who were in attendance (over 100 people!), the event was a success.

Host: Riccardo Giampiccolo

Women in Music Information Retrieval (WiMIR)

Women in Music Information Retrieval (WiMIR) is a group of people dedicated to promoting the role of, and increasing opportunities for, women in the MIR field. WiMIR's initiatives started as informal gatherings around breakfast or lunch during ISMIR conferences (2011–2014), and moved to formal WiMIR events included in the conference program (2015–today) garnering a high turnout of both women and allies. These events provide occasions for people to network and to discuss several important issues ranging from mentorship and conference support, to improving the representation of women and, more broadly, diversity in the community. In 2018, WiMIR started hosting its own workshop as a satellite event, in which attendees of all genders participated. These workshops aim to offer participants an opportunity for networking, put the spotlight on technical work done by women in the field, and foster collaboration between women and allies by proposing group work led by project guides to try to solve small research problems or to undertake new

research projects that could lead to longer-term collaborations. In 2023, due to the decline (and rotation) in volunteers, the WiMIR workshop initiative was suspended. However, the aim is to resume these very popular and successful workshops again in 2024. The ISMIR 2023 DEI Chairs gratefully acknowledge the support of this year's WiMIR sponsors, whose contributions support women in the field as well as the broader DEI efforts of this year's conference.

Newcomer Initiative

A mentoring program was offered in 2023 for prospective authors who are new to ISMIR. They were given feedback on their ideas and drafts of their ISMIR submissions. We would like to thank the following people who volunteered to be mentors for this initiative:

- Emmanouil Benetos
- Geoffroy Peeters
- Brian McFee
- Juhan Nam
- Chris Donahue
- Cheng-i Wang
- Cory McKay
- Jin Ha Lee
- Gus Xia
- Mitsunori Ogihara
- Andre Holzapfel

Special Sessions

The Scientific Program Chairs organized two special sessions. Brief introductions and session information are provided below:

Panel session: Hybrid deep learning for MIR

In MIR, as in many other domains, there is a significant trend towards purely data-driven approaches aimed at directly solving the machine learning problem at hand, while only crudely considering the nature and structure of the data being processed. In the music domain, prior knowledge can relate to the production of sound (using an acoustic model), the way music is perceived (based on a perceptual model), or how music is composed (using a musicological model).

These models can usually be encoded with only a few parameters, leading to controllable and interpretable systems that can be exploited in modern neural-based machine learning frameworks, resulting in so-called hybrid deep learning models.

The aim of this panel was to illustrate the concept of hybrid deep learning with some specific examples in MIR, and to discuss its limits, merits and potential for future machine learning based music applications.

Moderator: Gaël Richard, Télécom Paris
Panelists: George Fazekas, Queen Mary University of London; Changhong Wang, Télécom Paris, Zhiyao Duan, University of Rochester; Gus Xia, NYU Shanghai/MBZUAI

Industry Panel

The industrial panel aimed to facilitate a high-level discussion, providing conference participants with insights into MIR efforts by ISMIR's sponsoring companies. This initiative aimed to foster collaborations among participants, be they from the industry or academic realm. The primary focus was on delving into the future of multi-modal AI in music research – a burgeoning paradigm that leverages diverse data types like audio, image, text, and speech to enhance outcomes. Each panelist briefly presented their perspective on the topic, leading to an open discussion. Notably, the discourse also explored the relevance of existing Large Language Models and their impact on the field of music.

Moderator: Xavier Serra, Director of the Music Technology Group of the Universitat Pompeu Fabra.
Panelists: Justin Salamon, Senior Research Scientist at Adobe; Elio Quinton, VP, Artificial Intelligence at Universal Music Group; Akira Maezawa, Senior Engineer at Yamaha; Fabien Gouyon, Senior Director of Research at Pandora – SiriusXM; Romain Hennequin, Head of Research at Deezer; Maria Stella Tavella, Senior AI Engineer and Manager at Musixmatch; Filip Korzeniowski, Lead Data Scientist at Moises.AI

Late Breaking/Demo Session

The Late Breaking/Demo (LBD) Session is where we showcase cool works that are still in the making — prototypes, early ideas and results that generate excitement in the MIR community. This year we received more submissions than expected. To handle the demand, we split the session into two parts and papers were presented in-person as well as using the virtual platform. Following a light review process by the LBD chairs, we accepted 40 papers for live, in-person presentation, while another 8 papers were accepted for virtual presentation, ensuring broader accessibility to the valuable insights shared within the LBD Session. This decision allowed us to accommodate the diverse preferences and circumstances of our contributors and attendees. Following previous years' practice, LBD contributions are not part of the official ISMIR proceedings and should be seen as non-refereed works — think of them as fresh works in progress and exciting fun demos that led to an exceptionally lively session contributing to this year's edition of ISMIR.

Unconference

ISMIR 2023 reintroduced the "Unconference" session, where participants team up into small groups to engage in discussions on Music Information Retrieval (MIR) topics of their specific interest. Two weeks prior to the session, participants were invited to propose their preferred topics. The session then started with a brief plenary in which session topics of greatest interest were selected. The session gathered around 80 participants, including both students and senior researchers from academia and industry. Four topics, namely "MIR + music education," "Open review for MIR," "Human-centered AI," and "Evaluation of generative AI," were chosen for the initial round of discussions. Subsequently, participants were divided into four groups, engaging in impromptu discussions for 30 minutes. Although it is customary in the Unconference to choose new topics every 30 minutes, the participants in each group were so engrossed in their discussions that they extended the conversation into a second and even a third round, all centered around the same topics. The session concluded with a plenary session, during which one representative from each group was invited to provide a summary of their discussions for the benefit of the other groups. Following this, participants were encouraged to continue their conversations beyond the session and explore opportunities for potential collaborations on the discussed topics.

Music Session

The ISMIR 2023 Music Session received nine submissions, six of which were accepted for presentation. Half of the contributions were performed live, whereas the remaining ones were pre-recorded and reproduced during the session. All the pre-recorded contributions included video content. Overall, the selected submissions showed a high degree of variety, ranging from AI-assisted performances to improvisations with augmented instruments and combinations with visual arts.

Here is the complete list of music pieces that were presented at ISMIR 2023:

Conversations with our Digital Selves: the development of an autonomous music improviser Matthew Yee-King, Mark d'Inverno

"confluyo yo, el ambiente me sigue" Hugo Flores Garcia

Sliogán: a performance composed for the HITar Andrea Martelloni, Andrew McPherson, Mathieu Barthet

The Words I Tried to Say Angela Weihan Ng

Nor Hope Wenbin Lyu

AI Pianist Performance: Collaboration with Soprano Sumi Jo Taegyun Kwon, Joonhyung Bae, Jiyun Park, Jaeran Choi, Hyeyoon Cho, Yonghyun Kim, Dasaem Jeong, Juhan Nam

Satellite Events

In addition to the main conference, four satellite events took place immediately before or after ISMIR, and were attended by many ISMIR delegates:

- Sound Demixing Workshop, November 4, 2023
- Workshop on Reading Music Systems (WoRMS), November 4, 2023
- Workshop on Human-Centric Music Information Research (HCMIR), November, 10, 2023
- 10th International Conference on Digital Libraries for Musicology (DLfM), November, 10, 2023

Acknowledgements

We are happy to present to you the proceedings of ISMIR 2023. The conference program was made possible thanks to the hard work of many people, including the ISMIR 2023 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:

Platinum sponsors

- Moises

Gold sponsors

- Yamaha
- MusixMatch
- Algoriddim
- Deezer
- Adobe
- Google Research
- Universal Music Group

Silver sponsors

- ACRCloud
- Steinberg
- Native Instruments
- SiriusXM

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

Patron

- Deezer

Contributors

- Moises
- Google Research
- Native Instruments
- SiriusXM

Supporters

- Steinberg

ISMIR 2023 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.

Paolo Bestagini
Simon Dixon
Beici Liang
Gaël Richard
Scientific Program Chairs

Augusto Sarti
Fabio Antonacci
Mark Sandler
General Chairs

Table of Contents

Keynote Talks **1**

Help! - Bridging the Gap Between Music Technology and Diverse Stakeholder Needs
Christine Bauer 3

Building & Launching MIR Systems at Industry Scale
Rachel Bittner 4

Seeing the Light Through Music, a Blind Man's Journey of Discovery Through Audio and How to Navigate Making Music That Speaks to the World in the Age of the Screen Driven Universe
Joey Stuckey 5

Tutorials **7**

Analysing Physiological Data Collected During Music Listening: An Introduction
Laura Bishop, Geoffray Bonnin and Jérémy Frey 9

Introduction to Differentiable Audio Synthesizer Programming
Ben Hayes, Jordie Shier, Chin-Yun Yu, David Südholt and Rodrigo Diaz 10

Transformer-Based Symbolic Music Generation: Fundamentals to Advanced Concepts, Stylistic Considerations, Conditioning Mechanisms and Large Language Models
Berker Banar, Pedro Sarmento and Sara Adkins 12

Computer-Assisted Music-Making Systems: Taxonomy, Review, and Coding
Christodoulos Benetatos, Zhiyao Duan and Philippe Pasquier 14

Learning With Music Signals: Technology Meets Education
Meinard Müller 15

Kymatio: Deep Learning Meets Wavelet Theory for Music Signal Processing
Cyrus Vahidi, Christopher Mitcheltree, Vincent Lostanlen 16

Papers – Session I **19**

Exploring the Correspondence of Melodic Contour With Gesture in Raga Alap Singing
Shreyas Nadkarni, Sujoy Roychowdhury, Preeti Rao, Martin Clayton 21

TriAD: Capturing Harmonics With 3D Convolutions
Miguel Perez, Holger Kirchhoff, Xavier Serra 29

Data Collection in Music Generation Training Sets: A Critical Analysis
Fabio Morreale, Megha Sharma, I-Chieh Wei 37

A Review of Validity and Its Relationship to Music Information Research
Bob L. T. Sturm, Arthur Flexer 47

Segmentation and Analysis of Taniavartanam in Carnatic Music Concerts
Gowriprasad R, Srikrishnan Sridharan, R Aravind, Hema A. Murthy 56

Transfer Learning and Bias Correction With Pre-Trained Audio Embeddings
Changhong Wang, Gaël Richard, Brian McFee 64

Collaborative Song Dataset (CoSoD): An Annotated Dataset of Multi-Artist Collaborations in Popular Music
Michèle Duguay, Kate Mancey, Johanna Devaney 71

Human-AI Music Creation: Understanding the Perceptions and Experiences of Music Creators for Ethical and Productive Collaboration
Michele Newman, Lidia Morris, Jin Ha Lee 80

Impact of Time and Note Duration Tokenizations on Deep Learning Symbolic Music Modeling
Nathan Fradet, Nicolas Gutowski, Fabien Chhel, Jean-Pierre Briot 89

Musical Micro-Timing for Live Coding
Max Johnson, Mark R. H. Gotham 98

A Few-Shot Neural Approach for Layout Analysis of Music Score Images
Francisco J. Castellanos, Antonio Javier Gallego, Ichiro Fujinaga 106

TapTamDrum: A Dataset for Dualized Drum Patterns
Behzad Haki, Błażej Kotowski, Cheuk Lun Isaac Lee, Sergi Jordà 114

Real-Time Percussive Technique Recognition and Embedding Learning for the Acoustic Guitar
Andrea Martelloni, Andrew P. McPherson, Mathieu Barthet 121

IteraTTA: An Interface for Exploring Both Text Prompts and Audio Priors in Generating Music With Text-to-Audio Models
Hiromu Yakura, Masataka Goto 129

Similarity Evaluation of Violin Directivity Patterns for Musical Instrument Retrieval
Mirco Pezzoli, Raffaele Malvermi, Fabio Antonacci, Augusto Sarti 138

Polyrhythmic Modelling of Non-Isochronous and Microtiming Patterns
George Sioros 146

Papers – Session II **155**

CLaMP: Contrastive Language-Music Pre-Training for Cross-Modal Symbolic Music Information Retrieval
Shangda Wu, Dingyao Yu, Xu Tan, Maosong Sun 157

Gender-Coded Sound: Analysing the Gendering of Music in Toy Commercials via Multi-Task Learning
Luca Marinelli, György Fazekas, Charalampos Saitis 166

A Dataset and Baselines for Measuring and Predicting the Music Piece Memorability
Li-Yang Tseng, Tzu-Ling Lin, Hong-Han Shuai, Jen-Wei Huang, Wen-Whei Chang 174

Efficient Notation Assembly in Optical Music Recognition
Carlos Peñarrubia, Carlos Garrido-Munoz, Jose J. Valero-Mas, Jorge Calvo-Zaragoza 182

White Box Search Over Audio Synthesizer Parameters
Yuting Yang, Zeyu Jin, Connelly Barnes, Adam Finkelstein 190

Decoding Drums, Instrumentals, Vocals, and Mixed Sources in Music Using Human Brain Activity With fMRI
Vincent K. M. Cheung, Lana Okuma, Kazuhisa Shibata, Kosetsu Tsukuda, Masataka Goto, Shinichi Furuya 197

Dual Attention-Based Multi-Scale Feature Fusion Approach for Dynamic Music Emotion Recognition
Liyue Zhang, Xinyu Yang, Yichi Zhang, Jing Luo 207

Automatic Piano Transcription With Hierarchical Frequency-Time Transformer
Keisuke Toyama, Taketo Akama, Yukara Ikemiya, Yuhta Takida, Wei-Hsiang Liao, Yuki Mitsufuji 215

High-Resolution Violin Transcription Using Weak Labels
Nazif Can Tamer, Yigitcan Özer, Meinard Müller, Xavier Serra 223

Polyffusion: A Diffusion Model for Polyphonic Score Generation With Internal and External Controls
Lejun Min, Junyan Jiang, Gus Xia, Jingwei Zhao 231

The Coordinated Corpus of Popular Musics (CoCoPops): A Meta-Corpus of Melodic and Harmonic Transcriptions
Claire Arthur, Nathaniel Condit-Schultz 239

Towards Computational Music Analysis for Music Therapy
Anja Volk, Tinka Veldhuis, Katrien Foubert, Jos De Backer 247

Timbre Transfer Using Image-to-Image Denoising Diffusion Implicit Models
Luca Comanducci, Fabio Antonacci, Augusto Sarti 257

Correlation of EEG Responses Reflects Structural Similarity of Choruses in Popular Music
Neha Rajagopalan, Blair Kaneshiro 264

Chromatic Chords in Theory and Practice
Mark R. H. Gotham 272

Papers – Session III **279**

BPS-Motif: A Dataset for Repeated Pattern Discovery of Polyphonic Symbolic Music
Yo-Wei Hsiao, Tzu-Yun Hung, Tsung-Ping Chen, Li Su 281

Weakly Supervised Multi-Pitch Estimation Using Cross-Version Alignment
Michael Krause, Sebastian Strahl, Meinard Müller 289

The Batik-Plays-Mozart Corpus: Linking Performance to Score to Musicological Annotations
Patricia Hu, Gerhard Widmer 297

Mono-to-Stereo Through Parametric Stereo Generation
Joan Serrà, Davide Scaini, Santiago Pascual, Daniel Arteaga, Jordi Pons, Jeroen Breebaart, Giulio Cengarle 304

From West to East: Who Can Understand the Music of the Others Better?
Charilaos Papaioannou, Emmanouil Benetos, Alexandros Potamianos 311

On the Performance of Optical Music Recognition in the Absence of Specific Training Data
Juan C. Martinez-Sevilla, Adrián Roselló, David Rizo, Jorge Calvo-Zaragoza 319

Composer's Assistant: An Interactive Transformer for Multi-Track MIDI Infilling
Martin E. Malandro 327

The FAV Corpus: An Audio Dataset of Favorite Pieces and Excerpts, With Formal Analyses and Music Theory Descriptors
Ethan Lustig, David Temperley 335

LyricWhiz: Robust Multilingual Zero-Shot Lyrics Transcription by Whispering to ChatGPT
Le Zhuo, Ruibin Yuan, Jiahao Pan, Yinghao Ma, Yizhi Li, Ge Zhang, Si Liu, Roger B. Dannenberg, Jie Fu, Chenghua Lin, Emmanouil Benetos, Wenhu Chen, Wei Xue, Yike Guo 343

Sounds Out of Pläce? Score-Independent Detection of Conspicuous Mistakes in Piano Performances
Alia Morsi, Kana Tatsumi, Akira Maezawa, Takuya Fujishima, Xavier Serra 352

VampNet: Music Generation via Masked Acoustic Token Modeling
Hugo Flores García, Prem Seetharaman, Rithesh Kumar, Bryan Pardo 359

Expert and Novice Evaluations of Piano Performances: Criteria for Computer-Aided Feedback
Yucong Jiang 367

Contrastive Learning for Cross-Modal Artist Retrieval
Andres Ferraro, Jaehun Kim, Sergio Oramas, Andreas Ehmann, Fabien Gouyon 375

Repetition-Structure Inference With Formal Prototypes
Christoph Finkensiep, Matthieu Haeberle, Friedrich Eisenbrand, Markus Neuwirth, Martin Rohrmeier . . 383

Algorithmic Harmonization of Tonal Melodies Using Weighted Pitch Context Vectors
Peter van Kranenburg, Eoin J. Kearns 391

Text-to-Lyrics Generation With Image-Based Semantics and Reduced Risk of Plagiarism
Kento Watanabe, Masataka Goto 398

Papers – Session IV **407**

LP-MusicCaps: LLM-Based Pseudo Music Captioning
SeungHeon Doh, Keunwoo Choi, Jongpil Lee, Juhan Nam 409

A Repetition-Based Triplet Mining Approach for Music Segmentation
Morgan Buisson, Brian McFee, Slim Essid, Helene C. Crayencour 417

Predicting Music Hierarchies With a Graph-Based Neural Decoder
Francesco Foscarin, Daniel Harasim, Gerhard Widmer 425

Stabilizing Training With Soft Dynamic Time Warping: A Case Study for Pitch Class Estimation With Weakly Aligned Targets
Johannes Zeitler, Simon Deniffel, Michael Krause, Meinard Müller 433

Finding Tori: Self-Supervised Learning for Analyzing Korean Folk Song
Danbinaerin Han, Rafael Caro Repetto, Dasaem Jeong 440

Singer Identity Representation Learning Using Self-Supervised Techniques
Bernardo Torres, Stefan Lattner, Gaël Richard 448

On the Effectiveness of Speech Self-Supervised Learning for Music
Yinghao Ma, Ruibin Yuan, Yizhi Li, Ge Zhang, Chenghua Lin, Xingran Chen, Anton Ragni, Hanzhi Yin, Emmanouil Benetos, Norbert Gyenge, Ruibo Liu, Gus Xia, Roger B. Dannenberg, Yike Guo, Jie Fu 457

Transformer-Based Beat Tracking With Low-Resolution Encoder and High-Resolution Decoder
Tian Cheng, Masataka Goto 466

Adding Descriptors to Melodies Improves Pattern Matching: A Study on Slovenian Folk Songs
Vanessa Nina Borsan, Mathieu Giraud, Richard Groult, Thierry Lecroq 474

How Control and Transparency for Users Could Improve Artist Fairness in Music Recommender Systems
Karlijn Dinnissen, Christine Bauer 482

Towards a New Interface for Music Listening: A User Experience Study on YouTube
Ahyeon Choi, Eunsik Shin, Haesun Joung, Joongseek Lee, Kyogu Lee 492

FiloBass: A Dataset and Corpus Based Study of Jazz Basslines
Xavier Riley, Simon Dixon 500

Comparing Texture in Piano Scores
Louis Couturier, Louis Bigo, Florence Levé 508

Introducing DiMCAT for Processing and Analyzing Notated Music on a Very Large Scale
Johannes Hentschel, Andrew McLeod, Yannis Rammos, Martin Rohrmeier 516

Sequence-to-Sequence Network Training Methods for Automatic Guitar Transcription With Tokenized Outputs
Sehun Kim, Kazuya Takeda, Tomoki Toda 524

Papers – Session V **533**

PESTO: Pitch Estimation With Self-Supervised Transposition-Equivariant Objective
Alain Riou, Stefan Lattner, Gaëtan Hadjeres, Geoffroy Peeters 535

The Games We Play: Exploring the Impact of ISMIR on Musicology
Vanessa Nina Borsan, Mathieu Giraud, Richard Groult 545

Carnatic Singing Voice Separation Using Cold Diffusion on Training Data With Bleeding
Genís Plaja-Roglans, Marius Miron, Adithi Shankar, Xavier Serra 553

Unveiling the Impact of Musical Factors in Judging a Song on First Listen: Insights From a User Survey
Kosetsu Tsukuda, Tomoyasu Nakano, Masahiro Hamasaki, Masataka Goto 561

Towards Building a Phylogeny of Gregorian Chant Melodies
Jan Hajič jr., Gustavo A. Ballen, Klára Hedvika Mühlová, Hana Vlhová-Wörner 571

Audio Embeddings as Teachers for Music Classification
Yiwei Ding, Alexander Lerch 579

ScorePerformer: Expressive Piano Performance Rendering With Fine-Grained Control
Ilya Borovik, Vladimir Viro 588

Roman Numeral Analysis With Graph Neural Networks: Onset-Wise Predictions From Note-Wise Features
Emmanouil Karystinaios, Gerhard Widmer 597

Semi-Automated Music Catalog Curation Using Audio and Metadata
Brian Regan, Desislava Hristova, Mariano Beguerisse-Díaz 605

Crowd's Performance on Temporal Activity Detection of Musical Instruments in Polyphonic Music
Ioannis Petros Samiotis, Christoph Lofi, Alessandro Bozzon 612

MoisesDB: A Dataset for Source Separation Beyond 4-Stems
Igor Pereira, Felipe Araújo, Filip Korzeniowski, Richard Vogl 619

Music as Flow: A Formal Representation of Hierarchical Processes in Music
Zeng Ren, Wulfram Gerstner, Martin Rohrmeier 627

Online Symbolic Music Alignment With Offline Reinforcement Learning
Silvan David Peter 634

Inversynth II: Sound Matching via Self-Supervised Synthesizer-Proxy and Inference-Time Finetuning
Oren Barkan, Shlomi Shvartzman, Noy Uzrad, Moshe Laufer, Almog Elharar, Noam Koenigstein 642

A Semi-Supervised Deep Learning Approach to Dataset Collection for Query-by-Humming Task
Amantur Amatov, Dmitry Lamanov, Maksim Titov, Ivan Vovk, Ilya Makarov, Mikhail Kudinov 649

Towards Improving Harmonic Sensitivity and Prediction Stability for Singing Melody Extraction
Keren Shao, Ke Chen, Taylor Berg-Kirkpatrick, Shlomo Dubnov 657

Papers – Session VI **665**

Singing Voice Synthesis Using Differentiable LPC and Glottal-Flow-Inspired Wavetables
Chin-Yun Yu, György Fazekas 667

Harmonic Analysis With Neural Semi-CRF
Qiaoyu Yang, Frank Cwitkowitz, Zhiyao Duan 676

A Dataset and Baseline for Automated Assessment of Timbre Quality in Trumpet Sound
Alberto Acquilino, Ninad Puranik, Ichiro Fujinaga, Gary Scavone 684

Visual Overviews for Sheet Music Structure
Frank Heyen, Quynh Quang Ngo, Michael Sedlmair 692

Passage Summarization With Recurrent Models for Audio – Sheet Music Retrieval
Luís Carvalho, Gerhard Widmer 700

Predicting Performance Difficulty From Piano Sheet Music Images
Pedro Ramoneda, Jose J. Valero-Mas, Dasaem Jeong, Xavier Serra 708

Self-Refining of Pseudo Labels for Music Source Separation With Noisy Labeled Data
Junghyun Koo, Yunkee Chae, Chang-Bin Jeon, Kyogu Lee 716

Quantifying the Ease of Playing Song Chords on the Guitar
Marcel A. Vélez Vásquez, Mariëlle Baelemans, Jonathan Driedger, Willem Zuidema, John Ashley Burgoyne 725

FlexDTW: Dynamic Time Warping With Flexible Boundary Conditions
Irmak Bükey, Jason Zhang, TJ Tsai 733

Modeling Bends in Popular Music Guitar Tablatures
Alexandre D'Hooge, Louis Bigo, Ken Déguernel 741

Self-Similarity-Based and Novelty-Based Loss for Music Structure Analysis
Geoffroy Peeters 749

Modeling Harmonic Similarity for Jazz Using Co-occurrence Vectors and the Membrane Area
Carey Bunks, Tillman Weyde, Simon Dixon, Bruno Di Giorgi 757

SingStyle111: A Multilingual Singing Dataset With Style Transfer
Shuqi Dai, Yuxuan Wu, Siqi Chen, Roy Huang, Roger B. Dannenberg 765

A Computational Evaluation Framework for Singable Lyric Translation
Haven Kim, Kento Watanabe, Masataka Goto, Juhan Nam 774

Chorus-Playlist: Exploring the Impact of Listening to Only Choruses in a Playlist
Kosetsu Tsukuda, Masahiro Hamasaki, Masataka Goto 782

Papers – Session VII **793**

Supporting Musicological Investigations With Information Retrieval Tools: An Iterative Approach to Data Collection
David Lewis, Elisabete Shibata, Andrew Hankinson, Johannes Kepper, Kevin R. Page, Lisa Rosendahl, Mark Saccomano, Christine Siegert 795

Optimizing Feature Extraction for Symbolic Music
Federico Simonetta, Ana Llorens, Martín Serrano, Eduardo García-Portugués, Álvaro Torrente 802

Exploring Sampling Techniques for Generating Melodies With a Transformer Language Model
Mathias Rose Bjare, Stefan Lattner, Gerhard Widmer 810

Measuring the Eurovision Song Contest: A Living Dataset for Real-World MIR
John Ashley Burgoyne, Janne Spijkervet, David John Baker 817

Efficient Supervised Training of Audio Transformers for Music Representation Learning
Pablo Alonso-Jiménez, Xavier Serra, Dmitry Bogdanov 824

A Cross-Version Approach to Audio Representation Learning for Orchestral Music
Michael Krause, Christof Weiß, Meinard Müller 832

Music Source Separation With MLP Mixing of Time, Frequency, and Channel
Tomoyasu Nakano, Masataka Goto 840

Symbolic Music Representations for Classification Tasks: A Systematic Evaluation
Huan Zhang, Emmanouil Karystinaios, Simon Dixon, Gerhard Widmer, Carlos Eduardo Cancino-Chacón 848

The Music Meta Ontology: A Flexible Semantic Model for the Interoperability of Music Metadata
Jacopo de Berardinis, Valentina Anita Carriero, Albert Meroño-Peñuela, Andrea Poltronieri, Valentina Presutti 859

Polar Manhattan Displacement: Measuring Tonal Distances Between Chords Based on Intervallic Content
Jeff Miller, Johan Pauwels, Mark Sandler 868

Author Index **875**

Keynote Talks

Keynote Talk – 1

Help! – Bridging the Gap Between Music Technology and Diverse Stakeholder Needs

Christine Bauer

Professor of Interactive Intelligent Systems
Paris Lodron University Salzburg

Abstract

Music information retrieval (MIR) has become an indispensable asset in the music industry. It powers music recommendations for listeners and supports artists in mastering their crafts. While MIR has made remarkable progress, we need to improve in serving the multifaceted needs of stakeholders who rely on these technologies. Taking examples from music recommender systems, I will demonstrate the potential risks of neglecting artists' needs and provide strategies for mitigation.

Biography

Christine Bauer is EXDIGIT Professor of Interactive Intelligent Systems at the Department of Artificial Intelligence and Human Interfaces (AIHI) at the Paris Lodron University Salzburg, Austria.

Her research centers on interactive intelligent systems, where she integrates research on intelligent technologies, the interaction of humans with an intelligent system, and their interplay. She takes a human-centered perspective, where technology follows humans' and society's needs. In recent years, she worked on context-aware recommender systems in the music and media domains. The core interests in her research activities are fairness and multi-method evaluations.

She has authored more than 100 papers and holds several best paper awards and many awards for her reviewing activities. She received the prestigious Elise Richter career research grant (2017–2020), funded by the Austrian Science Fund (FWF). She is on the Editorial Board of ACM Transactions on Recommender Systems (TORS) and co-organizes the Workshop series "Perspectives on the Evaluation of Recommender Systems (PERSPECTIVES)".

She advocates for equal opportunities and engages in initiatives such as Women in Music Information Retrieval (WiMIR) and the Allyship program at CHI.

Further information can be found at https://christinebauer.eu.

Keynote Talk – 2

Building & Launching MIR Systems at Industry Scale

Rachel Bittner

Research Manager
Spotify

Abstract

There is a considerable gap in the research and engineering methods we use to build MIR systems for academic research and the way we build them for industry-scale systems. This keynote covers some of the many differences and challenges faced when building MIR systems for industry applications. We first discuss the way we define problems in the first place, and why the academic definition of problems is often ill-suited for a particular application. There are also substantial differences in engineering workflows – in particular when multiple researchers and engineers build a single system. We explore differences in academic datasets which are usually "small and clean" to real-world datasets which are "large and noisy". Academic metrics are useful for us scientists, but they often either don't match a product use case or mean nothing to product teams. Finally, we dig into deployment considerations including how to run inference flexibly, considering cost and speed, and where the system needs to run. We will explore numerous real-world examples throughout and provide insight into how to build MIR systems within industry.

Biography

Rachel is a Research Manager at Spotify in Paris. Before Spotify, she worked at NASA Ames Research Center in the Human Factors division. She received her Ph.D. degree in music technology and digital signal processing from New York University. Before that, she did a Master's degree in Mathematics at New York University, and a joint Bachelor's degree in Music Performance and Math at UC Irvine. Her research interests include automatic music transcription, musical source separation, metrics, and dataset creation.

Keynote Talk – 3

Seeing the Light Through Music, a Blind Man's Journey of Discovery Through Audio and How to Navigate Making Music That Speaks to the World in the Age of the Screen Driven Universe

Joey Stuckey

Professor of Music Technology
Mercer University

Abstract

This presentation will encompass:

- Diversity, Equity, Inclusion and Accessibility issues and best practices for a truly vibrant and equitable community in the audio industry and music business.
- Getting back to fundamentals, critical listening in the age of the "Screen Driven Universe".
- Important elements of music making and the recording sciences
- How to live a successful life of intention despite obstacles

Biography

Joey Stuckey is the Official Music Ambassador of his hometown of Macon, Georgia. Joey spends every moment living life to the fullest and sharing his story and inspirational spirit through his musical performances and speaking engagements. As a toddler, Joey was diagnosed with a brain tumor and underwent surgery with little hope of survival. Though the tumor left Joey blind and with other health challenges, today, he continues to live a successful life of intention in his chosen field of music. Joey is professor of music technology at Mercer University, the music technology consultant for Middle Georgia State University, and an official music mentor for the Recording, Radio and Film Connection in Los Angeles as well as an active voting member of the Grammys. He is the owner and senior engineer at Shadow Sound Studio which is a destination recording facility with state-of-the-art analog and digital technology. He has spoken and performed all over the world including at the University College of London, the Georgia Music Hall of Fame, and the Audio Engineering Society in New York City, just to name a few. In his roles as producer, engineer, recording artist and journalist, he has worked with many musical legends including Trisha Yearwood, Clarence Carter, James Brown, Alan Parsons, Gene Simmons (KISS), Al Chez (Tower of Power), Jimmy Hall (Wet Willie), Danny Seraphin (Chicago), Kevin Kenney (Drivin' and Cryin'), and many, many more.

For more information visit www.joeystuckey.com

Facebook: https://www.facebook.com/joeystuckey

Twitter: @jstuckeymusic

Instagram: @jstuckeymusic

Tutorials

Tutorial 1

Analysing Physiological Data Collected During Music Listening: An Introduction

Laura Bishop, Geoffray Bonnin and Jérémy Frey

Abstract

Music has diverse effects on listeners, including inducing emotions, triggering movement or dancing, and prompting changes in visual attention. These effects are often associated with psychophysiological responses like changes in heart activity, respiratory rate, and pupil size, which can themselves be influenced by the cognitive effort exerted during music listening, e.g., when engaging with unfamiliar tracks on a web radio for music discovery.

This tutorial aims to introduce psychophysiological data analysis for a broad MIR audience, with a particular focus on the analysis of heart rate, electrodermal activity and pupillometry data. It will be structured in three parts. The first part will provide a presentation of psychophysiological data that we collected in the context of a preliminary study related to music discovery. The second part will be a hands-on tutorial during which we will guide the participants to remake two of our data analyses. In the third part, we will assist participants in undertaking their own data analysis of our data. These analyses will be demonstrated using R and Python.

Our aim with this tutorial is twofold: to promote underrepresented topics in the MIR community, especially the recognition of induced emotions from physiological data and discovery-oriented music recommendation; and to encourage researchers from those domains to interact with the MIR community. The audience we target is therefore relatively large. Participants should, however, possess sufficient knowledge of R and/or Python and standard statistical analysis methods to participate in the hands-on parts of the tutorial.

Biographies of the Presenters

Laura Bishop is a researcher at the RITMO Centre for Interdisciplinary Studies in Rhythm, Time and Motion and the Department of Musicology at the University of Oslo. She specialises in pupillometry, eye-tracking, and motion capture using approaches mainly grounded in psychology. She completed her PhD in music psychology at the MARCS Institute, Western Sydney University, Australia, in 2013. She currently co-leads the Austrian Science Fund project "Achieving togetherness in music ensembles" in collaboration with the University for Music and Performing Arts Vienna (mdw), which investigates physiological and body motion coordination in ensemble playing.

Geoffray Bonnin is an Associate Professor at the Lorraine Research Laboratory in Computer Science and its Applications (Loria), Université de Lorraine. He obtained his Ph.D. in 2010 and joined the Loria lab in 2014 as an Associate Professor. His research topics are related to artificial intelligence for music and for education. He is currently in charge of the Music-Mouv' project, which is a collaboration with researchers in the domain of psychology that started in October 2021. The project aims at helping individuals with Parkinson's disease to walk by triggering relevant emotions through physiology-based music recommendations.

Jérémy Frey is the CTO and co-founder of Ullo. After a master degree in cognitive sciences, he obtained his PhD degree in computer science in 2015 from the University of Bordeaux, France. During his work within the Inria research team Potioc, he had been studying how passive brain-computer interfaces could contribute to the evaluation of user experience, using for example EEG to infer a continuous index of cognitive load. His current research interests revolve around increasing introspection and social presence, by displaying inner states through tangible interfaces or wearables, with applications ranging from well-being to education.

Tutorial 2

Introduction to Differentiable Audio Synthesizer Programming

Ben Hayes, Jordie Shier, Chin-Yun Yu, David Südholt and Rodrigo Diaz

Abstract

Differentiable digital signal processing is a technique in which signal processing algorithms are implemented as differentiable programs used in combination with deep neural networks. The advantages of this methodology include a reduction in model complexity, lower data requirements, and an inherently interpretible intermediate representation. In recent years, differentiable audio synthesizers have been applied to a variety of tasks, including voice and instrument modelling, synthesizer control, pitch estimation, source separation, and parameter estimation. Yet despite the growing popularity of such methods, the implementation of differentiable audio synthesizers remains poorly documented, and the simple formulation of many synthesizers belies their complex optimization behaviour. To address this gap, this tutorial offers an introduction to the fundamentals of differentiable synthesizer programming.

The tutorial will centre around practical demonstrations, which participants can follow using an accompanying suite of Jupyter notebooks. All tutorial content will be documented in an accompanying web book, and all tutorial materials and dependencies will be fully open source and accessible for free online. Prior experience with writing Python 3 code is assumed, and a basic knowledge of PyTorch is beneficial though not strictly required. The tutorial is targeted at music and audio researchers and engineers with a grounding in the basics of digital signal processing and machine learning. Our aim is to equip participants to apply these techniques in their own research, whilst enabling those with prior knowledge to sharpen their skills.

Biographies of the Presenters

Ben Hayes is a third year PhD student at the Centre for Digital Music's CDT in Artificial Intelligence and Music, based at Queen Mary University of London, under the supervision of Dr György Fazekas and Dr Charalampos Saitis. His research focuses on expanding the capabilities of differentiable digital signal processing by enabling control over non-convex operations. His work has been accepted to leading conferences in the field, including ISMIR, ICASSP, ICA, and the AES Convention, and published in the Journal of the Audio Engineering Society. He also holds an MSc with Distinction in Sound and Music Computing from QMUL and a first class BMus(Hons) in Electronic Music from the Guildhall School of Music and Drama, where he is now a member of teaching faculty. He is a founding member of the Special Interest Group on Neural Audio Synthesis at C4DM, and is the organizer of the international Neural Audio Synthesis Hackathon. Previously he was a Research intern at ByteDance, music lead at the award-winning generative music startup Jukedeck, and an internationally touring musician signed to R&S Records.

Jordie Shier is a first year PhD student in the Artificial Intelligence and Music (AIM) programme based at Queen Mary University of London (QMUL), studying under the supervision of Prof. Andrew McPherson and Dr. Charalampos Saitis. His research is focused on the development of novel methods for synthesizing audio and the creation of new interaction paradigms for music synthesizers. His current PhD project is on real-time timbral mapping for synthesized percussive performance and is being conducted in collaboration with Ableton. He was a co-organizer of the 2021 Holistic Evaluation of Audio Representations (HEAR) NeurIPS challenge and his work has been published in PMLR, DAFx, and the JAES. Previously, he completed an MSc in Computer Science and Music under the supervision of Prof. George Tzanetakis and Assoc. Prof. Kirck McNally.

Chin-Yun Yu is a first year PhD student in the Artificial Intelligence and Music (AIM) programme based at Queen Mary University of London (QMUL), under the supervision of Dr György Fazekas. His current research theme is on leveraging signal processing and deep generative models for controllable, expressive vocal synthesis. In addition, he is dedicated to open science and reproducible research by developing open-source packages and contributing to public research projects. He received a BSc in Computer Science from National Chiao Tung University in 2018 and was a research assistant at the Institute of Information Science, Academia Sinica, supervised by Prof. Li Su. His recent work has been published at ICASSP.

David Südholt is a first year PhD student in the Artificial Intelligence and Music (AIM) programme based at Queen Mary University of London (QMUL). Supervised by Prof. Joshua Reiss, he is researching parameter estimation for physical modelling synthesis, focussing on the synthesis and expressive transformation of the human voice. He received an MSc degree in Sound and Music Computing from Aalborg University Copenhagen in 2022, where he was supervised by Prof. Stefania Serafin and Assoc. Prof. Cumhur Erkut. His work has been published at the SMC conference and in the IEEE/ACM Transactions on Audio, Speech and Language Processing.

Rodrigo Diaz is a PhD candidate in Artificial Intelligence and Music at Queen Mary University in London, under the supervision of Prof. Mark Sandler and Dr. Charalampos Saitis. Rodrigo's work has been published in leading computer vision and audio conferences, including CVPR, ICASSP, IC3D, and the AES Conference on Headphone Technology. Before starting his PhD studies, he worked as a researcher at the Immersive Communications group at the Fraunhofer HHI Institute in Berlin, where he investigated volumetric reconstruction from images using neural networks. His current research focuses on real-time audio synthesis using neural networks for 3D objects and drums. Rodrigo's interdisciplinary background includes a Master's degree in Media Arts and Design from Bauhaus University in Weimar and a Bachelor of Music from Texas Christian University.

Tutorial 3

Transformer-Based Symbolic Music Generation: Fundamentals to Advanced Concepts, Stylistic Considerations, Conditioning Mechanisms and Large Language Models

Berker Banar, Pedro Sarmento and Sara Adkins

Abstract

With the rise of the attention mechanism and the success of auto-regressive generative modelling and large language models, the Transformer architecture has arguably been the most promising technology for symbolic music generation. While audio-based methods have shown promise, symbolic music generation offers distinct advantages in terms of control, long-term coherence and computational efficiency. This tutorial explores the potential of the Transformer architecture in symbolic music generation and aims to provide (1) a thorough understanding of the vanilla Transformer architecture (emphasising the reasoning behind its design choices) and the utilisation of large language models for symbolic music generation. Additionally, it offers (2) a comprehensive overview of the field, including a taxonomy and a curated list of valuable datasets. The tutorial delves into (3) an in-depth analysis of Transformer variants and large language models specifically tailored for symbolic music generation. Also, it examines (4) examples and advanced considerations such as style, musical conditioning, and real-time performance. Furthermore, the tutorial offers (5) two hands-on exercises using Google Colab Notebooks, enabling participants to apply the concepts covered. Overall, this tutorial equips participants with the theoretical knowledge and practical skills necessary to explore the power of the Transformer architecture in symbolic music generation.

Biographies of the Presenters

Berker Banar is a PhD Researcher (Comp. Sci.) at the Centre for Doctoral Training in AI and Music (AIM CDT) and the Centre for Digital Music (C4DM) at Queen Mary University of London (QMUL), and also an Enrichment Student at the Alan Turing Institute. His PhD focuses on 'Composing Contemporary Classical Music using Generative Deep Learning' under supervision of Simon Colton to enhance human creativity and enable new aesthetics. Berker's research interests include transformer-based generative modelling, optimisation, self-supervised representation learning for audio and music, explainable AI, quality-diversity analysis of generative model and out-of-distribution generation. He has worked at Sony and Bose as a research intern, and at Northwestern University Metamaterials and Nanophotonic Devices Lab as a nanophotonics researcher. Berker holds a BS in Electrical and Electronics Engineering from Bilkent University, Ankara, Turkey and a BM in Electronic Production and Design from Berklee College of Music, Boston, MA. His awards include Enrichment Community Award (The Alan Turing Institute), Exceptional Great Work Award (Bose), Outstanding Students of 2022 (EvoMUSART), Roland Award Endowed Scholarship (Berklee) and Outstanding Success Scholarship (Turkish Educational Foundation, upon ranking 17th in 1.5 million people in national university entrance exam). As a musician (drums and electronics), Berker has performed at venues such as the Museum of Fine Arts Boston, Harvard University Holden Chapel & Carpenter Center for Visual Arts (an original piece premiered as part of Berklee Interdisciplinary Arts Institute), Berklee Performance Center, Wally's Jazz Club Boston, Nardis Jazz Club Istanbul and Istanbul Jazz Festival.

Pedro Sarmento is a PhD researcher at the Centre for Digital Music (C4DM), Queen Mary University of London (QMUL), working under the supervision of Mathieu Barthet within the UKRI Centre for Doctoral Training in Artificial Intelligence and Music (AIM). His research focuses on guitar-focused symbolic music generation with deep learning. This concerns the exploration of techniques for the creation of novel music that is represented in a digital tablature format, in which additional information about how to play specific music passages is provided. He holds an Integrated MSc degree in Electrical Engineering from Faculdade de Engenharia da Universidade do Porto (FEUP), a degree in Classical Guitar from the Conversatory of Music of Porto, and a second MSc degree in Multimedia and Interactive Sound from FEUP. He has an ongoing collaboration with Orquestra de Jazz de Matosinhos (OJM) where he leads sessions that foster an approach to STEM via musical concepts for young students. He volunteers for an online music magazine, writing album reviews and conducting interviews with artists from the Metal scene.

Sara Adkins is a music technologist, machine learning engineer, and performer who is enthusiastic about promoting the use of machine learning and AI in the creative arts. Currently, she works as a Generative Music and Audio Developer at Infinite Album, developing a real-time, interactive, and copyright-safe music engine for Twitch streamers. Sara holds a Master of Science in Sound and Music Computing from Queen Mary University of London where she was funded through a US-UK Fulbright grant. Her master's thesis focused on developing a Transformer model capable of generating loopable musical phrases for live coding and algorave performances, and received an Outstanding Student Mention at EvoMUSART 2023. Before moving to London, Sara spent three years in Boston where she worked as a machine learning engineer at Bose and played as a freelance classical guitarist. At Bose, she worked on deep learning models for speech enhancement that were optimized to run live on a hearing aid micro-controller. She also led a research project that developed generative audio algorithms that adapt to biofeedback signals to induce sleep using soothing music. Sara graduated from Carnegie Mellon University with an interdisciplinary bachelor's degree in music technology and computer science. Her senior capstone project, "Creating with the Machine," combined algorithmic and traditional methods of composition into live performances to explore how interactive generative algorithms can influence creativity in musical improvisation. "Creating with the Machine" was premiered by the Carnegie Mellon Exploded Ensemble in the spring of 2018 and was awarded the Henry Armero Memorial Award for Inclusive Creativity.

Tutorial 4

Computer-Assisted Music-Making Systems: Taxonomy, Review, and Coding

Christodoulos Benetatos, Zhiyao Duan and Philippe Pasquier

Abstract

Computer-Assisted Music-Making (CAMM) systems, are software-based tools designed to assist and augment the musical creativity of composers, performers, and music enthusiasts. CAMM systems encompass a wide range of systems that can be broadly categorized into two main types according to their design purposes: to assist music performance and to assist music composition. This tutorial offers a comprehensive review of the design principles, practical applications, taxonomy, and the state-of-the-art research of CAMM systems, with an emphasis on systems assisting music performance, which are also called "interactive music systems" or "musical agents" in the literature. Research on CAMMs is interdisciplinary in its nature, combining fields such as Music Information Retrieval (MIR), Artificial Intelligence (AI) and Human-Computer Interaction (HCI). Participants will gain an understanding of how these fields converge to create innovative and interactive musical experiences. This tutorial will also feature a coding session for participants to build a real-time musical agent, under the framework of Euterpe, a prototyping framework for creating music interactions on the Web. The tutorial will examine existing systems built using Euterpe, provide insights into the development process, and guide participants through the creation of their own musical agents. Participants in the coding part should bring a laptop with Chrome and Node.js (https://nodejs.org/en/download) installed, as well as have some coding experience. Familiarity with JavaScript will be helpful, but not necessary.

Biographies of the Presenters

Christodoulos Benetatos is a 5th year Ph.D student in the Department of Electrical and Computer Engineering at the University of Rochester. He received his B.S and M.Eng in Electrical Engineering from National Technical University of Athens in 2018. His research interests are focused primarily on automatic music generation as well as the design and development of computer-assisted music-making systems. During his research internships at Kwai and TikTok, he worked on audio digital signal processing and music generation algorithms. As a classical guitarist, he has won several prizes in international guitar competitions and is a regular performer both as a soloist and as part of ensembles.

Philippe Pasquier is a professor at Simon Fraser University's School of Interactive Arts and Technology, where he directs the Metacreation Lab for Creative AI. He leads a research-creation program around generative systems for creative tasks. As such, he is a scientist specialized in artificial intelligence, a software designer, a multidisciplinary media artist, an educator, and a community builder. Pursuing a multidisciplinary research-creation program, his contributions bridge fundamental research on generative systems, machine learning, affective computing and computer-assisted creativity, applied research in the creative software industry, and artistic practice in interactive and generative art.

Zhiyao Duan is an associate professor in Electrical and Computer Engineering, Computer Science and Data Science at the University of Rochester. He received his B.S. in Automation and M.S. in Control Science and Engineering from Tsinghua University, China, in 2004 and 2008, respectively, and received his Ph.D. in Computer Science from Northwestern University in 2013. His research interest is in computer audition and its connections with computer vision, natural language processing, and augmented and virtual reality. He received a best paper award at the Sound and Music Computing (SMC) conference in 2017, a best paper nomination at the International Society for Music Information Retrieval (ISMIR) conference in 2017, and a CAREER award from the National Science Foundation (NSF). He served as a Scientific Program Co-Chair of ISMIR 2021, and is serving as an associate editor for IEEE Open Journal of Signal Processing, a guest editor for Transactions of the International Society for Music Information Retrieval, and a guest editor for Frontiers in Signal Processing. He is the President-Elect of ISMIR.

Tutorial 5

Learning With Music Signals: Technology Meets Education

Meinard Müller

Abstract

Music information retrieval (MIR) is an exciting and challenging research area that aims to develop techniques and tools for organizing, analyzing, retrieving, and presenting music-related data. Being at the intersection of engineering and humanities, MIR relates to different research disciplines, including signal processing, machine learning, information retrieval, musicology, and the digital humanities. In this tutorial, using music as a tangible and concrete application domain, we approach the concept of learning from different angles, addressing technological and educational aspects. In this way, the tutorial serves several purposes: we give a gentle introduction to MIR, highlight avenues for developing explainable machine-learning models, discuss how recent technology can be applied and communicated in interdisciplinary research and education, and introduce a new software package for teaching and learning music processing.

Our primary goal is to give an exciting tutorial that builds a bridge from basic to advanced techniques in MIR while highlighting technological and educational aspects. This tutorial should appeal to a broad audience, including students, educators, non-experts, and researchers new to the field, by covering concrete MIR tasks while providing many illustrative audio examples.

Links:

- Textbook: Fundamentals of Music Processing www.music-processing.de
- FMP Notebooks www.audiolabs-erlangen.de/FMP
- Python package: libfmp github.com/meinardmueller/libfmp
- PCP Notebooks www.audiolabs-erlangen.de/PCP

Biography of the Presenter

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 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 from 2010 to 2015, a member of the Senior Editorial Board of the IEEE Signal Processing Magazine (2018-2022), and a member of the Board of Directors of the 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.

Besides his scientific research, Meinard Müller has been very active in teaching music and audio processing. He gave numerous tutorials at major conferences, including ISMIR (2007, 2010, 2011, 2014, 2017, 2019), ICASSP (2009, 2011, 2019), Deep Learning IndabaX (2021), GI Jahrestagung (2017), Eurographics (2009, 2023), and ICME (2008). 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, 2015, www.music-processing.de). Recently, he released a comprehensive collection of educational Python notebooks designed for teaching and learning audio signal processing using music as an instructive application domain (https://www.audiolabs-erlangen.de/FMP).

Tutorial 6

Kymatio: Deep Learning Meets Wavelet Theory for Music Signal Processing

Cyrus Vahidi, Christopher Mitcheltree, Vincent Lostanlen

Abstract

We present a tutorial on MIR with the open-source Kymatio (Andreux et al., 2020) toolkit for analysis and synthesis of music signals and timbre with differentiable computing. Kymatio is a Python package for applications at the intersection of deep learning and wavelet scattering. Its latest release (v0.4) provides an implementation of the joint time—frequency scattering transform (JTFS), which is an idealisation of a neurophysiological model that is commonly known in musical timbre perception research: the spectrotemporal receptive field (STRF) (Patil et al., 2012). In the MIR research, scattering transforms have demonstrated effectiveness in musical instrument classification (Vahidi et al., 2022), neural audio synthesis (Andreux et al., 2018), playing technique recognition and similarity (Lostanlen et al., 2021), acoustic modelling (Lostanlen et al., 2020), synthesizer parameter estimation and objective audio similarity (Vahidi et al., 2023, Lostanlen et al., 2023).

The Kymatio ecosystem will be introduced with examples in MIR:

- Wavelet transform and scattering introduction (including constant-Q transform, scattering transforms, joint time–frequency scattering transforms, and visualizations)
- MIR with scattering: music classification and segmentation
- A perceptual distance objective for gradient descent
- Generative evaluation of audio representations (GEAR) (Lostanlen et al., 2023)

A comprehensive overview of Kymatio's frontend user interface will be given, with examples of extensibility of the core routines and filterbank construction.

We ask our participants to have some prior knowledge in:

- Python and NumPy programming (familiarity with Pytorch is a bonus, but not essential)
- Spectrogram visualization
- Computer-generated sounds

No prior knowledge of wavelet or scattering transforms is expected.

References

- Andreux, M., Angles, T., Exarchakisgeo, G., Leonardu, R., Rochette, G., Thiry, L., … & Eickenberg, M. (2020). Kymatio: Scattering transforms in python. The Journal of Machine Learning Research, 21(1), 2256-2261.
- Andreux, M., & Mallat, S. (2018, September). Music Generation and Transformation with Moment Matching-Scattering Inverse Networks. In ISMIR (pp. 327-333).
- Lostanlen, V., El-Hajj, C., Rossignol, M., Lafay, G., Andén, J., & Lagrange, M. (2021). Time–frequency scattering accurately models auditory similarities between instrumental playing techniques. EURASIP Journal on Audio, Speech, and Music Processing, 2021(1), 1-21.
- Lostanlen, V., Cohen-Hadria, A., & Bello, J. P. (2020). One or two components? the scattering transform answers. arXiv preprint arXiv:2003.01037.

- Lostanlen, V., Yan, L., & Yang, X. (2023). From HEAR to GEAR: Generative Evaluation of Audio Representations. Proceedings of Machine Learning Research, (166), 48-64.
- Muradeli, J., Vahidi, C., Wang, C., Han, H., Lostanlen, V., Lagrange, M., & Fazekas, G. (2022, September). Differentiable Time-Frequency Scattering On GPU. In Digital Audio Effects Conference (DAFx).
- Vahidi, C., Han, H., Wang, C., Lagrange, M., Fazekas, G., & Lostanlen, V. (2023). Mesostructures: Beyond spectrogram loss in differentiable time-frequency analysis. arXiv preprint arXiv:2301.10183.

Biographies of the Presenters

Cyrus Vahidi is a PhD researcher at the UKRI CDT in Artificial Intelligence and Music at the Centre for Digital Music, London and computer science graduate from Imperial College London. His research covers computational representations of auditory perception in machine listening and computer music. He is a core contributor to Kymatio, the open-source package for wavelet scattering. Previously, he was a visiting researcher at LS2N (CNRS, France) and worked on MIR/ML in ByteDance's SAMI group. He is the founder of Sonophase AI and performs experimental electronic music with Max/MSP and modular synthesis.

Christopher Mitcheltree is a PhD researcher at the UKRI CDT in Artificial Intelligence and Music at the Centre for Digital Music, London. He researches time-varying modulations of synthesizers / audio effects and is a founding developer of Neutone, an open-source neural audio plugin and SDK. In the past, he has worked on machine learning and art projects at a variety of different companies and institutions including: Google, Airbnb, AI2, Keio University, and Qosmo.

Dr. Vincent Lostanlen obtained his PhD in 2017 from École normale supérieure, under the supervision of Stéphane Mallat. Since then, he is a scientist (chargé de recherche) at CNRS and a visiting scholar at New York University. He is a founding member of the Kymatio consortium.

Papers – Session I

EXPLORING THE CORRESPONDENCE OF MELODIC CONTOUR WITH GESTURE IN RAGA ALAP SINGING

Shreyas Nadkarni [1] **Sujoy Roychowdhury** [1] **Preeti Rao**[1] **Martin Clayton**[2]

[1]Department of Electrical Engineering, Indian Institute of Technology Bombay, India

[2]Department of Music, Durham University, United Kingdom

prao@ee.iitb.ac.in, martin.clayton@durham.ac.uk

ABSTRACT

Musicology research suggests a correspondence between manual gesture and melodic contour in raga performance. Computational tools such as pose estimation from video and time series pattern matching potentially facilitate larger-scale studies of gesture and audio correspondence. We present a dataset of audiovisual recordings of Hindustani vocal music comprising 9 ragas sung by 11 expert performers. With the automatic segmentation of the audiovisual time series based on analyses of the extracted F0 contour, we study whether melodic similarity implies gesture similarity. Our results indicate that specific representations of gesture kinematics can predict high-level melodic features such as held notes and raga-characteristic motifs significantly better than chance.

1. INTRODUCTION

Manual gesturing by singers is an integral part of vocal music performances in the Indian classical traditions. Previous work has demonstrated that singers' gestures have several different referents and functions: for example, they may relate to the rhythmic structure of the music (marking a steady beat or tala cycle) or play a role in signalling to co-performers or audience members, as well as appearing to accompany or illustrate aspects of the melody being sung. In the latter case, hand movements sometimes appear to correspond to pitch height (i.e. ascending pitch co-occurs with one or both hands rising and/or moving to one side); at other times they relate to other aspects of melody, such as the tension felt while sustaining certain notes, or the image or abstract design visualised by the performer [1–5].

Little computational work has been carried out on gesture-to-audio correspondence in Hindustani vocal music. Paschalidou [6] carried out research on a motion capture dataset of solo alap recordings in the dhrupad genre, looking at a range of movement and audio features in relation to the concept of 'effort': although she found correspondences, generalising across performers proved challenging.

© S. Nadkarni, S. Roychowdhury, P. Rao and M. Clayton. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** 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 Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

Dataset	Singers	Ragas	Pakad	Alap	Dur(min)
Study in [7]	3 (1M,2F)	9	37	55	193
Current Work	11(5M,6F)	9	109	199	664

Table 1: A summary of the newly augmented audiovisual dataset compared with that of closest previous work [7].

Clayton et al [7] explored the use of movement data to classify 12-sec excerpts drawn from a corpus comprising 3 singers performing 9 common Hindustani ragas in the khyal genre. The use of solo alap meant the gestures cannot refer to either metric structure or interaction with co-performers, and thus relate predominantly to the melody of the ragas being presented. An inception block preceded by independently trained convolution layers for each of audio and gesture time series classification provided the best performance in the context of singer-dependent raga classification, especially reducing the confusion between melodically similar ragas with respect to the otherwise high-performing audio-only classification. While the work demonstrated the complementarity of gesture and melodic profiles in relation to raga identity, we are more interested in the present work in understanding which characteristics of gesture correlate with specific melodic characteristics. Further, given that the dataset of [8] was limited to 3 singers and therefore not suited to cross-singer studies, we present here a considerably enlarged corpus with 8 additional performers, collected following a similar methodology, as summarised in Table 1.

In the related Karnatak tradition, Pearson's research has looked at the role of gesture in vocal teaching [9]. The relation between the acoustics and the kinematics was studied in recent work by Pearson & Pouw using the tracking of left and right wrist positions [10]. They manually segmented the gesture tracks and studied the correspondence of various kinematic extrema with the temporally aligned changes in the acoustics (fundamental frequency, or F0, and amplitude envelope). A correspondence was established between the magnitudes of local peaks in acceleration and changes in F0, in line with previous work in co-speech gesturing [11].

In this work, we study the newly expanded corpus of

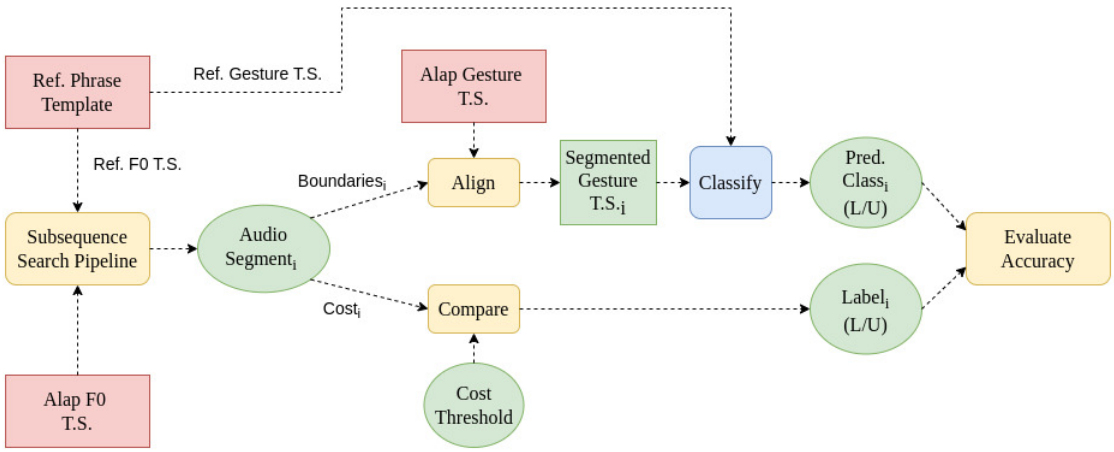


Figure 1: The overall testing and evaluation framework for the raga phrase-based segmentation. The audio and visual components of a candidate AV segment (i-th segment from an alap) are separately compared with the respective audio and visual components of a reference phrase segment (from a pakad) to see whether they are together consistent in their estimation of similarity with the reference phrase. We note that the gesture T.S. (time series) is multidimensional while the audio T.S. is a unidimensional sequence of F0 samples.

solo alap recordings. Since the same set of 9 ragas is performed by all the singers (11 in this case), we can explore commonalities in the gestures used by different singers for particular raga-specific melodic movements. That is, in contrast to the body of previous work, we use musically motivated units, implied by the raga melodic structure, to group the representations of melody and gesture. The aim of the study is to investigate correspondences between the singers' movements (captured in the time series for x- and y-coordinates of their wrist positions) and the melodies they sing (represented as F0 contours).

Figure 1 depicts our overall framework. The melodic phrase segments are obtained for each alap audio via a subsequence search using a reference audio template (such as a manually labeled phrase segment). The audio segment start and end times are then used to identify the corresponding time-synchronised video segment. The audio and video segments are individually processed to compute audio-based similarity and video-based similarity with respect to the corresponding components of the AV (audio-visual) reference template. We now seek to quantify the extent to which video-based similarity predicts audio similarity. We simplify the evaluation task to comparing, across the two modalities, the following binary labels: L (i.e. close to, or Like, the reference) or U (Unlike the reference).

In the next section, we provide the details of our dataset. This is followed by a discussion of the audiovisual segmentation methods. The experiments and results are presented in the final two sections of the paper.

2. DATASET AND PREPROCESSING

We consider our dataset of vocal alap performances by 11 professional musicians performing 2 alaps each of 9 ragas. Each alap is about 3 minutes long. The singers also contributed shorter 'pakad' recordings, rendering some of the key phrases of each raga in a brief format of a few seconds. The total duration of this newly expanded dataset (summarised in Table 1) is about 11 hours. Each piece was recorded using three video cameras and separate microphone; only the central camera is used in the current analyses. While each alap is labeled only by singer and raga, we carry out further manual annotation of the pakad audio files for selected raga phrases as used in this study. That is, all the pakads of a given raga across the 11 singers are searched for instances of the desired phrase (e.g. gmD in raga Bageshree). This task, carried out by a musician, is facilitated by the fact that the pakad is almost always sung with solfege (unlike the alap).

Our audio and video processing pipelines closely follow those of [7]. An initial stage of audio suppression of the background drone is obtained via source separation [12]. The suppression, while not complete, is adequate for the reliable estimation voicing and pitch at 10 ms intervals using monophonic pitch detection based on short-time autocorrelation analysis [13]. Brief unvoiced regions (less than 400 ms) arising from short breath pauses and consonant utterances are filled in via cubic spline interpolation to obtain the continuous pitch contours associated with melodic movements that are bounded by silence (>400 ms) on both ends. These are termed 'Silence-Delimited Segments' (SDS). The pitch contour is tonic-normalised using an automatically detected (and manually verified) tonic to obtain the F0 (cents) contour [14].

In order to extract the movement data, the central video view of each piece is processed using the OpenPose pose estimation algorithm, which generates x- and y-coordinates for 11 upper body joints [15]. We select the right and left wrist coordinates. Any missing data are interpolated and each of the time series is low-pass filtered to remove jitter. The position time-series, originally sampled at 25 fps is interpolated to 100 samples/sec to synchronise

it with the sampled F0 contour. Other important low-level human motion descriptors include velocity (rate of change of the 2d position) and acceleration (rate of change of the velocity) [16]. We derive velocity and acceleration profiles from the 2d position time-series of each joint by computing derivatives. A robust estimate of the derivative is obtained via a differencing kernel such as a biphasic filter with its controllable smoothing parameters [17, 18]. We find that a 101-point filter achieves a lowpass filtering of about 2 Hz, giving a sufficiently smooth and physiologically plausible movement acceleration profile [19]. We eventually obtain the 8-dim gesture time series of position (x and y), velocity and acceleration for each of the left and right wrists for each of the singer-alap and pakad recordings. In this, we include the synchronized F0 contour to get the complete audiovisual time series for an alap, now in the form of a sequence of SDS. A detailed review of the data collection and processing appears in the supplementary material.

3. SEGMENTATION METHODS

The first stage of segmentation of the synchronised AV time series comprises the silence-delimited segments (SDS) obtained in the previous section. We discard segments of duration less than 500 ms as too limited for our further analyses. The retained SDS, numbering approximately 30 per alap, have a mean duration of 5.2 s with less than 1% (of the total count of 6012) exceeding 20 s. As discussed next, we apply melodic segmentation principles to each SDS to obtain stable note and raga-characteristic melodic movements or phrases that can help us explore the links between specific musical expressions and the corresponding gestures.

In a top-down approach, the alap can be segmented into its phrases. A raga phrase, although notated simply by its solfege sequence, has a melodic-rhythmic realisation comprising specific intonations and durations of its constituent svaras, together with the transitions to/from neighbouring svaras [20]. On the other hand, in a bottom-up approach, the melodic contour can be viewed as comprising the following broad categories of segments: stable notes, and the transitions between the notes which can include distinctive melodic ornaments such as glides (meend) and oscillatory movements (andolan) apart from steep changes of pitch or pauses [21]. Figure 2 presents an example of an SDS that comprises a variety of stable and transitional sub-segments. It is therefore of interest to examine audiovisual correspondences in the context of the distinct types of melodic movements. The two different audio-based segmentation procedures are detailed next.

3.1 Stable note segmentation

To identify occurrences of stable or sustained tones, the continuous F0 contour corresponding to an SDS is searched for instances in which the same raga note (svara) is sustained for > 250 ms. That is, a stable note is defined as a region where the F0 lies within a 25 cent interval of the mean intonation of the raga note. This is based on

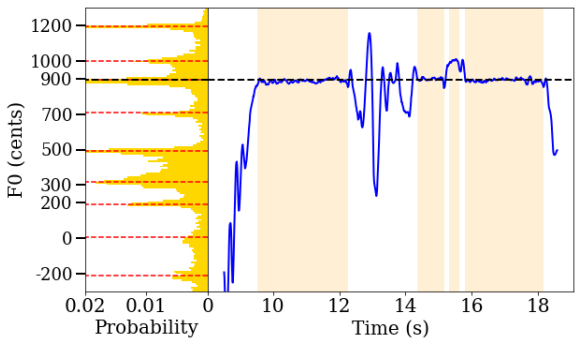


Figure 2: A sample SDS with identified steady notes (shaded regions of blue F0 contour) and pitch salience distribution (on the left) computed from the entire alap audio with detected svara locations highlighted.

past work that associated the similar duration and intonation parameters with a listener's percept of a held note [22]. Further, given that a svara may not be realised on the equitempered grid but rather with a raga-specific intonation, we use a finely binned pitch salience distribution computed across the alap to establish the svara locations [22]. Stable note regions corresponding to the same svara that are separated by less than 100 ms are next merged. The boundaries of the so detected stable notes are shown in the example of Figure 2. Across our alap dataset, stable notes were found to range from 0.25 s to 9.9 s with a mean of 0.73 s.

In a similar vein, we considered the segmentation of another characteristic melodic movement, the glide (or slide). This has been attempted previously via the quality of a linear fit to the F0 contour for Indian popular vocal music [23]. However we found that the variety and complexity of glide movements in raga music make it challenging to develop a universal glide detection algorithm. We therefore resort to template-based phrase detection for the purpose, as explained next.

3.2 Phrase-based segmentation

As depicted in Table 2, the raga motifs selected for our exploration include a distinctive upward slide of an augmented fourth in Shree, a falling slide of a fourth in Nand, and a three-note ascending phrase in Bageshree. The chosen phrases are highly characteristic of the corresponding raga and occur in the raga alap with relatively unchanged melodic shape, prompting the question about whether their gesture executions also bear some measurable similarity. The corresponding pakad phrases serve as templates for the segmentation of the alaps for the chosen raga across the 11 singers. We obtain a number of templates of the given phrase from across the 11 singers' pakads. The set of templates represents the diversity in the realization of the phrase across and within singers. This is manually reduced to a set of 6 templates per phrase while retaining the diversity. Figure 3 shows a few examples for each of the phrases chosen for the current study. We observe that the simple notation used to represent the up or down slide (/,

Raga	Svara (Notes)	Phrase
Bageshree	S R g m P D n	gmD
Shree	S r G M P d N	r/P
Nand	S R G m M P D N	P\R

Table 2: The ragas and phrases used in this study. The svaras S r R g G m M P d D n N correspond to the 12 notes of the Western chromatic scale with S representing the tonic. The symbols / and \ denote the upward and downward slide respectively [24], [25].

\) belies the complexity of contour shapes defined by raga grammar. Also clear are the essential shape features that point to the need for dynamic time warping (DTW) based comparisons [26]. Next, the following steps (also visualised in Figure 4) lead to the desired segmentation of the alap audio files for each selected raga phrase.

1. The six phrase templates from the pakads are warped to the same target length (that of the 3rd template in increasing length order in the set) using a penalty parameter that discourages large deviations from the diagonal path. This helps to ensure that the subsequence DTW matching costs can be meaningfully compared across the templates.

2. As shown in the middle panel of Figure 4, constrained DTW based subsequence search is executed on each SDS with each of the 6 warped audio templates (WAT) to obtain for each WAT the lowest cost match that satisfies a duration criterion (> 0.5s) in order to avoid cases of pathological warping [27]. Such matches are accepted as valid and stored with the cost, temporal boundaries and WAT index. In case no valid match is returned (in the top 20 retrieved responses) for a particular template, that SDS-template is not considered further. This step leaves us with between 1 to 6 best matched segments per SDS along with the associated DTW costs.

3. Next, we pick the single lowest cost for each SDS and use this value to cluster the entire set of SDS, across 22 alaps of the raga, into 2 clusters by fitting a kernel density estimate (KDE) to the distribution of costs as shown in Figure 5 [28]. The cost value coinciding with the lowest point in the valley between the peaks is used as a cost threshold to label each SDS as one of the two classes:'Like' (i.e. similar to the raga motif) and 'Unlike' (different from the raga motif). These are the labels we would like to predict from the corresponding gesture time series segments in the context of our investigation of audiovisual correspondence.

4. In order to increase the number of examples for the gesture-based prediction task, we club all the different template matches obtained in Step 2 for the same SDS under the same label. This was justified by our observation that the SDS labeled Like (L) in Step 3 typically exhibited similar low cost matches across all templates of the same phrase. The SDS marked Unlike (U), on the other hand, exhibited a relative wide spread in cost values above the threshold, similar to that depicted in Figure 5.

Finally, with the audio segments computed in this section, we extract the corresponding temporally synchronised gesture time series for each SDS and WAT pair. In the next section, we report our experiments on testing various kinematic features for the prediction of the corresponding audio-derived labels in our two distinct tasks.

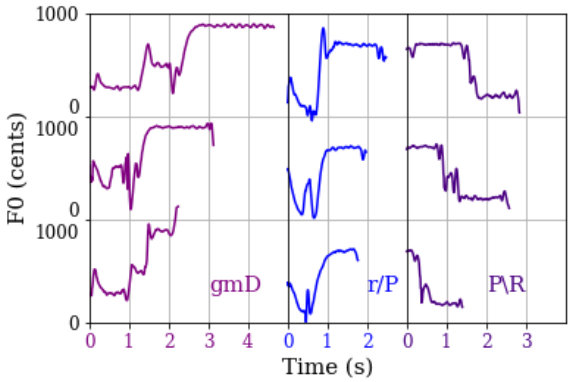


Figure 3: Sample templates for each of the three phrases: gmD (purple), r/P (blue) and P\R (violet).

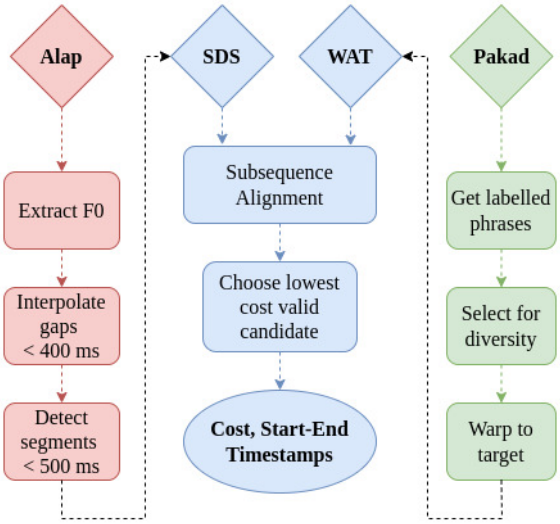


Figure 4: The pipeline for phrase-based segmentation using alap and pakad audio data. For warping the pakad phrase templates, a window size of 100 and a penalty of 200 was chosen, while for the subsequence alignment, K = 20 and penalty = 0.1 were chosen [27].

4. EXPERIMENTS

Past work on gesture kinematics in the context of speech and singing co-gesturing has considered velocity and acceleration parameters rather than the raw position time series with these parameters relating more directly to human effort or force [10, 29–31]. We therefore include the (x,y) position of each wrist as well as the corresponding velocity and acceleration profiles across the segment as input features for our two classification tasks. Summarising the previous section, the gesture time series is segmented based on the previously obtained audio segment boundaries giving us a time-aligned multidimensional time series

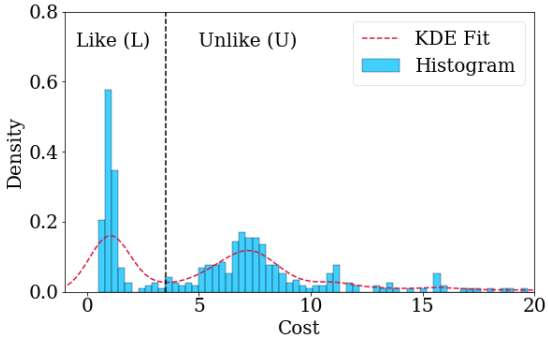


Figure 5: Distribution of the DTW subsequence cost across the SDS of all singer alaps for the best matched audio phrase template for P\R of raga Nand. The dashed vertical line shows the threshold derived from the KDE fit (dashed contour), using which the SDS are labeled as Like and Unlike with reference to the template phrase.

for (i) each stable-note and non-stable segment across all the alaps in the dataset, and (ii) each pakad phrase gesture template and its audio-matched gesture segment from the SDS. We consider supervised classification for each task with the different features as discussed next.

4.1 Stable-note prediction

Stable note segments were labeled as such based on the F0 variation across the segment as discussed in Section 3.1. We would like to investigate whether there is any consistency in the gesture kinematics corresponding to stable note regions. We implement a binary classifier trained and tested on the dataset of labeled stable notes and the (complementary) non-stable regions where the training and test data are both drawn from across singers and ragas. Although 250 ms regions of stable pitch qualified as stable notes, we restricted the examples of both categories used in this experiment to those with a minimum duration of 0.5 s in order to ensure that the training dataset was relatively balanced. When the segment duration was constrained to the range 0.5 s to 5 s, the stable notes constituted 39% of the total examples, with across-singer variability as captured in row 2 of Table 3.

Postulating that gesture kinematics are relatively more subdued during the stable note events, we investigate a simple set of explicit features using a Support Vector Machine (SVM) classifier. We compute the statistical aggregates of each of the velocity and acceleration in the form of the mean and variance across the duration of the corresponding time series segment. We thus have 4 features per wrist (i.e. 8 features in all) for the binary classification of segments into stable and non-stable pitch events. We carry out 10-fold cross-validation on the dataset and report the F1 score for the detection of stable notes with the SVM hyperparameters tuned to maximise the average performance across the folds. This exercise is carried out on the entire dataset as well as on singer-specific datasets, where the corresponding counts of examples are provided in Table 3.

4.2 Raga phrase detection

Our goal is to determine whether the L and U labels (that were assigned based on audio proximity) can be predicted by gesture alone at better than chance (i.e. based only on the priors) and, if so, which kinematic features are most useful in this task. Our measure of similarity is the DTW distance computed between the template and test (i.e.the alap SDS subsequence) time series. In the context of our alap gesture time series, already segmented based on the audio phrase matching, we now compute DTW distance between the multidimensional reference and candidate under test.

Multidimensional time series present us with some distinct options for the distance computation. Two obvious approaches are DTW_I and DTW_D depending on whether the individual time series are each warped independently or whether they are all forced into a single warping path [32]. The use of DTW_D appears meaningful for the incorporation of the velocity and acceleration contours derived from the corresponding position time series of the wrists. However, it is interesting also to test with independent DTW costs across the separate time series (to get an 8-dim feature vector of costs) to see if this helps reduce the effect of the less informative features, if any. We term this DTW_{IND}. Further, decoupling the left and right wrists to obtained two differently warped sets of time series (DTW_{LR}) is also perfectly meaningful in the current task.

With DTW cost(s) as the input features, we create 5 train-test splits with the uniform distribution of singers across the splits. Thus every example appears once in the test set. We train a logistic regression classifier with L2 regularizer and use 3-fold cross-validation within the train set to learn the best set of parameters.

5. RESULTS AND DISCUSSION

5.1 Stable-note detection

Table 3 presents classifier performance in terms of the F1 score for the retrieval of stable notes. We restrict ourselves to the set of labeled segments of duration between 0.5 s and 5 s, with 20897 examples in all. With 38.9% of these corresponding to stable notes, we find that the obtained F1 score is 65.7% when considering the overall dataset across singers and ragas. Given the known high singer-dependence of gesturing, we also evaluate singer-specific classification with the same kinematic features, now restricted to training and validation (10-fold CV as before) on the smaller dataset of each singer's alaps across ragas.

As anticipated, we note a large variation in the F1 scores across singers in Table 3 but with all values considerably above chance (which equals the corresponding % Stable entry in row 2). While some of the variation could be attributed to the differences in distributions of labels across the singers' datasets, we observe variation even across singers with similar distribution characteristics (such as AP and SM, for instance).

As for the singers with F1 scores well below the across-singers stable note detection F1 score (such as the case of

Singer	**All**	**AG**	**AK**	**AP**	**CC**	**MG**	**MP**	**NM**	**RV**	**SCh**	**SM**	**SS**
Count	20897	1242	1987	2382	2274	1822	2111	1769	1563	2069	2083	1595
% Stable	38.9	53.6	36.7	44.1	34.0	51.5	47.3	32.8	34.7	22.5	43.6	30.1
F1 Score (%)	65.7	81.1	63.6	69.5	68.2	72.5	71.6	65.8	65.2	60.5	75.1	49.2

Table 3: Overall and singer-specific performances for stable note detection from segmented gesture time series across the set of instances in the duration range [0.5, 5] s. Count indicates the number of instances in each singer (or overall) dataset. The F1 scores in the final row may be compared with the values in the row 2 that correspond to the chance-level F1 score.

Phrase	**Like**	**Unlike**	**Chance Accuracy**	**DTW_D (1)**	**DTW_I (1)**	**DTW_{Ind} (8)**	**DTW_{LR} (2)**
gmD	944	827	50.2	52.2	48.6	51.8	**52.4**
r/P	1035	1268	50.5	**55.3**	47.1	**56.1**	**55.1**
P\R	817	1340	53.0	**65.0**	45.7	**65.2**	**65.1**

Table 4: Classification accuracy (%) for Like and Unlike phrase detection with gesture time series and different DTW distance measures. Feature dimensionality (i.e. DTW path costs) appears in parantheses. Bold font indicates that the model performance is significantly better ($p<0.005$) than chance, with the chance accuracy (%) also mentioned in the table.

SCh and SS), we note the relatively low proportions of stable notes in their data. Such behaviour can arise, for example, when the singer makes a choice to focus more on melodic movements in their alap rather than long periods of held notes. With a relatively low representation of their stable note examples in the training data, it is probable that idiosyncratic aspects of their stable note gestures, if any, were not learned by the classifier. We did not find much of raga dependence in stable note detection performance. We also did an analysis of tonic versus other stable notes to find that the tonic notes (fewer in number overall) were harder to detect; this observation needs more data for a better understanding.

5.2 Raga-phrase detection

Table 4 displays the Like/Unlike classification of raga phrases across the alaps of all singers. We see a roughly equal proportion of L and U examples and therefore chance baseline accuracies close to 50%. Both P\R and r/P exhibit gesture classification accuracies that are statistically better than chance for all versions of DTW distance except the DTW_I which is the simple summing of independent path costs across the 8 different series. In the case of the gmD phrase, we see a relatively small increase over chance with the only significant difference provided by the DTW_{LR} that combines left and right wrist paths, each computed independently of the other. A singer-based breakdown of the overall accuracy showed relatively uniform behaviour across singers for all the phrases except for one outlier (out of the 11) for each of P\R and r/P phrases.

We would also like to comment on the equal proportion of L and U examples in our data for this task. Although there is a far larger number of U instances (that is alap segments that probably do not contain the phrase of interest and therefore expected to return a high cost in the DTW subsequence search of the audio), we found that many of these actually led to invalid paths from pathological warping and thus were unusable candidates for this study.

6. CONCLUSION

This work proposed a new approach to examining melodic similarity captured in co-singing gestures by analysing audiovisual recordings. With a new dataset of 11 singers, raga-characteristic phrases were proposed as a proxy for similar melodic movements within and across singers. As in previous work, wrist movements that accompanied the solo alap singing were represented as kinematics time series. In the absence of ground-truth phrase labels for the alap data, we developed a pipeline for achieving the AV segmentation for the chosen phrases via DTW-based audio template matching using a small set of hand-labeled segments. We also considered the classification task for more generic AV segments defined in a bottom-up manner such as stable-note regions. Overall, our experimental results indicate that there is significant kinematic information linked to the selected melodic events, and confirm the importance of computed velocity and acceleration profiles in the gesture representation.

A useful contribution of this work is the musicological questions it encourages. Apart from the aspects already mentioned in the discussion of the results, we note that the use of multiple phrase templates can facilitate larger experimental validation of hypotheses, such as that of Rahaim [5], that gestures could function to draw attention to *what is different* between two semantically close melodic patterns. Finally, several enhancements to the presented methods are possible including better-motivated movement features, more keypoints (elbow and hand joints) and using all 3 camera views to include depth movement.

Suppl. material: https://dap-lab.github.io/audioGestureCorrespondence/

The authors S. Nadkarni and S. Roychowdhury contributed equally to this work.

7. REFERENCES

[1] M. Clayton, "Time, gesture and attention in a khyāl performance," *Asian Music*, vol. 38, no. 2, pp. 71–96, 2007.

[2] L. Leante, "The lotus and the king: Imagery, gesture and meaning in a hindustani rāg," *Ethnomusicology Forum*, vol. 18, no. 2, pp. 185–206, 2009.

[3] Leante, "Gesture and imagery in music performance: Perspectives from north indian classical music," in *The Routledge Companion to Music and Visual Culture*. Routledge, 2013, pp. 145–152.

[4] L. Leante, "The cuckoo's song : imagery and movement in monsoon ragas," in *Monsoon feelings : a history of emotions in the rain*, I. Rajamani, M. Pernau, and K. R. B. Schofield, Eds. New Delhi: Niyogi Books, 2018.

[5] M. Rahaim, *Musicking Bodies: Gesture and Voice in Hindustani Music*. Wesleyan University Press, 2012.

[6] 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.

[7] 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.

[8] M. Clayton, J. Li, A. R. Clarke, M. Weinzierl, L. Leante, and S. Tarsitani, "Hindustani raga and singer classification using pose estimation," 2021. [Online]. Available: https://doi.org/10.17605/OSF.IO/T5BWA

[9] L. Pearson, "Gesture and the sonic event in karnatak music," *Empirical Musicology Review*, vol. 8, no. 1, pp. 2–14, 2013.

[10] 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.

[11] W. Pouw *et al.*, "A kinematic-acoustic analysis of gesture-speech coupling in persons with aphasia," 2021.

[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*, vol. 5, p. 2154, 2020.

[13] Y. Jadoul, B. Thompson, and B. de Boer, "Introducing parselmouth: A python interface to praat," *Journal of Phonetics*, vol. 71, pp. 1–15, 2018.

[14] D. Bogdanov, N. Wack, E. Gómez, S. Gulati, P. Herrera, O. Mayor *et al.*, "Essentia: an audio analysis library for music information retrieval," in *Proc. of the 14th Int. Soc. for Music Information Retrieval Conference*, Curitiba, Brazil, 2013.

[15] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "Openpose: Realtime multi-person 2d pose estimation using part affinity fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 1, pp. 172–186, 2021.

[16] C. Larboulette and S. Gibet, "A review of computable expressive descriptors of human motion," in *Proceedings of the 2nd International Workshop on Movement and Computing*, 2015, pp. 21–28.

[17] D. J. Hermes, "Vowel-onset detection," *The Journal of the Acoustical Society of America*, vol. 87, no. 2, pp. 866–873, 1990.

[18] P. Rao, T. P. Vinutha, and M. A. Rohit, "Structural segmentation of alap in dhrupad vocal concerts," *Transactions of the International Society for Music Information Retrieval*, vol. 3, no. 1, 2020.

[19] W. Pouw, J. de Wit, S. Bögels, M. Rasenberg, B. Milivojevic, and A. Ozyurek, "Semantically related gestures move alike: Towards a distributional semantics of gesture kinematics," in *Digital Human Modeling and Applications in Health, Safety, Ergonomics and Risk Management. Human Body, Motion and Behavior: 12th International Conference, DHM 2021, Held as Part of the 23rd HCI International Conference, HCII 2021, Virtual Event, July 24–29, 2021, Proceedings, Part I*. Springer, 2021, pp. 269–287.

[20] K. Ganguli and P. Rao, "A study of variability in raga motifs in performance contexts," *Journal of New Music Research*, vol. 50, pp. 1–15, 02 2021.

[21] W. Van der Meer, *Hindustani music in the 20th century*. Springer Science & Business Media, 2012.

[22] K. K. Ganguli and P. Rao, "On the distributional representation of ragas: Experiments with allied raga pairs," *Transactions of the International Society for Music Information Retrieval*, vol. 1, no. 1, 2018.

[23] C. Gupta and P. Rao, "An objective assessment tool for ornamentation in singing," in *Proceedings of the International Symposium of Frontiers of Research on Speech and Music and Computer Music Modelling and Retrieval*, 2011.

[24] "Music in motion, the automatic transcription system for indian music," https://autrimncpa.wordpress.com/, note = Last Accessed: 2023-04-14.

[25] S. Kulkarni, *Shyamrao Gharana*. Prism Books Pvt. Ltd, 2017, vol. 1.

[26] M. Müller, *Fundamentals of Music Processing*. Springer, 2015.

[27] T. V. C. . P. R. Wannes Meert, Kilian Hendrickx, "Dtaidistance (version v2)," last Accessed: 2023-04-14. [Online]. Available: http://doi.org/10.5281/zenodo.5901139

[28] S.-T. Chiu, "Bandwidth selection for kernel density estimation," *The Annals of Statistics*, pp. 1883–1905, 1991.

[29] R. C. Madeo, C. A. Lima, and S. M. Peres, "Gesture unit segmentation using support vector machines: segmenting gestures from rest positions," in *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, 2013, pp. 46–52.

[30] W. Pouw and J. A. Dixon, "Quantifying gesture-speech synchrony," in *the 6th gesture and speech in interaction conference*. Universitaetsbibliothek Paderborn, 2019, pp. 75–80.

[31] Y. Ferstl, M. Neff, and R. McDonnell, "Expressgesture: Expressive gesture generation from speech through database matching," *Computer Animation and Virtual Worlds*, vol. 32, no. 3-4, p. e2016, 2021.

[32] M. Shokoohi-Yekta, B. Hu, H. Jin, J. Wang, and E. Keogh, "Generalizing dtw to the multi-dimensional case requires an adaptive approach," *Data mining and knowledge discovery*, vol. 31, pp. 1–31, 2017.

TRIAD: CAPTURING HARMONICS WITH 3D CONVOLUTIONS

Miguel Perez[♯♭] **Holger Kirchhoff**[♯] **Xavier Serra**[♭]
Huawei, Munich Research Center[♯] MTG, Universitat Pompeu Fabra[♭]
miguel.perez.fernandez@huawei.com xavier.serra@upf.edu

ABSTRACT

Thanks to advancements in deep learning (DL), automatic music transcription (AMT) systems recently outperformed previous ones fully based on manual feature design. Many of these highly capable DL models, however, are computationally expensive. Researchers are moving towards smaller models capable of maintaining state-of-the-art (SOTA) results by embedding musical knowledge in the network architecture. Existing approaches employ convolutional blocks specifically designed to capture the harmonic structure. These approaches, however, require either large kernels or multiple kernels, with each kernel aiming to capture a different harmonic. We present TriAD, a convolutional block that achieves an unequally distanced dilation over the frequency axis. This allows our method to capture multiple harmonics with a single yet small kernel. We compare TriAD with other methods of capturing harmonics, and we observe that our approach maintains SOTA results while reducing the number of parameters required. We also conduct an ablation study showing that our proposed method effectively relies on harmonic information.

1. INTRODUCTION

When a note is played, a set of strongly related frequencies start to sound leading to a pitch sensation for the listener. These strongly related frequencies are what we call the *harmonic spectrum*, in which we distinguish two parts: the fundamental frequency (f_0) and the harmonics. The fundamental is the frequency associated with the pitch, and the harmonics are integer multiples of f_0. Different instruments reinforce different harmonics, achieving different timbres; but the underlying structure created by f_0 and its harmonics remain present.

Traditional Automatic music transcription (AMT) systems based on manual feature design employed this property to look for harmonic patterns given an observed spectrogram [1]. When DL became more popular, many researchers refrained from incorporating expert knowledge into their model architectures, but relied on generic models in combination with large amounts of task-specific training data. Even though these systems significantly outperformed traditional approaches, models utilized large numbers of parameters. [2, 3].

The number of parameters plays an important role, as more parameters can help capture the harmonic pattern better; in exchange, larger models require more computing resources as the number of operations grows. Many DL practitioners do not always have access to large GPU clusters, and might not be able to train such large models. Moreover, many portable devices such as phones have limited battery and memory, and such large models in those devices will either quickly drain their battery or be directly impossible to employ. Part of the research focused on reducing the number of models' parameters without harming the transcription's accuracy. This was achieved in many cases through the incorporation of pitch expert knowledge within the architecture neural network (NN) [4–9].

The main challenge resides in the unequal distances between harmonics in the spectrum, so previous approaches employ either large kernels or several ones running in parallel. This paper introduces a ***tri***dimensional kernel h***a***rmonically ***d***ilated (TriAD), a neural block that captures music intervals and is capable of observing multiple harmonics while using a single yet small kernel.

The rest of the paper is divided into the following sections: Section 2 gives more details about prior work capturing harmonics from the spectrum. Section 3 describes our method, including the processing of the signal and the design of the kernels. The experimental setting is described in Section 4. We present the results for these experiments as well as an ablation study in Section 5. Finally, Section 6 contains our conclusions for this paper and future work.

2. RELATED WORK

As mentioned in Section 1, harmonics played an important role in the first AMT systems. For example, [1] creates a dictionary of sets of expected harmonics for each fundamental. These ideal patterns were then matched to the spectrograms used as input for the system using the non-negative least squares (NNLS) algorithm. The result is an estimation of fundamental frequencies that along with their respective harmonics, would resemble the input's spectrogram.

For AMT systems using DL, prior work has incorporated domain-specific knowledge in two ways: 1. by choosing a custom input representation that allows the model to detect harmonic structures [4, 10, 11]; 2. by employing specific network architectures to search for pat-

© M. Perez, H. Kirchhoff, and X. Serra. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).
Attribution: M. Perez, H. Kirchhoff, and X. Serra, "TriAD: Capturing harmonics with 3D convolutions", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

terns in a given feature map obtained at any point of the network [6–8, 12]. Within the first category, one of the most popular approaches is the harmonic constant Q transform (HCQT) [4], a feature that extends the constant Q transform (CQT) [13]. The standard CQT returns a log-frequency representation of the spectrum, where the nth bin is associated with the frequency $f_n = f_{min} \cdot 2^{n/p}$ where f_{min} is the minimum frequency to be considered, and p is the number of bins per octave. The magnitude of CQT spectrogram is a representation containing a single channel, F_{bins} frequency bins, for T frames; its shape is $[1, F_{bins}, T]$. The HCQT extends the CQT the channel dimension, where now H harmonics are aligned, resulting in a tensor with dimensions $[H, F_{bins}, T]$. This extension is done by stacking a number of H CQTs through the channel dimension. Each one of these H CQTs is a regular one whose f_{min} has been scaled by a harmonic factor h: $f_n = h \cdot f_{min}$; the CQTs with $h = 1$ will refer to the fundamental, $h = 2$ will refer to the first harmonic, $h = 3$ to the third harmonic, etc. up to H different values. Similarly, sub-harmonics can be added by making $h = 0.5, 0.25$, etc. In a nutshell, the HCQT facilitates information about the fundamentals directly at the network's input.

As mentioned, other works incorporated the harmonic knowledge within the architecture of NNs, e.g. [6] extended the idea of frequency-shifted representations, for the internal feature maps obtained inside NNs. The authors named this method multiple rates dilated harmonic causal convolution (MRDC-Conv). Let $\mathcal{X}$ denote a feature map, with shape $[C_{in}, F_{bins}, T]$ at an arbitrary point of the network. The number of channels for that map is C_{in}. In a CQT spectrum, the distance d_n between the fundamental frequency and the nth harmonic is given by:

$$d_n = \text{round}(p \cdot \log_2(n)) \tag{1}$$

Where p is a parameter that determines the number of bins per octave in the CQT spectra. To capture k harmonics with MRDC-Conv, the feature map $\mathcal{X}$ is convolved with k different kernels in parallel, resulting in k outputs. Each of the outputs is shifted following the harmonic factors given by Equation 1. E.g. to capture the first three harmonics, three different kernels are required, thus, producing three different outputs. In the case of $p = 12$ and following Equation 1, the shifts associated with the $2^{nd}, 3^{rd}$ and $4^t h$ harmonics are 12, 19, and 24. The sum across the k outputs is taken, leading to a single final output of shape $[C_{out}, F_{bins}, T]$, where C_{out} is the number of output channels. This method is illustrated in Figure 1a. MRDC-Conv achieves a convolution able to observe the input at the precise position of the harmonics; its drawback is that for each of the harmonics, a different kernel is needed, thus requiring a different feature map stored in memory for each of the k harmonics before they can be aggregated.

Some other authors embedded harmonic knowledge within the convolutional kernels rather than in the manipulation of their inputs/outputs. In [12] the authors use sparse convolutions so that only relevant parts of the spectrum are considered. Sparse convolutions allow the kernels to "ignore" certain parts of the input, so they do not contribute either to the output or to backpropagation during training [14]. According to [15], the harmonics are *positive* indicators that a certain pitch is present, but some frequencies indicate that the pitch might not be present at all. The latter are called *negative* indicators. The sparse convolutions from [12] are used in such a way that only *positive* and *negative* indicators defined in [15] are taken into account. Sparse convolutions require nonetheless using large kernels to cover relevant parts of the spectrum, i.e. [12] resulted in around 650k parameters exclusively for harmonic processing, accounting for the major portion of the model's parameters.

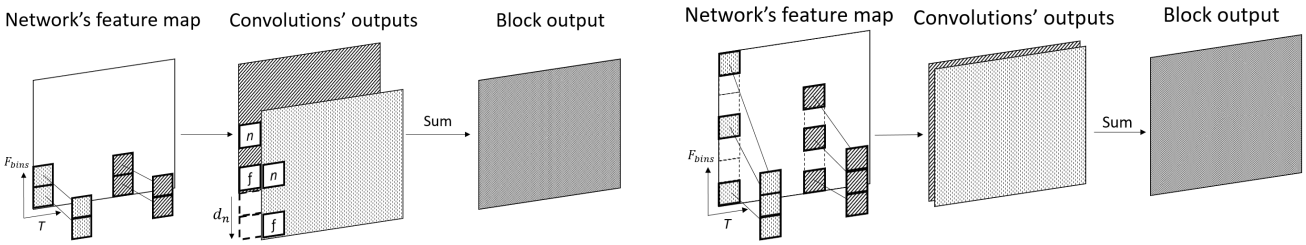
Harmonics	Music Interval	pitc class distance
$2, 4, 8, 16$	octave	$b \cdot 12$
17	minor second	$b \cdot 1$
$9, 18$	major second	$b \cdot 2$
19	minor third	$b \cdot 3$
$5, 10, 20$	major third	$b \cdot 4$
21	perfect fourth	$b \cdot 5$
$11, 22$	augmented fourth	$b \cdot 6$
$3, 6, 12, 24$	perfect fifth	$b \cdot 7$
25	minor sixth	$b \cdot 8$
27	major sixth	$b \cdot 9$
$7, 14, 28$	minor seventh	$b \cdot 10$
$15, 30$	major seventh	$b \cdot 11$

Table 1: The harmonics of the first 3 octaves, and their associated music intervals. The rightmost column indicates the distance in bins associated with each interval, where b is the number of bins per semitone.

In [8], dilated convolutions are used to capture the harmonics from the spectrum, with a method named harmonic dilated convolution (HD-Conv). Dilated convolutions are a special kind of convolution, where the kernels' inputs are spaced by a fixed amount. An example of dilated convolutions can be seen in Figure 1b. By controlling the dilation size, the authors space the kernels' inputs, so each kernel obtains a specific harmonic. The outputs of the different kernels are aggregated by summing across the kernels' outputs as shown in Figure 1b. The size of the dilations is given by Equation (1). E.g. for $p = 12$, the second harmonic is separated from the fundamental by $d_2 = 12$ bins, the third one by $d_3 = 19$; to capture both the second and the third harmonic, we would need to create two convolutional kernels with a dilation size of 12 and 19 at the frequency dimension. This method has the same drawback as MRDC-Conv, as different harmonics also require a different kernel.

3. OUR METHOD

Similarly to [8], our method uses dilated convolutions to capture the harmonics of the spectrum. As mentioned before, a constant dilation can not capture multiple harmonics given the logarithmic nature of these. If it was possible to use different dilations for the same kernel, this problem



(a) MRDC-Conv

(b) HD-Conv [8]

Figure 1: Figure (a) An example of MRDC-Conv [6]. Two kernels are applied to the same input. The fundamental f is separated from the harmonic n by d_n bins. One output gets shifted by d_n, and so f and n get aligned. Figure (b) An example of HD-Conv [8], with two kernels applied to the same input, each one with a different dilation (3, and 2 respectively).

would have been already solved, but currently, DL frameworks support only dilations with constant spacing. Our method is able to partially overcome this technical limitation and achieve a convolution at the frequency axis with different dilation rates; thanks to this, our proposed method captures multiple harmonics by just using a single kernel.

We named our method *TriAD*, and it involves a series of steps. The first step is to split the frequency dimension into two new ones, each representing different octaves and pitch classes. We call this representation the *pitch/octave* spectrogram. Next, we create the kernels for our method. Previous works used kernels spanning 2 dimensions: frequency and time; our method's kernels however span 3 dimensions: octave, pitch class, and time. An arbitrary number of m different kernels can be created, each one capturing a different music interval. The m kernels are convolved with the previously described pitch/octave spectrogram, resulting in m different outputs. Finally, these outputs are aggregated by taking the sum across them. The consecutive steps are illustrated in Figure 2.

Subsection 3.1 details the procedure followed to convert a log-frequency spectrogram onto a pitch/octave spectrogram. Subsection 3.2 explains how our convolutional kernels are created and the difference they have with the method described in [8]. At the end of that subsection, we describe a special kind of padding used in our technique, the octave-circular padding.

3.1 The pitch/octave spectrogram

Let $\mathcal{X}^{C_{in} \times F_{bins} \times T}$ be a feature map, with F_{bins} logarithmically spaced frequency bins, T frames, and C_{in} channels. Our goal is to separate octave and pitch class information. We split the F_{bins} bins into two dimensions representing the octave (o) and pitch class (p) information. The number of pitch classes is simply the number of bins per octave used, and the number of octaves can be obtained by $o = \frac{F_{bins}}{p}$. Note that o must be an integer, and so when this condition is not met, we pad the upper part of $\mathcal{X}$'s frequency dimension with the minimum amount of zeros that satisfies the condition. The result is the pitch/octave spectrogram $\mathcal{Y}^{C_{in} \times o \times p \times T}$, a view of $\mathcal{X}$ where F_{bins} has been separated into its octave and pitch class information.

3.2 The harmonic convolutions

Our aim is to compare two pitch classes across multiple octaves to capture harmonically related information. As shown in Table 1, harmonics and music intervals are closely related. Comparing two pitch classes separated by a certain interval at multiple octaves simultaneously will effectively obtain the harmonics associated with that music interval.

As previously mentioned, our kernels have 3 dimensions: $K^{k_o \times k_p \times k_t}$, related to the octaves (k_o), pitch classes (k_p), and frames (k_t) of the pitch/octave spectrogram; this means that our method uses 3D convolutions [1]. By changing the convolution dilation at the pitch class dimension we control which interval we capture, and consequently its associated harmonics. Since our goal is to compare the same two pitch classes, our method has a fixed $k_p = 2$, but the sizes of k_o and k_t can be varied, spanning many octaves and timesteps. The effect of dilation exclusively on pitch classes is what achieves the aforementioned non-constant dilation at the frequency dimension. E.g. Let $p = 12$ and a kernel K with $k_o = 3$ and a perfect fifth dilation at the pitch class dimension, in a certain position, this kernel would see $C_1, G_1, C_2, G_2, C_3, G_3$ simultaneously. The distance from each C to the next G is 7 bins, but the distance from each G to the next C is 5 bins. Our method is to the best of our knowledge, the only one capable of achieving that effect in dilation. In the same scenario using linear dilations [8], a kernel with the same size and dilation of a perfect fifth would see instead $C_1, G_1, D_2, A_2, E_3, B_3$. Using our method, a single kernel with $k_o = 3$ and a dilation of perfect fifths at the k_p dimension capture 5 of the first 7 harmonics (see Table 1).

As can be observed in Figure 2, the inputs and outputs of the convolutions have the same size, which is achieved by padding the pitch/octave spectrogram. The values used to pad follow the values of the continuous log-frequency spectrogram. E.g. given $p = 12$, to pad above B_1, we use the values of the bins C_2,$C\sharp_2, etc.$ In contrast, values above the highest octave of the pitch/octave spectrogram will be padded with zeros. We call this method *circular-octave padding*.

[1] When $k_t = 1$, our method can be implemented with 2D convolutions by stacking frames across the batch dimension. 3D is just the general case for an arbitrary k_t

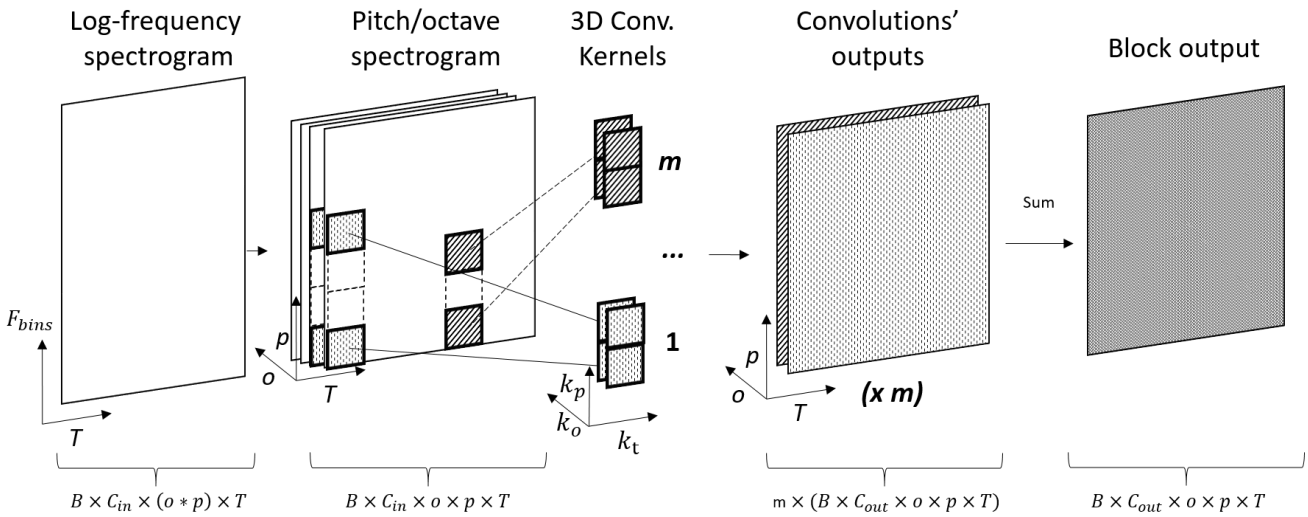


Figure 2: An overview of TriAD. The channel dimension has been omitted in the image. The first stage converts a log-frequency spectrogram onto a pitch/octave one. We apply m of our harmonically motivated kernels to the pitch/octave spectrogram. Each kernel captures different harmonics, depending on the dilation at the p dimension. The kernels' outputs are aggregated by summing the m outputs. B stands for the batch dimension.

4. EXPERIMENTS

We test the performance of our method on AMT for the subtask of piano transcription. Our method is compared with other SOTA approaches of capturing the harmonic spectrum within the architecture itself; concretely, we used the harmonic blocks MRDC-Conv [6], and HD-Conv [8]. We do not include input manipulations such as the HCQT, since these are input manipulations rather than network-internal musically motivated convolutional operations, and a fair comparison is not straightforward.

4.1 Datasets

We used two datasets in our experiments: *MIDI and audio edited for synchronous track and organization* (MAESTRO) [16], and *MIDI aligned piano sounds* (MAPS) [17]. MAESTRO contains about 200 hours of audio for complex piano performances precisely aligned to note labels. Some compositions appear multiple times, each played by a different interpreter. In the paper where MAESTRO is presented, an official train/validation/test configuration was also proposed so that compositions played by different interpreters are in the same split group. We use the latest version of this dataset, version 3, in our experiments. MAPS is another popular dataset used in piano transcription. In contrast to MAESTRO that contains only complete piano pieces, this dataset also contains isolated notes and chords.

Following the practice used in previous works [7, 8, 16], we use the train and validation splits from MAESTRO to train our NNs, and the test sets of MAESTRO and MAPS for testing the trained models. Chunks of audio of 20 seconds and a sample rate of 16.000Hz were used and transformed into a CQT spectrogram, with 352 bins, $f_{min} = 32.070Hz$, and a resolution of 4 bins per semitone. A hop size of 320 samples is employed, resulting in a time resolution of 20 milliseconds.

4.2 The model

We use the HPPNet-base model from [8] for our experiments. This model consists of a backbone and 4 different heads; each head is in charge respectively of predicting which notes are present in each frame, its velocity and whether there is an onset or offset happening. Figure 3 shows an overview of the network. The backbone consists of multiple convolutional layers, and it is divided into three main sections. The first section consists of 3 blocks with 2D convolutions, whose kernels are squarely shaped (7×7) and perform initial processing of the CQT spectrogram. The second section is in charge of doing the backbone's harmonic processing; this is where either HD-Conv, MRDC-Conv, or TriAD will be placed. The last block consists of 5 2D convolutional layers with filter shape (1×5), spanning across the time dimension [2].

The output of the backbone is then used as input for the four heads. Each head consists of a bidirectional long short-term memory (LSTM) [18] and a dense layer. LSTMs model sequential data, which are the features associated with each output bin in this case. The dense layer takes the features outputted by the LSTM and produces a single value for each of the 88 notes of a piano. Details about the design choices of HPPNet can be found in [8].

We run our experiments by comparing the model's performance when the backbone's harmonic processing is done either by our method (TriAD), MRDC-Conv [6], or HD-Conv [8]. We use those methods as employed in their respective papers: 12 kernels of shape (1×1) in the case of [6], and 8 kernels with shape (3×1) in the case of [8]. For our method, we use just two kernels, one dilated for perfect fifths, and another one for major thirds; these are

[2] The third block differs from the original paper description; following their description, that block of the backbone alone has 983.040 parameters, whereas the paper specifies that the backbone contains 421K parameters. We used the network as implemented in the official repo, which matches the number of parameters and replicates their reported results

the intervals with the most associated harmonics. Our kernels span 3 octaves ($k_o = 3$) and a single frame ($k_t = 1$). The code for MRDC-Conv and HD-Conv can be found in their official repositories [3] [4]. We do not train a version of the model with a "harmonically agnostic" block, as [8] already shows in an ablation study that the model's performance drops significantly in that case.

As optimizer, ADAM [19] with a learning rate of $6 \cdot 10^{-3}$ was used. We trained all the models for 200.000 steps, where each step consists of a batch size of 4 chunks of audio. The evaluation was done on MAESTRO's evaluation dataset every 500 steps, to check for possible cases of overfitting. The models were trained 3 times, each one with a random weight initialization. All the harmonic blocks take a similar time to train, around 24h to complete in a V100 GPU.

The employed loss is the same one as in HPPNet's paper [8], a combination of individual losses for the frame, onset, offset, and velocity heads. Weighted binary cross entropy (see Equation 2) was used as loss for the frame, onset and offset heads. This loss is used since there are few positive onset labels, yet predicting onsets is necessary to distinguish consecutive notes. The parameter w controls the relevance of positive labels in the loss and is chosen as $w = 1$ for offsets and frames, and $w = 2$ for onsets. The loss for the velocity head is the mean squared error between the expected and estimated velocities of each individual note.

$$l_{bce}(y, \hat{y}) = -wy \cdot \log(\hat{y}) - (1 - y) \cdot \log(1 - \hat{y}) \quad (2)$$

5. RESULTS

The metrics reported follow the convention described in [20]. These metrics report different aspects of the transcription. The *frame* metric operates at the frame level, while the other three operate at the *note* level. Within the note level, three different metrics exist, considering offsets and/or velocity. This is due to the partially subjective nature of this task. The onset (referred to as the moment when a certain note starts to sound) is not very subjective given the sharp attack of the piano [21]. In contrast, offset (the moment when a certain note stops sounding) and velocity are less objective aspects of the transcription. An estimate of a note is assumed to be correct if its onset is within ± 50ms of the reference, and its pitch is correct. When contemplating offsets, in addition to the previous requisites, the estimation's offset should also be within a certain range; this range is either ± 50ms or 20% of the reference note's duration, whatever is larger.

Velocity estimation is more intricate, as depending on the microphone position a note played with a certain velocity can sound louder or quieter. We use the procedure described in [2], which involves rescaling velocities and using linear regression to account for the aforementioned difference in loudness. All the metrics were calculated using *mir_eval* [22].

[3] https://github.com/WX-Wei/HarmoF0

[4] https://github.com/WX-Wei/HPPNet

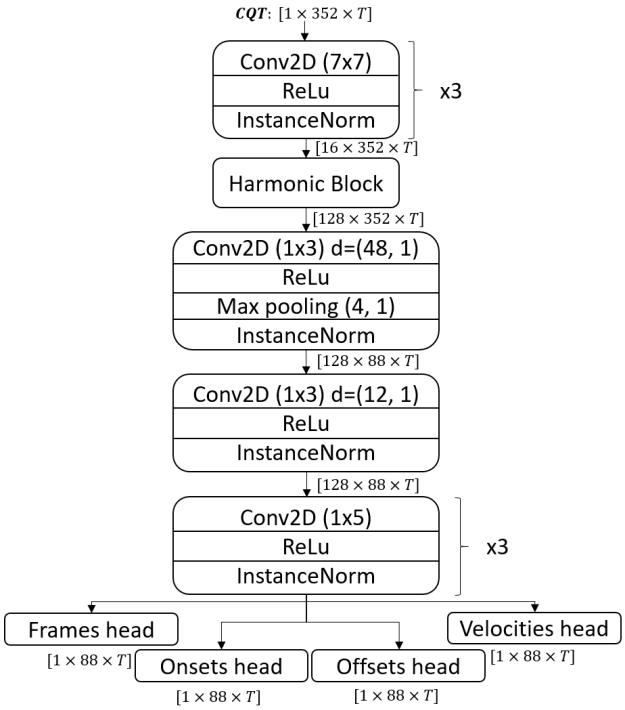


Figure 3: A diagram of HPPNet. The brackets' numbers represent the sizes of the channel, frequency, and frame dimensions. Letter d indicates the dilation rate.

The scores for Section 4 experiments are in Table 2. We also report the results of some larger models of the SOTA as reference. Onsets & Frames [2] is among the most well-known DL models for piano transcription. Semi-CRFs [3] is a method designed to improve the predictions made about the offsets. These are large and capable models, but the ones using harmonic knowledge also manage to achieve similar results with notably fewer parameters. Both TriAD and HD-Conv blocks achieve similar results, in pair with large models. The MRDC-Conv block uses fewer parameters than HD-Conv, but in exchange drops in performance. Noticeably, the model using TriAD has the same number of parameters as the one MRDC-Conv, yet it does not drop in performance.

5.1 Kernel dilation relevance

In music theory, some intervals are more important than others. Equally, some music intervals have more harmonics associated with them than others, as shown in Table 1. It could be expected, that using a kernel dilated with a highly relevant interval yields better results than a kernel associated a with less relevant interval. We tested whether this assumption held or not in our method; instead of using multiple kernels as previously described, our block consists of a single kernel for these experiments. We used 2 relevant intervals (perfect fifth, major third), and 2 lesser relevant intervals (minor second, major seventh) to test the aforementioned assumption. These kernels span 3 octaves ($k_o = 3$) and a single frame ($k_t = 1$), as in the previous ex-

Model	# Parameters	FRAME F1	NOTE F1	NOTE W/OFFSET F1	NOTE W/OFFSET & VEL. F1
		MAESTRO			
Onsets & Frames [2]*	26M	89.68%	95.22%	79.44%	78.85%
Semi-CRFs [3]	9M	90.75%	96.11%	**88.42%**	**87.44%**
HPPNet + HD-Conv	820K	**91.62%**(±.02)	96.14%(±.01)	82.91%(±.02)	80.91%(±.02)
HPPNet + MRDC-Conv	780K	78.69%(±.01)	84.71%(±.01)	58.77%(±.01)	52.15%(±.03)
HPPNet + TriAD (ours)	780K	91.50%(±.02)	**96.16%**(±.01)	82.62%(±.02)	80.76%(±.01)
		MAPS			
HPPNet + HD-Conv	820K	**72.45%**(±.02)	**86.09%**(±.01)	**42.77%**(±.02)	40.11%(±.02)
HPPNet + MRDC-Conv	780K	63.25%(±.01)	73.87%(±.02)	32.68%(±.02)	32.68%(±.01)
HPPNet + TriAD (ours)	780K	72.39%(±.03)	85.06%(±.02)	42.41%(±.02)	**40.17%**(±.02)

Table 2: Results for the experiments described in Section 4. In our experiments, each model was trained three different times. The metrics here reported are the average across these runs and in parenthesis the variance. * Results from [8].

Model	Major third		Perfect fifth		Minor second		Major seventh	
	MAESTRO	MAPS	MAESTRO	MAPS	MAESTRO	MAPS	MAESTRO	MAPS
HPPNet + TriAD	**90.14%**(±.02)	**71.58%**(±.01)	**90.23%**(±.02)	**71.98%**(±.01)	83.16%(±.01)	**68.53%**(±.01)	83.36%(±.01)	**69.19%**(±.02)
HPPNet + HD-Conv	84.89%(±.01)	69.96%(±.02)	85.98%(±.02)	70.50%(±.03)	**84.23%**(±.03)	67.86%(±.03)	**84.79%**(±.01)	68.69%(±.02)

Table 3: F1 framewise results for the single kernel experiments described at section 5.1. Our method obtains worse results if a "less relevant" music interval is chosen. HD-Conv achieves more similar results regardless of the dilation, with just a small improvement for the case of the perfect fifth (where it employs two kernels).

periment. We also used the method with constant dilations i.e. HD-Conv from [8], equally using single kernels except for the case of the perfect fifth. There are two harmonics associated with the perfect fifth within the first 3 octaves, so we employ two rather than a single kernel. The constant dilations capture in this case major third: $5th$ harmonic; perfect fifths, $3rd$ and $6th$ harmonics; minor second, $17th$ harmonic; and major seventh $30th$ harmonic. We noticed that after 50.000 steps, the speed at which the loss diminished slowed down sensibly, and therefore, we reduced the number of training steps for this experiment and trained for 70.000 steps in each run.

The results can be seen in Table 3. HD-Conv [8] obtains slightly better results for the perfect fifth kernels, but similar results for other cases. Our method (TriAD) has a distinguishable performance gap depending on the interval. Results are worse for minor second and major seventh intervals, compared to the cases of the major third and the perfect fifth. Moreover, in those two cases, our method achieves notably better results than HD-Conv.

6. CONCLUSIONS

In this paper, we presented TriAD, a novel convolutional block for NNs capable of capturing the harmonics related to music intervals. To obtain such information, we separate octave and pitch class dimensions from log-frequency spectrograms and create convolutional kernels specifically designed to process this disentangled representation. We tested and compared our method with other ones designed to capture harmonic information, in the task of piano-AMT. We also compared how our model performed when only a single kernel was employed. To the best of our knowledge, our method is the only one capable of achieving dilated convolutions which are not "equally spaced" along the frequency axis, allowing our model to capture multiple harmonics using a small kernel. To achieve this effect, other approaches require applying different convolutional layers to the same input [6, 8] or using large kernels [12].

Our method is still capable of reaching the performance achieved by other harmonic blocks while making use of fewer parameters, showing the effectiveness of our approach. Furthermore, the results from the experiment described in Subsection 5.1 show that our method's performance highly depends on the dilation choice, thus hinting that our method is indeed using the harmonics to determine which pitches are present. Moreover, with an appropriate dilation choice our model outperforms other methods also using a single kernel.

Harmonic series are relevant for other tasks beyond AMT, for example, instrument recognition. Some works have found that the harmonics and their respective amplitudes are crucial to correctly classifying instruments [23, 24]. Our method could be employed to capture the amplitude of different harmonics and learn specific patterns for each instrument. In future work, we will use "harmonically designed" networks in other AMT related tasks. Recent advances in AMT such MT3 [25] demonstrate that with the current DL techniques is possible to transcribe an arbitrary number of instruments from a piece of music audio instead of just piano as shown here. Since the harmonics are relevant for instrument recognition, we hypothesize using harmonic blocks such as the ones presented here, the accuracy with which notes are assigned to each instrument in systems like MT3 could improve. We release code for reproducibility experimentation [5] .

[5] https://github.com/migperfer/TriAD-ISMIR2023

7. REFERENCES

[1] M. Mauch and S. Dixon, "Approximate note transcription for the improvedidentification of difficult chords," in *Proceedings of the 11th International Society for Music Information Retrieval Conference*, 2010.

[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," in *Proceedings of the 19th International Society for Music Information Retrieval Conference*, 2018.

[3] 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*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34, 2021.

[4] R. M. Bittner, B. McFee, J. Salamon, P. Li, and J. P. Bello, "Deep Salience representations for F0 estimation in polyphonic music," in *Proceedings of the 18th International Society for Music Information Retrieval Conference*, 2017.

[5] Jiří Balhar and Jan Hajič jr., "Melody extraction using a harmonic convolutional neural network," MIREX Melody Extraction Report, Tech. Rep., 2019.

[6] W. Wei, P. Li, Y. Yu, and W. Li, "HarmoF0: Logarithmic Scale Dilated Convolution for Pitch Estimation," in *2022 IEEE International Conference on Multimedia and Expo (ICME)*, Jul. 2022.

[7] X. Wang, L. Liu, and Q. Shi, "Enhancing Piano Transcription by Dilated Convolution," in *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*, Dec. 2020.

[8] W. Wei, P. Li, Y. Yu, and W. Li, "HPPNet: Modeling the Harmonic Structure and Pitch Invariance in Piano Transcription," in *Proceedings of the 23th International Society for Music Information Retrieval Conference*, 2022.

[9] 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)*, May 2022.

[10] V. Lostanlen and C. Carmine-Emanuele, "Deep convolutional networks on the pitch spiral for musical instrument recognition," in *Proceedings of the 18th International Society for Music Information Retrieval Conference*, 2017.

[11] J.-F. Ducher and P. Esling, "Folded cqt rcnn for real-time recognition of instrument playing techniques," in *Proceedings of the 20th International Society for Music Information Retrieval Conference*, 2019.

[12] X. Wang, L. Liu, and Q. Shi, "Harmonic Structure-Based Neural Network Model for Music Pitch Detection," in *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*, Dec. 2020.

[13] J. C. Brown, "Calculation of a constant Q spectral transform," *The Journal of the Acoustical Society of America*, vol. 89, no. 1, Jan. 1991.

[14] B. Graham, M. Engelcke, and L. v. d. Maaten, "3d semantic segmentation with submanifold sparse convolutional networks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[15] A. Elowsson, "Polyphonic pitch tracking with deep layered learning," *The Journal of the Acoustical Society of America*, vol. 148, no. 1, 2020. [Online]. Available: https://doi.org/10.1121/10.0001468

[16] C. Hawthorne, A. Stasyuk, A. Roberts, I. Simon, Cheng-Zhi, A. Huang, S. Dieleman, E. Erich, J. Engel, and D. Eck, "Enabling factorized piano music modeling and generation with the maestro dataset," in *Proceedings of the 7th International Conference on Learning Representations*, 2019.

[17] 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, 2010.

[18] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, 1997.

[19] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, 2015.

[20] J. Salamon, "Melody extraction from polyphonic music signals," Ph.D. dissertation, Universitat Pompeu Fabra, Barcelona, Spain, 2013.

[21] A. Ycart, L. Liu, E. Benetos, and M. T. Pearce, "Investigating the perceptual validity of evaluation metrics for automatic piano music transcription," *Transactions of the International Society for Music Information Retrieval*, vol. 3, no. 1, 2020.

[22] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, and D. P.W., "mir_eval: A transparent implementation of common mir metrics," in *Proceedings of the 15th International Society for Music Information Retrieval Conference*, 2014.

[23] Y. Mo, "Music timbre extracted from audio signal features," *Mobile Information Systems*, vol. 2022, Jun 2022. [Online]. Available: https://doi.org/10.1155/2022/1349935

[24] A. Livshin and X. Rodet, “The significance of the non-harmonic "noise" versus the harmonic series for musical instrument recognition,” in *Proceedings of the 7th International Society for Music Information Retrieval Conference*, 2006.

[25] 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.

DATA COLLECTION IN MUSIC GENERATION TRAINING SETS: A CRITICAL ANALYSIS

Fabio Morreale
University of Auckland
f.morreale@auckland.ac.nz

Megha Sharma
University of Tokyo
meghas@g.ecc.u-tokyo.ac.jp

I-Chieh Wei
University of Auckland
iwei022@aucklanduni.ac.nz

ABSTRACT

The practices of data collection in training sets for Automatic Music Generation (AMG) tasks are opaque and overlooked. In this paper, we aimed to identify these practices and surface the values they embed. We systematically identified all datasets used to train AMG models presented at the last ten editions of ISMIR. For each dataset, we checked how it was populated and the extent to which musicians wittingly contributed to its creation. Almost half of the datasets (42.6%) were indiscriminately populated by accumulating music data available online without seeking any sort of permission. We discuss the ideologies that underlie this practice and propose a number of suggestions AMG dataset creators might follow. Overall, this paper contributes to the emerging self-critical corpus of work of the ISMIR community, reflecting on the ethical considerations and the social responsibility of our work.

1. INTRODUCTION

The quest to generate music with AI (Automatic Music Generation, AMG) is undergoing crucial but overlooked ontological, artistic, and political transformations. Originally confined to academic labs and employed in niche music genres, this quest is gaining traction mostly among commercial companies [1] aiming at automatically generating music in all genres. These transformations are enabled by a combination of socio-technical novelties, including i) the growing influx of money in the field [2, 3]; ii) advanced in Deep-Learning (DL) techniques, such as Transformers [4]; and iii) the increase of cheap computational power. While, from a purely musical perspective, the quality of the music created with AI is undoubtedly rising, from a socio-political perspective, a new gold rush resulting from efforts to outperform competitors and make the best AMG model is following the typical blueprint of capitalist innovation [5–7]: corners are being cut; critical questions have not been asked; short-term gains are prioritised; *permission* is not being sought.

These ethically questionable practices are causing increased concerns. A group of artists [2] recently released a manifesto that identifies one of the most urgent ethical issues arising from AI-generated art: the exploitation of artists' work in training AI generation systems. Similarly, Holly Herndon, a musician famous for popularising AI-generated music, recently criticised OpenAI for not asking living performers' permission to use their music in their AI model, JukeBox [8]. Most systems that generate artistic content using Machine Learning (ML) indeed often indiscriminately populate their training datasets by accumulating original material that is available online [9–12].

Within the ISMIR community, occasional fiery calls requested the community to reflect on the ethical implications [13–15] of, and demanded accountability [2] for the work we produce. However, no specific work investigated the potentially exploitative nature of the datasets we use, and no ethical consideration has been given to how data has been generated. We argue that such investigation is long overdue, especially as the publication of AMG models proceeds undisturbed - and actually, as we will show in the paper, is steadily increasing.

To fill this gap, we aimed to assess the extent to which training sets used in ISMIR papers that propose new AMG models are affected by this issue. We first identified all papers presented at the last ten editions of the conference, from 2013 to 2022, that introduced a new music generation model or a pertinent dataset. Then we identified all dataset(s) that have been used in these papers. Finally, we surveyed information for each dataset, including how data was populated and the extent to which musicians wittingly contributed to its creation.

The contribution of this paper is threefold. First, we provide descriptive statistics about the datasets that are mostly used at ISMIR in AMG applications and how they are populated. Second, we report the ideologies that are embedded in them and outline a lack of adequate engagement with musicians and carelessness on ethical matters. Third, we offer suggestions for dataset creators interested in following responsible practices in their work.

The rest of the paper is structured as follows. We first review literature in Critical Data set Studies and report discussions on ethical issues within MIR. We then describe

[1] A list from Water & Music [1] includes, as of July 2023, companies like Microsoft, Facebook, Google, Spotify, Deezer, and ByteDance.

[2] The European Guild for Artificial Intelligence Regulation, https://www.egair.eu/

© F. Morreale, M. Sharma, and I. Wei. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).
Attribution: F. Morreale, M. Sharma, and I. Wei, "Data Collection in Music Generation Training Sets: A Critical Analysis", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

the research process, report the results, identify the values that are inscribed in the datasets, and offer suggestions for dataset creators. We conclude the paper with a summary of the study and directions for future work.

2. BACKGROUND

Deep-learning (DL), which nowadays is the most commonly adopted method to generate music automatically [16–19], significantly relies on the quality and volume of vast training data. Despite this reliance, dataset development remains an underappreciated element in DL practice.

2.1 Critical Data Set Studies

A growing literature on *critical data set studies* [20] aims at identifying the ethical issues and hegemonic power structures of datasets, in particular when used to train ML models [10,12,21]. One of the most urgent issues concerns the exploitation of user labour in AI systems: datasets are populated with data generated by "unwitting labourers" [9] and scraped from the Internet "without context and without consent" [11]. The question around consent is particularly convoluted: consent may have been given unwittingly, for a specific use only, and "some people may never have been given the chance to offer their consent at all" [20]. This concern is not limited to the ivory tower of academia. Musicians are gaining awareness of this issue, and an increasing number of complaints arise from the unfair or unconsented use of original material in AI-generated art. [3]

Most critical work on dataset creation addressed Computer Vision (CV) sets [11, 24, 25] like ImageNet and MS-Celeb, which contain tens of millions of digital images uploaded by platform users. [25] identified the values embedded into these datasets and their formation: evaluating *model work* is prioritised to the detriment of careful *data work*. Another case study that received attention is that of reCAPTCHA [26–28]. Disguised as a *human authentication tool*, reCAPTCHA can be seen as a capture-machine that exploits unpaid individuals' perceptual abilities and micro-labour to train AI datasets [27, 29].

The very way in which most datasets are created embeds specific neoliberal values, like extractivism and deregulation, as exemplified by OpenAI's argument that "IP should be free to use for AI, with training constituting fair use" [30, p. 54]. The all-you-can-scrap ideology dismisses individuals' contributions to dataset creation, which can be met by their creators with a *laissez-faire* attitude [31] that overlooks the ethical implication and liability of scraping the whole Internet [21,25]. In fact, when concerns are voiced, they are specifically aligned with libertarian values and related to how data privacy and data ownership are barriers to collecting data [25].

The practices and routines of data accumulation are not secret. The opposite is true. Among dataset creators, they have become widely accepted, unquestioned, and unchallenged following a process of *dataset naturalisation*: "the contingencies of dataset creation are eroded in a manner that ultimately renders the constitutive elements of their formation invisible" [24]. Notably, the values and ideologies are not only inscribed in how technology is used but also in how it is taught. The lack of interest in how datasets are constructed can indeed be found in the lack of guidance in typical ML textbooks or syllabi [24, 32].

While many dataset creators do not consciously attempt to hide their data accumulation practices, they do not try to fully disclose them either. Dataset naturalisation is indeed exacerbated by ill documentary practices: as reported by [33], ML communities pay little attention to documenting data creation and use. [24] proposes that the lack of information on dataset creation (e.g. how datasets have been created, and whether and how much annotators have been paid) is *structural* - thus ideological - rather than accidental. Every decision and every step in dataset development that is left unaccounted and unarticulated from documentary practices has a political meaning as these steps and decisions are related as "not important" [25]. We will return to this point in the discussions.

2.2 Critical turn in MIR and AIM

Several technology communities are undergoing a *critical turn* [34–37] that challenges existing knowledge production methods and political positions as well as ethical and political thoughts within a field. This turn is ethico-onto-epistemological [38, 39] insofar as it questions what kinds of work, knowledge, and social commitment is pursued within and by the community.

While most criticisms of MIR research come from outside the community [3, 40–42], recent academic production within MIR [2, 3, 14, 43–45] and the development of a workshop series on Human-Centric MIR [46] testify that we might be close to a *Critical MIR* - i.e. MIR scholarship devoted to critically analysing the work produced in the field. However, the sort of work that is (not) published at ISMIR (less than 0.5% of the ISMIR submissions engage with any sort of ethical discussions [2]) indicates that the response of the field on ethical issues is still inadequate.

With respect to AMG, the ethical issues that have been identified include copyright issues [15, 47], a narrow and Western-centered understanding to music [43, 45], the risk of *musician redundancy* [2, 14] or the *crisis of proliferation* [44, 48], diversity issues [49], colonialist and extractive practices [2], and assumptions and bias that are embedded in the AI systems [13–15]. To the best of our knowledge, the potentially exploitative nature of AMG datasets remains uncharted territory.

3. METHODOLOGY

3.1 Researcher Positionality and Motivation

Positionality statements are common in critical studies and serve as a foundation for critical work to understand the research context and the authors' interpretation of the results. Since the outset of the research process, we have

[3] Notable cases include GettyImages suing Stable Diffusion's creators [22] and audiobook narrators complaining against Apple for using their voices to train AI [23].

strived to maintain objectivity and reflexivity by acknowledging our unique positions and backgrounds. All three authors are actively involved in MIR. The first author is formally trained in computer science and is expert in critical theory and technology studies; the second author has a background in computer science; and the third author has a background in electronic engineering and is specialised in machine learning algorithms.

The motivation for undertaking this study is twofold. First, we aimed to support the growth of ISMIR community by contributing to the corpus of self-reflective work, identifying ideologies that might be latent but, once surfaced, can be considered problematic by community members. Second, the development of the suggestions for dataset creation derived from the personal experience of one of the authors, who was involved in dataset creation for AMGs and acknowledged the importance of community guidance on the best ethical practices to adhere to.

3.2 Analysis of ISMIR publications

We conducted a systematic review of the last ten editions of ISMIR (2013-2022). A total of 1078 publications were sourced from the conference proceedings. Two of the authors manually filtered the papers adopting two inclusion criteria. First, we included all papers presenting a new music generation model. We included all models that generate new compositions or performances, including in-painting, style transfer, and improvisation. Second, we included all papers that introduced a new dataset that could potentially be utilised as training material for AMG models but rejected works that did not contain symbolic or raw audio music files. For example, we did not include the NSynth Dataset [50], which contains sampled notes from different instruments, but we included MedleyDB [51], which contains annotated multitrack audio.

The analysis proceeded in two phases. First, for each paper, we identified whether authors employed existing datasets (i.e. datasets released or introduced before the publication of the ISMIR paper) or created new ones (i.e. datasets created or introduced as part of the original research reported in the paper). We also examined the presence of any discussions of ethics and permission for using data entries training data for AMG models.

In the second phase, we examined the datasets identified in the first phase. For papers that used an existing dataset, we retrieved dataset information from the original paper (whether or not it was published at ISMIR) in which it was introduced. When we could not find sufficient information in the paper, we checked dataset release links, which were found either in the original paper or by a web search of the dataset name. The information we collected included i) data format (symbolic or audio), ii) how datasets were populated; iii) whether data contained original performances, compositions, or arrangements; iv) the data type; v) the extent to which musicians were involved in the dataset creation and whether they were aware of the intended purposes for the dataset; and vi) whether ethical concerns were discussed.

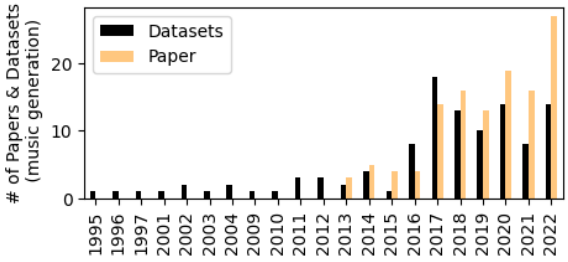


Figure 1: Distribution of selected papers and datasets over the years (only papers after 2003 were considered).

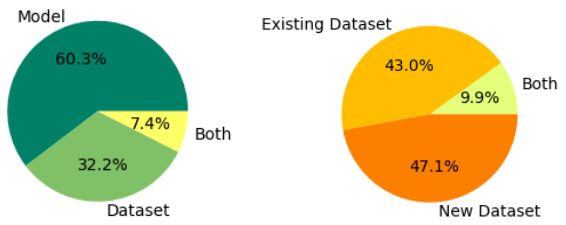
Dataset Name	Format	Occurrence
POP909	Symbolic	11
Nottingham	Symbolic	9
Lakh MIDI	Symbolic	5
HTPD3	Symbolic	3
Yamaha e-Competition	Symbolic	3
Lakh Pianoroll	Symbolic	3
MusicNet	Both	3
URMP	Both	3
AILABS17k	Both	3
Bach Music21	Symbolic	3
Bach Chorales	Symbolic	3
RWC	Both	3

Table 1: The most popular datasets and their occurrences. Each dataset comprises data in symbolic form, but three of them also include audio files.

From a methodological point of view, most of these investigations involved checking the aspect under scrutiny (e.g. whether ethics was discussed) from the dataset sources. The task of identifying how datasets were populated was not as straightforward. In order to streamline the analysis and facilitate the report of the findings, we aimed to cluster datasets into categories that reflected different ways of populating datasets. Two of the authors performed this categorisation following a deductive approach. As they analysed more datasets, they introduced new categories and deleted or merged existing ones. The analysis spreadsheet is available at `https://github.com/Sma1033/amgdatasetethics`.

4. FINDINGS

A total of 121 papers survived the filtering. Fig. 1 shows the significant rise of interest in AMG in recent years. Three fourth of the articles (82) introduced a new model, which was either introduced on its own or with a new dataset (Fig. 2a). From this list of papers, we identified 115 datasets (Fig. 2b). When only considering the 82 papers that introduced a new model, most (62 papers, 75.6%) ISMIR researchers use, at least in part, existing datasets to train their AMG models. Tab. 1 shows the 12 most frequently used datasets in our survey, along with data format and their occurrence in our survey. The remaining 104 datasets were only used in one or two papers.



(a) (b)

Figure 2: **(a) What is introduced:** new model (73 papers), new dataset (39), both (9). **(b) Dataset Originality:** new datasets (58), existing datasets (53), both (12).

4.1 Dataset Creation

We clustered datasets into nine categories to reflect the different ways in which entries were collected (Tab. 2). The categories are non-orthogonal: datasets could be associated with more than one category. In most cases, datasets were populated without creators incurring any costs. Only five datasets used paid online sources or compensated the involved musicians. With the exception of those belonging to the 'Involved musicians' and 'Synthesised music' categories, all datasets were populated with existing music data. New music data accounted for 16.5% of all datasets. We found evidence of poor documentary practices for 18 datasets (15.7%): in 10 cases, there was no information on how data was collected; in 8 cases, we could not find any documentation reporting how datasets were created.

4.2 Musicians' involvement

Only 17 datasets (14.8%) involved musicians in any capacity (category 'Involved musicians'). In 11 cases, musicians performed or arranged existing compositions. Three of these datasets (ASF-4, HP-10, and AIST) were entirely created from novel compositions. The remaining three datasets from this category did not contain compositions or performances created specifically for the dataset, or at least, it was not explicitly mentioned. The Irish Traditional Dance Music dataset [61] used the recordings of one of the authors' own performances. For the remaining two datasets [51, 62], the creators mentioned that professional musicians had created those recordings. However, it is unclear whether these recordings are specifically created for the dataset. The other two datasets that included new music belonged to the 'Synthesised music' category and algorithmically generated monophonic melodies [59] or polyphonic MIDI sequences [63].

4.3 Musicians' permission and awareness

We checked whether explicit permission was sought from musicians to use their creations to train an AMG model. Only three datasets creators reported having asked such permission. The authors of ASF-4 and HP-10 datasets [64] explicitly mentioned that the musicians involved were made aware of the purpose of the dataset for AMG. Jazz players participating in the creation of the FILOSAX dataset [54] signed a document that provided explanations about the goals of the dataset. However, it is not clear whether these goals included AMG: whereas the authors mention "music generation" in the Abstract, AMG was not included in the list of potential applications of the dataset. In the Mozart Piano Music Dataset [65], pianists gave permission to use their performances for the intended use of the dataset (music analysis), but they were probably not aware and did not consent to have their performances used to train the AMG model introduced in [66].

Two cases were particularly problematic. The MAST Dataset [67], which was introduced for automatic rhythm assessment, was sourced from student entrance exams without seeking consent from the students. Another popular dataset, the Yamaha e-Competition dataset [4] features MIDI files of piano performances obtained from the entries of the piano competition. Although Yamaha claims ownership over all data generated during the event, competitors are unlikely aware their performances are used to train AMG models, as seen in [68]. The lack of permission sought from the musicians clashes with the several comments offered by dataset creators that often acknowledged the valuable contributions made by these musicians, which allows the dataset to existing in the first place.

4.4 Discussions on Ethical Issues

Our analysis revealed a lack of engagement with ethical issues, corroborating findings from [25] in their analysis of CV datasets. Only four datasets included any ethical considerations, and only two of them contained an explicit ethics statement. The authors of [69], which presented a new GuitarPro dataset, listed several questions, some of which are particularly relevant to this paper: "*How to acknowledge, reward and remunerate artists whose music has been used to train models?*" and "*What if an artist does not want to be part of a dataset?*". While their spontaneous engagement with these issues is commendable, it is not clear to which extent the authors used these questions in the development of their dataset.

In [70], the authors raised concerns about the impact of AMG for "human musicians of the future". They also stated "care have (*sic*) to be given regarding the fair use of existing musical material for model training" but did not further explain what sort of care and what constitutes unfair use. [57] included an analysis concerning plagiarism issues and observed that their model demonstrated a potential tendency for plagiarism. This issue was also recently highlighted in [47], similar to the level exhibited by a human musician.

5. DISCUSSIONS

By leveraging our findings, this section first reports and discusses the values embedded in the datasets used at ISMIR for AMG models. Then, we move to offer practical suggestions to AMG dataset creators.

[4] https://www.piano-e-competition.com/

Category	Description	Occurrence
Scraped online	Existing music data collected online from websites [52] or databases [53]	49
Existing datasets	Existing music data collected from existing datasets [16]	26
Involved musicians	New music data was created by involving musicians in some capacity [54]	17
Private data	Existing music data was collected from private databases [55]	5
Book collection	Existing music data collected from printed books [56]	5
Online store	Existing music data collected from an online commercial website [57]	4
CD collection	Existing music data collected from published CD recordings [58]	3
Synthesised music	New music synthesised using rule-based heuristics or other methods [59]	2
Not mentioned	No explicit information about how data was obtained [60]	18

Table 2: Categorisation of how data was collected in the datasets. For each category, we included exemplary references.

5.1 The Values Embedded in Our Datasets

Our analysis extends what [24, 25, 33] have suggested for other ML applications areas: *data work* and data collection practices are de-prioritised and de-valued in AMG datasets used at ISMIR. Most datasets ($\sim$60%) had been populated either by scraping songs from the Internet or by accumulating data from existing ones. Considering original music as a *terra nullius* that is free for the taking means addressing dataset creation with expediency. This approach follows the hegemonic narrative that compares *data* to *oil*. This comparison is highly ideological [71–73] as it disguises the origin (and ends) of data [11], and de-penalises and justifies extractive practices using neo-colonial rhetoric that data is something waiting to be discovered [71, 73].

This narrative underestimates or blatantly neglects the human labour necessary for its development - which includes writing, performing, transcribing, and recording music. The majority of datasets were created by amassing musical compositions initially intended for purposes other than AMG. In these cases, the original labour that was put into the creative acts of composition or performance is simply neglected. Relatively few datasets included original material, and in only two cases, specific permission was asked to use musicians' work to train datasets for AMG purposes. This discussion point resonates with objections to the unfair and exploitative practices of capturing individuals' labour and *humanness* [9] when creating data for digital platforms [6, 74–76] and training AI systems [10–12]. As proposed by [77], human labour is *structurally obfuscated* in ML applications to the benefit of profit and innovation. Similarly, [78] proposes that hiding the labour in this context is crucial to attracting capital investments.

Our direct knowledge and lived experience of MIR offers us a vantage point that we can employ in our reflexive inquiry. We propose that dataset creators might have prioritised safety over criticality and followed common, albeit questionable, procedures simply because these are the procedures that are typically employed in AMG research. This comment is not intended to absolve dataset creators from the responsibilities that come with their work. Rather, it is an invitation to self-assess one's alignment with the exploitative ideologies we surfaced in this section. Yet, we unequivocally found a lack of *data work* - including a limited interest in creating one's own data, exploitation of the labour of unwitting musicians (e.g. in the e-piano competition) and students [67] in dataset curation, and poor documentary practices regarding the source of data [60, 79]. We argue that this lack is ideological. What we leave unaccounted for or unspoken in dataset creation and documentation signs what we consider important or irrelevant [25].

Our findings indicate that the rights and demands of musicians are not prioritised by dataset creators and that the degree to which new models and datasets advance or curb a fair model for musicians is largely ignored. This comment resonates with a note from [80], who explained that streaming services overlook "the rights of musicians or users because their decisions are made based on wholly other problems". It is thus essential that ISMIR researchers and practitioners reflect on the *problems* they drive their decisions on and the *agendas* they implicitly or explicitly follow. Answering questions like "what is the agenda we are following and who benefits from it?" [81] requires community discussions that are difficult, uncomfortable, and controversial but nevertheless necessary. Avoiding engaging with these questions is not a political absence but rather a political tacit acceptance of the status quo [36, 37] as datasets do not exist in a political void [20, 82].

5.2 Suggestions

In this section, we offer suggestions to the broader community and to individual authors interested in creating new datasets or using existing ones to train AMG models. We developed these suggestions by integrating results from our analysis with findings from other academic contributions, including ethical CV datasets recommendations [25]. These suggestions are not intended to be meticulously followed as a recipe book. Rather, we devised them as probes, navigation tools, or structured conversations whose development should continue in a participatory way with the rest of the community.

Develop one's own dataset. While exploiting musicians' labour in AI dataset creation is a questionable practice [9], expecting dataset creators to seek and obtain consent from all humans involved in AMG datasets is unrealistic [30]. Thus, we recommend creating, whenever possible, one's own dataset and hire musicians for as many tasks as possible (i.e. composing, performing, arranging songs). A small but important amount of datasets in our

investigation followed this practice. We acknowledge that this suggestion might lead to equity issues. If it were to be enforced, only big companies and top university labs would have the economic means to develop such datasets. However, rather than dismissing this issue as unsolvable and continuing business as usual, we propose that the community interrogates itself and finds strategies to tackle it. As an alternative, efforts might be made to develop models that are trainable on small or procedurally-generated datasets following recent successful examples like [30, 83].

Receive consent from musicians and remunerate them. Dataset creators should inform musicians about the specific goals of the dataset. It is possible that musicians would willingly consent to train a dataset for several MIR tasks but not for training AMG. When possible, dataset creators should consider paying musicians for their labour and disclose the amount [24], as found in [54]. Given the equity problem discussed above, when paying musicians is not feasible, that should be reported [25], and musicians should be at least acknowledged. When AMG systems are integrated into commercial products, a technical infrastructure might be implemented to distribute royalties to dataset contributors. This suggestion shares Holly Herndon's vision for a novel IP framework "compensates me for my likeness when (and only when) money is made from it" [8].

Document the process of dataset development. Our analysis revealed a general lack of care not only in *doing* but also in *documenting* data work. For instance, POP909 dataset's creators did not mention the source or selection process of the "909 popular songs" used to generate piano arrangements [84] and the Lakh dataset's creators simply mentioned that they extracted songs from "publicly-available sources on the internet" [5] website. Careless documentary practices, which we believe were mostly involuntary and caused by an undervaluing of this process in the field [24, 25], implicitly reveal that *how a dataset is developed* and *whose labour goes in it* is not important. We suggest the community develop protocols, guidelines, or templates offering *fair practice* suggestions for dataset creators to follow.

Report the intended use of the dataset. Our findings indicate that it is a common practice among AMG dataset creators to reuse existing datasets. We suggest that dataset creators should report the original intended use of their dataset and list the potential 'allowed' applications, following the example of [54]. This practice would prevent, or at least dissuade, future dataset creators from using that data for purposes other than the ones envisioned by the creators and that musicians agreed on. This suggestion is grounded on the observation that technologies are often interpreted, used, and appropriated in ways that their creators cannot foresee or control (what [85] terms *designer's fallacy*). As new applications of datasets are discovered, measures should be taken to ensure that permission from involved musicians is obtained to use their work for uses other than the ones they agreed on.

When borrowing data, maintain the purpose of the original datasets. Connected to the above suggestion, creators should maintain their original purpose when borrowing entries for new datasets and avoid misappropriation. This is particularly important when dealing with culturally relevant and sensitive music. This is, for instance, the case of the dataset on the Australian Aboriginal language used by [86]. The author reported: "These datasets were public domain and encouraged for use by the creator as a way to share the sound of the language. Even so, it is not clear that the creators of the dataset from the late nineties could predict this (AI generation) 'future use' case" [30].

Volunteer ethical considerations. Our analysis revealed that almost the entirety of the papers did not engage in any form of ethical considerations. Authors can show commitment to advancing more just practices in dataset creation by reflecting on potential ethical limitations in their datasets. Preferably, they should also include documents approved by an Ethics board, if applicable, that were given and signed by the participating musicians.

6. CONCLUSIONS AND FUTURE WORK

We identified the dominant approaches to dataset creation within ISMIR and analysed them with critical lenses to understand their ideological substrate. Most authors seem to handle dataset creation with neoliberal attitudes and expediency. However, a small - yet significant - number of dataset creators showed that other attitudes and values are at play within ISMIR when creating datasets for AMG. Our analysis did not explain the motivations for dataset creators to engage, or not engage, with ethical issues in their work, and this investigation is left for future work. Finally, we aim to extend the analysis to papers other than the ones published at ISMIR and to conduct an ethnographic study with AMG dataset creators to give voice to their perspectives on the topic. To conclude, ISMIR has been playing a significant role in the growth of ML models for AMGs but the lack of an ethical infrastructure may facilitate an exploitative industry. It is our responsibility as the main academic hub of AMG to recognise the need to engage in discussions around the matters raised in the article and to establish ISMIR as the home of this debate.

7. ETHICAL STATEMENT

In this study, we only used secondary data (desk research) that is publicly available online. We reflect that analysing existing information does not incur ethical issues.

8. CONFLICTS OF INTEREST

We have no relevant financial or non-financial interests to disclose and no financial or proprietary interests in anything discussed in this paper.

9. DATA AVAILABILITY

The spreadsheet with our analysis is available at `https://github.com/Sma1033/amgdatasetethics`.

[5] `https://colinraffel.com/projects/lmd/`

10. REFERENCES

[1] D. Edwards and D. McGlynn, "Creative AI for artists: Track 80+ tools," https://www.waterandmusic.com/data/creative-ai-for-artists/, [Accessed: 12-Apr-2023].

[2] F. Morreale, "Where does the buck stop? Ethical and political issues with AI in music creation." in *Transactions of the International Society for Music Information Retrieval*, 2021, pp. 105–114.

[3] E. Drott, "Copyright, compensation, and commons in the music AI industry," in *Creative Industries Journal*, 2021, pp. 190–207.

[4] 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*, 2017.

[5] D. Harvey, *A brief history of neoliberalism*. Oxford University Press, USA, 2007.

[6] S. Zuboff, "The age of surveillance capitalism: The fight for a human future at the new frontier of power," *PublicAffairs*, 2018.

[7] E. Morozov, *To save everything, click here: The folly of technological solutionism*. Public Affairs, 2013.

[8] M. Clancy, "The Artist: Interview with Holly Herndon," in *Artificial Intelligence and Music Ecosystem*, 2023, pp. 44–51.

[9] F. Morreale, E. Bahmanteymouri, B. Burmester, A. Chen, and M. Thorp, "The unwitting labourer: extracting humanness in ai training," *AI & SOCIETY*, pp. 1–11, 2023.

[10] P. Tubaro, "Learners in the loop: Hidden human skills in machine intelligence," in *Sociologia Del Lavoro*, 2022, pp. 110–129.

[11] K. Crawford, "The atlas of AI: Power, politics, and the planetary costs of Artificial Intelligence," *Yale University Press*, 2021.

[12] N. Dyer-Witheford, A. M. Kjøsen, and J. Steinhoff, "Inhuman power: Artificial Intelligence and the future of capitalism," *Pluto Press*, 2019.

[13] A. Holzapfel, B. Sturm, and M. Coeckelbergh, "Ethical dimensions of music information retrieval technology," in *Transactions of the International Society for Music Information Retrieval*, 2018, pp. 44–55.

[14] G. Born, J. Morris, F. Diaz, and A. Anderson, "Artificial Intelligence, music recommendation, and the curation of culture," *Schwartz Reisman Institute for Technology and Society White Paper*, 2021.

[15] B. L. Sturm, M. Iglesias, O. Ben-Tal, M. Miron, and E. Gómez, "Artificial Intelligence and Music: Open questions of copyright law and engineering praxis," in *Arts*, 2019, p. 115.

[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 *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

[17] W. Chen, J. Keast, J. Moody, C. Moriarty, F. Villalobos, V. Winter, X. Zhang, X. Lyu, E. Freeman, J. Wang, S. Cai, and K. M. Kinnaird, "Data usage in MIR: History & future recommendations," in *Proceedings of the International Society for Music Information Retrieval Conference*, 2019, pp. 25–30.

[18] 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.

[19] 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.

[20] N. B. Thylstrup, "The ethics and politics of data sets in the age of machine learning: deleting traces and encountering remains," *Media, Culture & Society*, vol. 44, no. 4, pp. 655–671, 2022.

[21] R. Van Noorden, "The ethical questions that haunt facial-recognition research," *Nature*, vol. 587, no. 7834, pp. 354–359, 2020.

[22] J. Vincent, "Getty images is suing the creators of AI art tool Stable Diffusion for scraping its content," https://www.theverge.com/2023/1/17/23558516/ai-art-copyright-stable-diffusion-getty-images-lawsuit, [Accessed: 11-Apr-2023].

[23] S. Agarwalt, "Audiobook narrators fear apple used their voices to train AI," https://www.wired.com/story/apple-spotify-audiobook-narrators-ai-contract/, [Accessed: 11-Apr-2023].

[24] E. Denton, A. Hanna, R. Amironesei, A. Smart, and H. Nicole, "On the genealogy of machine learning datasets: A critical history of imagenet," in *Big Data & Society*, 2021.

[25] M. K. Scheuerman, A. Hanna, and E. Denton, "Do datasets have politics? Disciplinary values in computer vision dataset development," in *Proceedings of the ACM on Human-Computer Interaction*, 2021, pp. 1–37.

[26] V. Avanesi and J. Teurlings, "I'm not a robot, or am i?: Micro-labor and the immanent subsumption of the social in the human computation of recaptchas," in *International Journal of Communication*, 2022, p. 19.

[27] B. T. Pettis, "reCAPTCHA challenges and the production of the ideal web user," in *Convergence*, 2022.

[28] R. Mühlhoff, "Human-aided Artificial Intelligence: Or, how to run large computations in human brains? Toward a media sociology of machine learning," in *New media & Society*, 2020, pp. 1868–1884.

[29] J. O'Malley, "Captcha if you can: How you've been training AI for years without realising it," https://www.techradar.com/news/captcha-if-you-can-how-youve-been-training-ai-for-years-without-realising-it, [Accessed: 12-Apr-2023].

[30] R. Savery and G. Weinberg, "Robotics: Fast and curious: A CNN for ethical deep learning musical generation," in *Artificial Intelligence and Music Ecosystem*, 2022, pp. 52–67.

[31] E. S. Jo and T. Gebru, "Lessons from archives: Strategies for collecting sociocultural data in machine learning," in *Proceedings of the conference on fairness, accountability, and transparency*, 2020, pp. 306–316.

[32] I. Goodfellow, Y. Bengio, and A. Courville, "Deep learning," *MIT press*, 2016.

[33] T. Gebru, J. Morgenstern, B. Vecchione, J. W. Vaughan, H. Wallach, H. D. III, and K. Crawford, "Datasheets for datasets," *Commun. ACM*, vol. 64, no. 12, p. 86–92, nov 2021. [Online]. Available: https://doi.org/10.1145/3458723

[34] N. E. Gold, R. Masu, C. Chevalier, and F. Morreale, "Share your values! Community-driven embedding of ethics in research," in *CHI Conference on Human Factors in Computing Systems Extended Abstracts*, 2022, pp. 1–7.

[35] S. Benford, C. Greenhalgh, B. Anderson, R. Jacobs, M. Golembewski, M. Jirotka, B. C. Stahl, J. Timmermans, G. Giannachi, M. Adams *et al.*, "The ethical implications of HCI's turn to the cultural," in *ACM Transactions on Computer-Human Interaction*, 2015, pp. 1–37.

[36] F. Morreale, A. Bin, A. McPherson, P. Stapleton, and M. Wanderley, "A NIME of the times: Developing an outward-looking political agenda for this community," in *New Interfaces for Musical Expression*, 2020.

[37] O. Keyes, J. Hoy, and M. Drouhard, "Human-computer insurrection: Notes on an anarchist HCI," in *Proceedings of the CHI conference on human factors in computing systems*, 2019, pp. 1–13.

[38] K. Barad, "Meeting the universe halfway: Quantum physics and the entanglement of matter and meaning," *duke university Press*, 2007.

[39] C. Frauenberger, "Entanglement HCI the next wave?" in *ACM Transactions on Computer-Human Interaction*, 2019, pp. 1–27.

[40] J. W. Morris, "Curation by code: Infomediaries and the data mining of taste," in *European journal of cultural studies*, 2015, pp. 446–463.

[41] N. Seaver, "Computing taste: Algorithms and the makers of music recommendation," *University of Chicago Press*, 2022.

[42] J. Sterne and E. Razlogova, "Machine learning in context, or learning from LANDR: Artificial Intelligence and the platformization of music mastering," in *Social Media + Society*, 2019.

[43] G. Born, "Diversifying mir: Knowledge and real-world challenges, and new interdisciplinary futures," in *Transactions of the International Society for Music Information Retrieval*, 2020.

[44] M. Clancy, "Reflections on the financial and ethical implications of music generated by Artificial Intelligence," Ph.D. dissertation, Trinity College Dublin. School of Creative Arts. Discipline of Music, 2021.

[45] R. Huang, B. L. Sturm, and A. Holzapfel, "Decentering the west: East asian philosophies and the ethics of applying Artificial Intelligence to music." in *Proceedings of the International Society for Music Information Retrieval Conference*, 2021, pp. 301–309.

[46] L. Porcaro, C. Castillo, and E. Gómez Gutiérrez, "Music recommendation diversity: a tentative framework and preliminary results," in *Workshop on Designing Human-Centric Music Information Research Systems.*, 2019.

[47] Z. Yin, F. Reuben, S. Stepney, and T. Collins, "Deep learning's shallow gains: a comparative evaluation of algorithms for automatic music generation," in *Machine Learning*, 2023, pp. 1–38.

[48] J. Attali, "Noise: The political economy of music," *Manchester University Press*, 1985.

[49] L. Porcaro, C. Castillo, and E. Gómez Gutiérrez, "Diversity by design in music recommender systems," in *Transactions of the International Society for Music Information Retrieval*, 2021.

[50] 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*, 2017.

[51] 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 *Proceedings of the International Society for Music Information Retrieval Conference*, 2014, pp. 155–160.

[52] 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*, 2017.

[53] M. Pesek, P. Godec, M. Poredos, G. Strle, J. Guna, E. Stojmenova, M. Pogacnik, and M. Marolt, "Introducing a dataset of emotional and color responses to music." in *Proceedings of the International Society for Music Information Retrieval Conference*, 2014, pp. 355–360.

[54] D. Foster, S. Dixon *et al.*, "Filosax: A dataset of annotated jazz saxophone recordings," in *Proceedings of the International Society for Music Information Retrieval Conference*, 2021.

[55] 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.

[56] E. Parada-Cabaleiro, A. Batliner, A. Baird, and B. W. Schuller, "The seils dataset: Symbolically encoded scores in modern-early notation for computational musicology." in *Proceedings of the International Society for Music Information Retrieval Conference*, 2017, p. 575–581.

[57] S. H. Hakimi, N. Bhonker, and R. El-Yaniv, "Bebopnet: Deep neural models for personalized jazz improvisations." in *Proceedings of the International Society for Music Information Retrieval Conference*, 2020, pp. 828–836.

[58] 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*, 2018, pp. 483–490.

[59] A. Pati, S. Gururani, and A. Lerch, "dmelodies: A music dataset for disentanglement learning," in *Proceedings of the International Society for Music Information Retrieval Conference*, 2020.

[60] L. N. Ferreira and J. Whitehead, "Learning to generate music with sentiment," in *Proceedings of the International Society for Music Information Retrieval Conference*, 2019.

[61] P. Beauguitte, B. Duggan, and J. D. Kelleher, "A corpus of annotated irish traditional dance music recordings: Design and benchmark evaluations." in *Proceedings of the International Society for Music Information Retrieval Conference*, 2016, pp. 53–59.

[62] 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 International Society for Music Information Retrieval Conference*, 2017.

[63] A. Ycart, E. Benetos *et al.*, "A study on lstm networks for polyphonic music sequence modelling," in *Proceedings of the International Society for Music Information Retrieval Conference*, 2017.

[64] L. Angioloni, V. Borghuis, L. Brusci, and P. Frasconi, "Conlon: A pseudo-song generator based on a new pianoroll, wasserstein autoencoders, and optimal interpolations." in *Proceedings of the International Society for Music Information Retrieval Conference*, 2020, pp. 876–883.

[65] G. Widmer, "Discovering simple rules in complex data: A meta-learning algorithm and some surprising musical discoveries," in *Artificial Intelligence*, 2003, pp. 129–148.

[66] S. Lattner, M. Grachten, and G. Widmer, "A predictive model for music based on learned interval representations," in *Proceedings of the International Society for Music Information Retrieval Conference*, 2018.

[67] 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 International Society for Music Information Retrieval Conference*, 2019.

[68] H. H. Tan and D. Herremans, "Music fadernets: Controllable music generation based on high-level features via low-level feature modelling," in *Proceedings of the International Society for Music Information Retrieval Conference*, 2020.

[69] P. Sarmento, A. Kumar, C. Carr, Z. Zukowski, M. Barthet, and Y.-H. Yang, "DadaGP: A dataset of tokenized guitarpro songs for sequence models," in *Proceedings of the International Society for Music Information Retrieval Conference*, 2021.

[70] 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.

[71] L. Gitelman, "Raw data is an oxymoron," *MIT press*, 2013.

[72] J. Sadowski, "When data is capital: Datafication, accumulation, and extraction," *Big data & society*, vol. 6, no. 1, p. 2053951718820549, 2019.

[73] L. Stark and A. L. Hoffmann, "Data is the new what? popular metaphors & professional ethics in emerging data culture," *Journal of Cultural Analytics*, 2019.

[74] H. R. Ekbia and B. A. Nardi, "Heteromation, and other stories of computing and capitalism," *MIT Press*, 2017.

[75] B. Brown, "Will work for free: The biopolitics of unwaged digital labour," in *tripleC: Communication, Capitalism & Critique. Open Access Journal for a Global Sustainable Information Society*, 2014, pp. 694–712.

[76] M. Pasquinelli, "Google's pagerank algorithm: A diagram of cognitive capitalism and the rentier of the common intellect," in *Deep search: The politics of search beyond Google*, 2009, pp. 152–162.

[77] J. Sadowski, "Planetary potemkin AI: The humans hidden inside mechanical minds," in *Digital Work in the Planetary Market*, p. 229.

[78] L. Irani, "Difference and dependence among digital workers: The case of amazon mechanical turk," in *South Atlantic Quarterly*, 2015, pp. 225–234.

[79] C. Ó Nuanáin, H. Boyer, S. Jordà Puig *et al.*, "An evaluation framework and case study for rhythmic concatenative synthesis," in *Devaney J, Mandel MI, Turnbull D, Tzanetakis G, editors. ISMIR 2016. Proceedings of the 17th International Society for Music Information Retrieval Conference; 2016 Aug 7-11; New York City (NY).[Canada]: ISMIR; 2016. p. 67-72.* International Society for Music Information Retrieval (ISMIR), 2016.

[80] J. W. Morris, "Selling digital music, formatting culture," *University of California Press*, 2015.

[81] C. Cath, "Governing artificial intelligence: ethical, legal and technical opportunities and challenges," in *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 2018.

[82] J.-P. Deranty and T. Corbin, "Artificial intelligence and work: a critical review of recent research from the social sciences," *AI & SOCIETY*, pp. 1–17, 2022.

[83] T. Moore and J. Brazeau, "Serge modular archive instrument (smai): Bridging skeuomorphic machine learning enabled interfaces," in *New Interfaces for Musical Expression*, 2023.

[84] 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.

[85] D. Ihde, "The Designer Fallacy and Technological Imagination," in *Defining Technological Literacy: Towards an Epistemological Framework*, 2006, pp. 121–131.

[86] R. Savery, R. Rose, and G. Weinberg, "Establishing human-robot trust through music-driven robotic emotion prosody and gesture," in *International Conference on Robot and Human Interactive Communication*, 2019, pp. 1–7.

A REVIEW OF VALIDITY AND ITS RELATIONSHIP TO MUSIC INFORMATION RESEARCH

Bob L. T. Sturm
Division of Speech, Music and Hearing
KTH Stockholm, Sweden
bobs@kth.se

Arthur Flexer
Institute of Computational Perception
Johannes Kepler University Linz, Austria
arthur.flexer@jku.at

ABSTRACT

Validity is the truth of an inference made from evidence and is a central concern in scientific work. Given the maturity of the domain of music information research (MIR), validity in our opinion should be discussed and considered much more than it has been so far. Puzzling MIR phenomena like adversarial attacks, horses, and performance glass ceilings become less mysterious through the lens of validity. In this paper, we review the subject of validity as presented in a key reference of causal inference: Shadish et al., *Experimental and Quasi-experimental Designs for Generalised Causal Inference* [1]. We discuss the four types of validity and threats to each one. We consider them in relationship to MIR experiments grounded with a practical demonstration using a typical MIR experiment.

1. INTRODUCTION

The multi-disciplinary field of Music Information Research (MIR) is focused on making music and information about music accessible to a variety of users. This ranges from systems for search and retrieval, to recommendation, and even to more creative applications like music generation. The effectiveness and reliability of MIR systems are of prime importance to the MIR researcher, not to mention other stakeholders. The researcher thus performs experiments to compare approaches for modeling and retrieving music data. A principal focus is on users, but the cost of performing experiments with users is high, and the replicability of such studies is difficult. This has motivated the *Cranfield Paradigm* [2]: computer-based experiments where "test collections" serve as proxies for human users. While such an approach is inexpensive and replicable, its relevance and reliability for MIR, and information retrieval in general, have been questioned [3, 4].

Under the Cranfield Paradigm, state-of-the-art MIR systems perform exceptionally well in reproducing the ground truth of some datasets, e.g., inferring rhythm, genre or emotion from audio data. This leads to conclusions that the systems are actually learning to perform the task believed necessary to recover the ground truth from audio data. However, slight and irrelevant transformations of the audio, e.g., "adversarial attacks", can suddenly render these systems ineffectual [5–9]. Such attacks can reveal what an MIR system is relying on for its success. In one case, a "genre recognition" system relies on infrasonic signatures that are imperceptible and irrelevant for human listeners [8]. In another, a "rhythm recognition" system is recognising tempo instead of rhythm, a confounding originating from the data collection [6]. Systems relying on such "tricks" have been called "horses" [5]. A related topic in MIR is "glass ceilings" [10, 11], i.e., that an observed barrier to improving system performance to perfect or human level is claimed as coming from psychophysical and cultural factors of music missing from features extracted from audio recordings [12].

In order to better understand the problems described above it is necessary to consider what lies at the heart of any experiment: the relationship between conclusions drawn from its results and their *validity*, or "truth value" [1]. Ideally, an experiment will be carefully designed and implemented to answer a well-defined hypothesis. Its components – units, treatments, design, observations, and settings – should be carefully operationalised (translated from theory into practice) to maximize quality and minimize cost (e.g., money and time). This is the purview of the discipline *Design of Experiments*: how can one get the strongest evidence for the least cost?

Despite a small chorus of calls to improve validity of conclusions in MIR, e.g., [4–6, 13–19], there has yet to be published a systematic and critical engagement of what validity means in the context of MIR, and how to consider it when designing, implementing and analyzing experiments. In this paper, we focus on the four principal types of validity in Shadish et al. [1], an authoritative resource about validity in causal inference and experimental science. Other typologies exist, e.g., [20], but we use that of Shadish et al. [1] because it is an established point of reference, and has already been mentioned in the context of MIR, e.g., in [4]. We review the four types of validity and present actionable questions that can help MIR researchers to scrutinize the conclusions they draw from their experiments. We ground our general discussion of validity in this paper by a practical demonstration, [1] which presents a typical MIR experiment

© Bob L. T. Sturm, Arthur Flexer. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Bob L. T. Sturm, Arthur Flexer, "A Review of Validity and its Relationship to Music Information Research", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

[1] See supplementary material here: https://github.com/boblsturm/mirvaliditytutorial

Model	Accuracy	Precision	Recall	f1-score
LDA	0.714	0.711	0.711	0.703
QDA	0.719	0.715	0.723	0.717
1NN	0.662	0.644	0.635	0.638
3NN	0.681	0.673	0.651	0.656
5NN	0.719	0.699	0.687	0.689
7NN	0.695	0.669	0.656	0.659
9NN	0.700	0.681	0.664	0.668
unif	0.12 ± 0.02	0.13 ± 0.03	0.12 ± 0.02	0.12 ± 0.02
freq	0.13 ± 0.02	0.13 ± 0.03	0.13 ± 0.02	0.13 ± 0.02
maj	0.16	0.02	0.12	0.03

Table 1. Accuracy, and macro-averaged precision, recall and f1-score observed for several models in a testing partition of BALLROOM [21]. The performance of two models selecting labels randomly (with standard deviation) are shown in the rows labeled: *unif* samples labels uniformly; *freq* samples labels according to training data label frequency. The last row *maj* shows the performance of a model choosing the label most frequent in the training data.

that exemplifies a considerable amount of MIR research: music classification using machine learning (ML) and a benchmark dataset. We use the BALLROOM dataset [21], which has appeared in dozens of studies seeking to build MIR systems sensitive to rhythm [6]. We partition the dataset into training and testing sets, extract features and train ML models, then label test set recordings and count coincident ground truth labels, and finally compute figures of merit for the different ML models. Table 1 presents the results from which we wish to draw valid conclusions.

A less abridged version of this paper [22] integrates the supplementary material in more detail. We hope that these materials will help MIR researchers to design, implement and analyze experiments in MIR and draw valid conclusions, but also convince them that the language of validity is reason. Creative thinking is necessary when examining the truth value of any conclusion drawn from an experiment.

2. COMPONENTS OF EXPERIMENTS

Before discussing the validity of conclusions drawn from an experiment, we must identify its components: units, treatments, design, observations, and settings. *Treatments* are the things applied to units in order to cause an effect (or not in the case of a *control*), *units* are the things that are treated, and *observations* are what is measured on a unit. The *design* specifies which treatment is applied to which unit, and *settings* involve time, place, and condition. To make this more concrete, consider a medical experiment in which the effect of a treatment on blood pressure is being studied. A number of people are sampled from a population, some of whom will receive the treatment while the others receive a placebo (control). The design describes which people get the treatment, and which do not. The observation is the blood pressure of a person after one month. The setting can include particulars of the population (rural or urban), place of treatment (hospital or home), and so on. The experimentalist contrasts blood pressure observations across groups to conclude, e.g., the effect of the treatment (causes a decrease in blood pressure).

Our typical MIR experiment measures the effectiveness of different ML models in predicting the labels of a test recording dataset. There are two ways to see its components. We can see the treatments as the ten models and the units as replicates of the entire testing dataset, or we can see the entire testing dataset as the one treatment and the units as the ten models. Since Table 1 reports figures of merit (observations) of each model on the entire test dataset, the latter interpretation motivates conclusions about the effectiveness of particular models. In this case, the design is simple: each unit (ML model) is given the same treatment (dataset). The setting involves the dataset partitioning, the extracted features, random seeds, software libraries, etc.

3. STATISTICAL CONCLUSION VALIDITY

Statistical conclusion validity is "the validity of inferences about covariation between two variables" [1]. This includes concluding that a covariation exists, and perhaps its strength as well. This is the level at which one is concerned with *statistical significance*, i.e., that an observed covariation between treatment and effect is not likely to arise by chance. As a concrete example, an experiment measuring the effects of two different medicines on lowering blood pressure seeks to determine which of the medicines has the greatest effect, if at all. The statistical conclusion validity of a conclusion resulting from this experiment relies on its power, but can be threatened in other ways. Shadish et al. [1] (p. 45) includes a table of nine different threats to statistical conclusion validity. Four threats relevant to computer-based experiments are: violated assumptions about the statistics underlying the observations (and the use of the wrong statistical test, a *type III error* [23]); a sample size too small to reliably detect covariation (lack of power); the purposeful search for significant results by trying multiple analyses and data selections ("p-hacking" [24]); and increased variance in observations due to the heterogeneity of units.

Are my results statistically significant? Null hypothesis statistical testing (NHST) quantifies whether the observed effects of the treatments on the responses arise by mere chance, as well as the direction of effect and its size. This answers the question: are the results statistically significant? Fundamentals about statistical testing in MIR have already been discussed [25], also for Artificial Intelligence in general [26], and for ML [27]. One must take care in selecting a statistical test to use; each one makes strong assumptions that could be violated. NHST is most straightforwardly applicable to completely randomized experimental designs [28], thereby reducing the possibility of structure in units and treatments interfering with the responses (which results in confounding). Most MIR experiments cannot use complete randomisation because the target population from which samples come is unclear (what is a random sample of "sad" music, with the term "sad" being quite ill-defined?), and so the kinds of conclusions that can be made with NHST in MIR are limited. [2]

[2] Experimental designs that cannot be completely randomised are called *quasi-experimental designs*, another major topic of Shadish et al. [1].

Is the observed statistical significance relevant for a user? In MIR, even if one finds statistical significance, this may not generalise to a perceivable difference for actual users interacting with the "improved" MIR system. As an example from MIR, a crowd-sourced user evaluation [29] demonstrates that there is an upper bound of user satisfaction with music recommendation systems of about 80%, since this was the highest percentage of users agreeing that two systems "are equally good." In addition, for the MIREX task of *Audio Music Similarity and Retrieval* it has been demonstrated [29] that statistically significant differences between algorithms can be so small that they make no practical difference for users.

Let us now consider the typical MIR experiment and reason about what conclusions we can draw from it that have statistical conclusion validity. Table 1 clearly shows that each response of model to the dataset is greater than the random approaches *unif*, *freq* and *maj*. How likely is it that any of the responses of models is due to chance, i.e., that any of the models is actually no better than one of the random approaches? Since we have the empirical distributions for *unif* and *freq*, we can estimate the probability of either of them resulting in, e.g., a macro-average recall at least as large as 0.6: $p < e^{-200}$.[3] Hence, a valid statistical conclusion is that we observe a significant covariation between the use of a machine learning model with these particular features and the responses measured on a specific partition of BALLROOM.

One might consider statistical conclusions relating to the type of ML, i.e., Gaussian modeling (LDA and QDA) vs. nearest neighbour modeling (KNN), or LDA vs. QDA. If we conclude from Table 1 that Gaussian modeling performs better than nearest neighbour modeling with these features on 70/30 partitions of BALLROOM, we would be wrong. This is a "type I error", which is concluding there to be a significant difference when in fact there is none. When we perform this experiment 1000 times with random 70/30 partitions we observe that the difference between the best response of a Gaussian model and the best response of a nearest neighbour model is distributed Gaussian, and that the probability of observing zero difference or less is $p > 0.41$ for any of the figures of merit.

The most general statistical conclusion we can make from Table 1 is that the responses we observe from ML models are highly inconsistent with the responses of choosing randomly. Each ML model knows *something* about BALLROOM linking the features computed from a music recording with its ground truth label. Because we do not know the amount of variation in any response due to partitioning, we cannot make any valid statistical conclusion about which type of ML model is the best for this particular dataset. In order to go further, we must run the experiment multiple times to obtain distributions of the contrasts. Even then, however, we cannot say anything about the *cause* of significant differences yet. This is where the notion of internal validity becomes relevant.

[3] See the supplementary material for an explanation.

4. INTERNAL VALIDITY

Internal validity is "the validity of inferences about whether the observed covariation between two variables is causal" [1]. While statistical conclusion validity is concerned only with the strength of covariation between treatment and responses, internal validity is focused on the *cause* of a particular response to the treatment. Shadish et al. [1] (p. 55) includes a table of nine different threats to causal conclusions. Several of these involve *confounding*, which is the confusion of the treatment with other factors arising from poor operationalisation in an experiment. As a concrete example, consider an experiment measuring the effects of two different medicines on lowering blood pressure, but where one medicine is given to young patients and the other is given to elderly patients. This experimental design confounds the two medicines and patient age, and so the effects caused by the two factors cannot be disambiguated. Any conclusion from this experiment about the effects of the medicines lacks internal validity.

Does my data collection introduce confounds? One's methodology for collecting data might unintentionally introduce structure. For instance, it has been discussed that BALLROOM was assembled by downloading excerpts of music CDs sold at a website selling music for ballroom dance competitions [6]. Ballroom dance competitions are regulated by organisations, e.g., World DanceSport Federation (WDSF),[4] to ensure uniformity of events for competitors around the world. These organisations set strict requirements of tempo of each dance such that high skill is required of the dancers. Hence, the labels of BALLROOM can reflect any of the following: 1) the rhythm of the music; 2) the type of dance performed to the music; 3) the strict tempo requirements of the dance in the context of competition. As a result, good performance in BALLROOM can be due to rhythm detection and/or tempo estimation. Tempo and rhythm are related musical characteristics, but they are not the same thing [30].

Does my data partitioning introduce confounds? Dataset partitioning can also introduce confounds, e.g., "bleeding ground truth." An example is to first segment recordings into short (e.g., 40ms) time frames and then partition these frames into training and testing sets, thus spreading highly correlated features across these sets. In the context of audio-based genre classification, the presence of songs from the same artists or albums in both training and test data has been shown to artificially inflate performance [31, 32]. Audio-based genre classification using very direct representations of spectral content has been shown [33] to degrade more when employing artist/album filters than classification based on more abstract kind of features like rhythmic content (fluctuation patterns). This insight that problems of data partitioning can affect MIR systems in quite different ways and hence change performance rankings has been confirmed in another meta-study [34].

Returning to our typical MIR experiment, of interest is *what* it is in our trained ML models causing their response to be inconsistent with random selection. Knowing how

[4] https://www.worlddancesport.org/

Gaussian models used in LDA and QDA are built – mean and covariance parameters are estimated from training data – an internally valid conclusion is that these models work well in BALLROOM because likelihood distributions estimated from the training data also fit the testing data well. Another internally valid conclusion is that the high performances of these ML models in BALLROOM are caused by the features together with the expressivity of the models capturing information related to the labels in BALLROOM.

With reference to the aims of MIR research, we want to conclude something more specific, e.g., our ML models have learned to recognize the rhythms in BALLROOM. This is certainly one explanation consistent with our observations, but is it the only one? The internal validity of this conclusion relies on a key assumption: inferring the labels of BALLROOM can *only* be the result of learning to discriminate between and identify its rhythms. In other words, we must assume that there is no other way to infer labels in BALLROOM than by perceiving rhythm.

Since we know tempo is highly correlated with rhythm in BALLROOM, we thus perform an experiment to test the sensitivity of our trained ML models to tempo: we alter all test recordings by some amount of pitch-preserving time dilation, and then measure the responses of the models to these new treatments. We see that the responses of all ML models decay to being not significantly different from random selection with dilations in the range of ± 15 %. We see this intervention clearly reveals the extent to which the ML models we test rely on the tempi in the test data.

The experimental design of the typical MIR experiment does not account for the structure present in the dataset; we do not control for other ways of inferring the labels of BALLROOM, which are guaranteed to exist by its very construction. From Table 1 and our experimental design, we thus cannot be any more specific in our causal inference than this: the responses of our ML models are caused by their having learned *something* about BALLROOM. This then calls into question how comparing predictions with ground truth in BALLROOM relates to the ability we might actually want to measure, that is the recognition of rhythm. This is where the notion of construct validity becomes relevant.

5. CONSTRUCT VALIDITY

Construct validity is "the validity of inferences about the higher order constructs that represent sampling particulars" [1]. This involves the relationship between what is meant to be inferred by the experimentalist from an experiment and what is actually measured, i.e., the *operationalisation* of the experimentalist's intention. For instance, directly measuring the blood pressure of a person involves an invasive procedure inserting a measuring device in their veins. Blood pressure can be measured less invasively but indirectly by externally applying known pressure to a vein and listening for when blood flow ceases. Knowledge about the incompressibility of liquids in closed systems makes the measurement of pressure in the balloon a relevant measure of blood pressure. Shadish et al. [1] (p. 73) includes a table of fourteen different threats to construct validity, but several of these are irrelevant to computer-based experiments. The main threat is a questionable relationship between what is being measured and what is intended to be measured. Selecting a measure by convenience but not relevance, sampling from convenient populations, and a lack of definition of what is intended to be measured, are threats to construct validity. Construct validity involves more than just how something is measured; it also involves what is measured and in what settings.

How is classification accuracy, or any figure of merit, in a labeled music dataset related to X*?* Two examples in MIR are the use of "genre" classification accuracy as an indirect measure of music similarity [11], or user satisfaction (see, e.g., [14] for a discussion). The relationship between these is very tenuous, especially so considering that accuracy itself is an unreliable measure of whether or not a system has learned anything relevant to music [5, 15]. A key reference in this respect is that of Pfungst [35] describing a series of experiments in trying to reliably measure the arithmetic acumen of a horse that was only able to tap out answers. Counting the number of correct answers tapped out by the horse, no matter how many questions are asked, is irrelevant without considering how each question is posed (the setting). The key to Pfungst discovering the cause of the horse's apparent arithmetic acumen involved changing the setting: the questions remained the same, and accuracy of correct response was measured, but how the questions were posed was changed in order to control for different factors of the experiment. The same is true for MIR.

What is the "use case" of the system to be tested? To counter threats to construct validity the MIR experimentalist must operationalise as much as possible the use case of the system to be built and tested. One attempt to do so for music description [36] emphasises the need to define success criteria. The experimentalist must determine how their method of measurement relates to the success criteria, e.g., relating accuracy in genre classification to the satisfaction of a specific type of user.

How can we test the construct validity of a conclusion? One possibility is to assess the outcomes of different experiments which are supposed to measure the same higher order constructs. An example in MIR is to study correlations of different genre classifiers when given identical inputs [18]. Low correlations between classifiers point to problems of construct validity. A related topic is that of adversarial examples, which casts doubt on the conclusion that the high accuracy of an MIR system in some dataset reflects its "perception" of the music in the waveform. Adversarial examples have first been described in image analysis [37], where imperceptible perturbations of input data significantly degraded classification accuracy. For music genre classification systems, imperceptible audio filtering transformations of music signals have been used [5] to both deflate and inflate classification accuracy to be no better than chance level or perfect 100%. Following these so-called untargeted attacks which try to change a prediction to an arbitrary target, targeted attacks aiming at changing predictions to specific classes have been explored. A targeted

attack on genre recognition has been reported [7], where magnitude spectral frames computed from audio are treated as images and attacked using approaches from image object recognition. For music instrument classification a targeted attack allowing to add perturbations directly to audio waveforms instead of spectrograms has also been presented [9]. The attacks were able to reduce the accuracy close to a random baseline and produce misclassifications to any desired instrument. The authors also artificially boosted playcounts via an attack on a real-world music recommender, thereby demonstrating that such attacks can be a security issue in MIR. Follow-up work presented lines of defence against such malicious attacks [38].

Returning to our typical MIR experiment, we are interested in making construct inferences around the latent ability of rhythm recognition we are supposedly measuring in our ML models. For instance, one construct inference is that our features measure relevant aspects of rhythm in recorded music. In some sense, by their definition from basic signal processing components, our features come from temporal aspects that are certainly relevant to rhythm. Our features are also reliant on acoustic information, and in particular there being high-contrast differences in onsets captured by spectral flux – hence limiting their relationship to rhythms played by particular kinds of instruments with sharp attacks. However, we have seen above that the features are also indicative of tempo, and that tempo is another path an ML model can use to infer the rhythm label. Hence we are left to question the relationship of our features to the concept we are trying to operationalise, i.e., rhythm.

Having a system label any partition of the BALLROOM dataset provides no reliable measure of a system's ability to recognise rhythm without changing the setting to control for other factors. It is not as simple as choosing a different feature, measure, cross-validation method, or using a particular statistical test. One must change the experiment itself such that *rhythm recognition* is what is actually being measured. This means that BALLROOM can still be useful to measuring the rhythm recognition of an ML model. Indeed, in the previous section we used it to disprove the causal claim that the good performance of the ML systems of Table 1 is caused by their ability to recognize rhythm. Might performance in BALLROOM also be an indication of performance in other datasets focused on rhythm? This is where the notion of external validity becomes relevant.

6. 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" [1]. More generally, external validity is the truth of a generalised causal inference drawn from an experiment. An example is inferring that medicine found to lower blood pressure in patients living in Germany will also lower blood pressure in people living in Mexico – a conclusion that can lack validity due to differences in diet, living and working conditions, and so on. Another example is that increasing the dose of the medicine will cause blood pressure to lower further in the studied population. If a causal inference we draw from an experiment lacks internal validity, then generalising that inference to include variations not tested will not have external validity. Shadish et al. [1] (p. 87) includes a table of five different threats to external validity, which are in addition to the threats to internal validity. The main threat is that variation of the components of the experiment might destroy the causal inference that holds in the experiment. For instance, a medication may work for the type of illness tested, but that type of illness may not be generalisable to other closely related illnesses.

Does my model generalize to out-of-sample data? The standard approach in evaluating MIR classification systems is to use separate train and test sets in cross-validation experiments to obtain seemingly unbiased estimates of performance. However, if such MIR systems are exposed to independent out-of-sample data often severe loss of performance is observed. One example are experiments on genre recognition where accuracy results do not hold when evaluated across different collections that are not part of the training sets [39, 40]. The results do not generalize to supposedly identical genre labels in different collections, which reflects a lack of external validity. Genre labels like 'Rock' will be used differently by different annotators working on these collections – which is also a threat to construct validity. Another example are how different audio encodings affect subsequent computation of descriptors and classification results [41], or how in general differences in software implementations diminish replicability [42].

Do different raters agree on a ground truth? Human perception of music is highly subjective resulting in possible low inter-rater agreement. Therefore only a certain amount of agreement can be expected if several human subjects are asked to rate the same song pairs according to their perceived similarity, depending on a number of subjective factors [14, 43] like personal taste, listening history, familiarity with the music, current mood, etc. Concerning annotation of music, it has been shown [44] that the performance of humans classifying songs into 19 genres ranges from modest 26% to 71%. Audio-based grounding of everyday musical terms shows the same problematic results [45]. It has even been argued [12] that no such thing as an immovable 'ground' exists in the context of music, because music itself is subjective, highly context-dependent and dynamic.

The 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 is also a problem of reliability, since different groups of users or annotators with their differing subjective opinions will impede repeatability of experimental results. This lack of inter-rater agreement presents an upper bound for MIR approaches, since it is not meaningful to have computational models going beyond the level of human agreement. Such upper bounds have been reported [14,43,46] for the MIREX tasks of 'Audio Music Similarity and Retrieval' (AMS) and 'Music Structural Segmentation' (MSS). For AMS the upper

	Accuracy	Precision	Recall	f1-score
LDA	0.659	0.647	0.643	0.643
QDA	0.682	0.678	0.672	0.673
1NN	0.622	0.616	0.602	0.604
3NN	0.636	0.629	0.610	0.613
5NN	0.644	0.643	0.617	0.619
7NN	0.647	0.646	0.619	0.621
9NN	0.645	0.643	0.615	0.618
unif	0.12 ± 0.01	0.13 ± 0.01	0.12 ± 0.01	0.12 ± 0.01
freq	0.13 ± 0.01	0.13 ± 0.01	0.12 ± 0.01	0.12 ± 0.01
maj	0.13	0.02	0.12	0.03

Table 2. As in Table 1, models trained in BALLROOM and tested in all of X-BALLROOM [51].

bound has already been reached in 2009, while for MSS the upper bound is within reach for at least some genres of music. Comparable results exist concerning music structure analysis [47] and chord estimation [48, 49].

Do raters agree with themselves at different points in time? Going beyond the question of whether different annotators agree on a ground truth one can also access what the level of agreement within one person is when faced with identical annotation tasks at different points in time. A high intra-rater agreement would help to overcome the problem of upper bounds in MIR systems since it would make personalization of models meaningful, i.e. to have separate models for individual persons. However, at least for the task of general music similarity it has been shown that intra-rater agreement is only slightly higher than inter-rater agreement [19], with the absolute level also depending on music material and mood of raters at test time. An approach to personalize chord labels for individual annotators via deep learning was more successful [50].

Returning to the typical MIR experiment, we cannot validly conclude that any of our models is recognizing rhythm in general because we do not know if they are recognizing rhythm in BALLROOM. Our dilation intervention experiment in Sec. 4 reveals that all of the models lose their supposed ability to recognize rhythm in BALLROOM, so there is no reason to infer they will recognize rhythm elsewhere. One causal conclusion we might make is that our models perform well in BALLROOM because they have learned something about BALLROOM – a curated set of recordings downloaded from a specific website in 2004. Might they have learned something about other recordings from that same website, but collected many years later?

The extended BALLROOM dataset (X-BALLROOM) [51] consists of 3,484 audio recordings in the same eight dance styles or music rhythms as BALLROOM, but downloaded from the same website over a decade later. This gives us a chance to test our conclusion. The figures of merit measured from our models trained in BALLROOM but applied to all of X-BALLROOM are shown in Table 2. We still see significant covariation between response and the use of ML with our features. By and large, whatever concepts our ML models have learned about BALLROOM carry over to X-BALLROOM – but we still do not know whether or not those concepts have to do with rhythm.

7. CONCLUSION

This paper provides a review of the notion of validity based on the typology given in Shadish et al. [1]. It brings together the few sources in MIR that mention validity, and several sources that do not but are related. This paper does not aim to prescribe how to design and perform experiments such that valid conclusions can be drawn from them. Instead, it aims to bring within the realm of MIR what validity means, why it is important, and how it can be threatened. One thing to reiterate is that one does not talk about the "validity of an experiment". An experiment does not possess "truth value". Validity is a property of a conclusion made given evidence collected from an experiment. The components of an experiment – units, treatments, design, observations, and setting – have major consequences for the validity of conclusions drawn from it, whether it is statistical conclusion validity, internal validity, construct validity, or external validity.

In MIR the predominant experimental methodology is the Cranfield Paradigm: train a model on a partition of a dataset and count the number of correct answers on another partition. This kind of experiment is inexpensive, and provides numbers that can be compared in ways that convince peer reviewers that progress has been accomplished [52]. Despite various appeals [14, 53] and beseechings [4, 5, 15, 16, 19, 29, 43], such an experimental approach is still standard in the field and its serious flaws are ignored. Any conclusion from this experiment that is more general than "the system has learned something about the dataset" lacks internal, construct and external validity. This does not mean that all such inferences are false, just that they cannot follow from the experiment as designed and implemented. Reproducing the ground truth of a dataset represents a beginning and must be followed by a search for the causes of the observed behavior.

Shadish et al. [1] provides an established starting point for MIR, but there exist other types of validity. For instance, Lund [20] revises the typology of [1] to address ambiguities between causes and treatments, to better define aspects of settings, and to establish a hierarchical ordering of five types of validity: statistical conclusion, causal, construct, generalization and theoretical. Other kinds of validity include ecological, convergent, and criterion [13], but these still deal with the kind of conclusion one is drawing from evidence collected in some way.

As a final note, a frustration when encountering Shadish et al. [1] as an engineer is that of its 623 pages there are only five pages with at least one equation on them. Instead, Shadish et al. [1] describe experiments and how each type of validity manifests in the conclusions drawn, with specific threats to the reasoning of those conclusions. Experiments, not to mention experimentalists, are such complex assemblages that expressing them in formal ways that appear to permit the computation of numbers that relate to each type of validity would probably have very limited applicability, and then only be understood by a limited audience. The language of validity is *reason*, and we hope this article will inspire MIR researchers to think creatively about the phenomena they observe to discover their causes.

8. ACKNOWLEDGMENTS

We thank J. Urbano, H. Maruri-Aguilar, and various anonymous reviewers of previous versions of this paper. The contribution of Sturm is supported by a project that has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant agreement No. 864189 MUSAiC: Music at the Frontiers of Artificial Creativity and Criticism). The contribution of Flexer is supported by funding from the Austrian Science Fund (FWF, project numbers P 31988 and P 36653). For the purpose of open access, the authors have applied a CC BY public copyright license to any author accepted manuscript version arising from this submission.

9. REFERENCES

[1] W. R. Shadish, T. D. Cook, and D. T. Campbell, *Experimental and quasi-experimental designs for generalized causal inference*. Boston: Houghton Mifflin, 2002.

[2] C. W. Cleverdon, "The significance of the Cranfield tests on index languages," in *Proc. Int. ACM SIGIR Conf. Research and Development in Info. Retrieval*, 1991, pp. 3–12.

[3] E. M. Voorhees, "The philosophy of information retrieval evaluation," in *Proc. Cross-Language Evaluation Forum*, 2001.

[4] 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.

[5] 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.

[6] ——, "The "horse" inside: seeking causes behind the behaviors of music content analysis systems," *Computers in Entertainment (CIE)*, vol. 14, no. 2, pp. 1–32, 2017.

[7] 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.

[8] F. Rodríguez-Algarra, B. L. Sturm, and H. Maruri-Aguilar, "Analysing scattering-based music content analysis systems: Where's the music?" in *Proc. Int. Symp. Music Information Retrieval*, 2016, pp. 344–350.

[9] K. Prinz, A. Flexer, and G. Widmer, "On end-to-end white-box adversarial attacks in music information retrieval," *Transactions of the International Society for Music Information Retrieval*, vol. 4, no. 1, pp. 93–104, 2021.

[10] J.-J. Aucouturier and F. Pachet, "Improving timbre similarity: How high is the sky?" *J. Neg. Results Speech Audio Sci.*, vol. 1, no. 1, pp. 1–13, 2004.

[11] T. Pohle, E. Pampalk, and G. Widmer, "Evaluation of frequently used audio features for classification of music into perceptual categories," in *Proc. Int. Workshop Content-Based Multimedia Indexing*, 2008.

[12] G. A. Wiggins, "Semantic gap?? Schemantic schmap!! Methodological considerations in the scientific study of music," in *Proc. Int. Symp. Multimedia*. IEEE, 2009, pp. 477–482.

[13] J. Urbano, "Information retrieval meta-evaluation: Challenges and opportunities in the music domain," in *Proc. Int. Symp. Music Information Retrieval*, 2011, pp. 609–614.

[14] 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.

[15] 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.

[16] ——, "Revisiting priorities: Improving MIR evaluation practices," in *Proc. ISMIR*, 2016.

[17] J. Urbano and A. Flexer, "Statistical analysis of results in music information retrieval: why and how," in *Tutorial at Int. Symp. Music Information Retrieval*, 2018. [Online]. Available: http://ismir2018.ircam.fr/pages/events-tutorial-17.html

[18] C. C. Liem and C. Mostert, "Can't trust the feeling? How open data reveals unexpected behavior of high-level music descriptors," in *Proc. Int. Symp. Music Information Retrieval*, 2020, pp. 240–247.

[19] 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, no. 1, pp. 182–194, 2021.

[20] T. Lund, "A revision of the Campbellian validity system," *Scandinavian J. Educational Research*, vol. 65, no. 3, pp. 523–535, 2021.

[21] S. Dixon, F. Gouyon, and G. Widmer, "Towards characterisation of music via rhythmic patterns," in *Proc. Int. Symp. Music Information Retrieval*, 2004, pp. 509–517.

[22] B. L. T. Sturm and A. Flexer, "Validity in music information research experiments," *arxiv*, vol. arXiv:2301.01578, 2023.

[23] A. W. Kimball, "Errors of the third kind in statistical consulting," *J. American Statistical Assoc.*, vol. 52, no. 278, pp. 133–142, June 1957.

[24] M. L. Head, L. Holman, R. Lanfear, A. T. Kahn, and M. D. Jennions, "The extent and consequences of p-hacking in science," *PLOS Biology*, vol. 13, no. 3, pp. 1–15, 2015.

[25] A. Flexer, "Statistical evaluation of music information retrieval experiments," *Journal of New Music Research*, vol. 35, no. 2, pp. 113–120, 2006.

[26] P. R. Cohen, *Empirical methods for artificial intelligence*. MIT press Cambridge, MA, 1995, vol. 139.

[27] N. Japkowicz and M. Shah, *Evaluating Learning Algorithms: A Classification Perspective*. New York, NY, USA: Cambridge University Press, 2011.

[28] R. A. Bailey, *Design of comparative experiments*. Cambridge University Press, 2008.

[29] J. Urbano, J. S. Downie, B. Mcfee, and M. Schedl, "How significant is statistically significant? the case of audio music similarity and retrieval." in *Proc. Int. Symp. Music Information Retrieval*, 2012, pp. 181–186.

[30] W. A. Sethares, *Rhythm and Transforms*. Springer, 2007.

[31] E. Pampalk, A. Flexer, G. Widmer *et al.*, "Improvements of audio-based music similarity and genre classificaton." in *Proc. Int. Symp. Music Information Retrieval*, 2005, pp. 634–637.

[32] 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, pp. 20–28, 2010.

[33] A. Flexer, "A closer look on artist filters for musical genre classification," in *Proc. Int. Symp. Music Information Retrieval*, 2007, pp. 341–344.

[34] 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.

[35] O. Pfungst, *Clever Hans (The horse of Mr. Von Osten): A contribution to experimental animal and human psychology*. New York: Henry Holt, 1911.

[36] B. L. Sturm, R. Bardeli, T. Langlois, and V. Emiya, "Formalizing the problem of music description," in *Proc. Int. Symp. Music Information Retrieval*, 2014, pp. 89–94.

[37] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *Proc. Int. Conf. Learning Representations*, 2014.

[38] K. Hoedt, A. Flexer, and G. Widmer, "Defending a Music Recommender Against Hubness-Based Adversarial Attacks," in *Proceedings of the 19th Sound and Music Computing Conference*, 2022, pp. 385–390.

[39] D. Bogdanov, A. Porter, H. Boyer, X. Serra *et al.*, "Cross-collection evaluation for music classification tasks," in *Proc. Int. Symp. Music Information Retrieval*, 2016, pp. 379 – 385.

[40] 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. Int. Symp. Music Information Retrieval*, 2019.

[41] J. Urbano, D. Bogdanov, H. Boyer, E. Gómez Gutiérrez, X. Serra *et al.*, "What is the effect of audio quality on the robustness of MFCCs and chroma features?" in *Proc. Int. Symp. Music Information Retrieval*, 2014, pp. 573–578.

[42] B. McFee, J. W. Kim, M. Cartwright, J. Salamon, R. M. Bittner, and J. P. Bello, "Open-source practices for music signal processing research: Recommendations for transparent, sustainable, and reproducible audio research," *IEEE Signal Processing Magazine*, vol. 36, no. 1, pp. 128–137, 2018.

[43] 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.

[44] 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. Int. Workshop Adaptive Multimedia Retrieval*, 2010, pp. 118–131.

[45] J.-J. Aucouturier, "Sounds like teen spirit: Computational insights into the grounding of everyday musical terms," *Language, evolution and the brain*, pp. 35–64, 2009.

[46] M. C. Jones, J. S. Downie, and A. F. Ehmann, "Human similarity judgments: Implications for the design of formal evaluations." in *Proc. Int. Symp. Music Information Retrieval*, 2007, pp. 539–542.

[47] O. Nieto, M. M. Farbood, T. Jehan, and J. P. Bello, "Perceptual analysis of the f-measure for evaluating section boundaries in music," in *Proc. Int. Symp. Music Information Retrieval*, 2014, pp. 265–270.

[48] Y. Ni, M. McVicar, R. Santos-Rodriguez, and T. De Bie, "Understanding effects of subjectivity in measuring chord estimation accuracy," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 12, pp. 2607–2615, 2013.

[49] H. V. Koops, W. B. De Haas, J. A. Burgoyne, J. Bransen, A. Kent-Muller, and A. Volk, "Annotator subjectivity in harmony annotations of popular music," *Journal of New Music Research*, vol. 48, no. 3, pp. 232–252, 2019.

[50] H. V. Koops, W. B. de Haas, J. Bransen, and A. Volk, "Automatic chord label personalization through deep learning of shared harmonic interval profiles," *Neural Computing and Applications*, vol. 32, no. 4, pp. 929–939, 2020.

[51] U. Marchand and G. Peeters, "Scale and shift invariant time/frequency representation using auditory statistics: Application to rhythm description," in *Proc. IEEE Int. Workshop Machine Learning for Signal Processing*, 2016.

[52] D. J. Hand, "Classifier technology and the illusion of progress," *Statistical Science*, vol. 21, no. 1, pp. 1–15, 2006.

[53] 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. Int. Symp. Music Information Retrieval*, 2012.

SEGMENTATION AND ANALYSIS OF TANIĀVARTANAM IN CARNATIC MUSIC CONCERTS

Gowriprasad R[1] **Srikrishnan S**[2] **R Aravind**[1] **Hema A Murthy**[1]
[1] Indian Institute of Technology Madras, [2] Carnatic Percussionist, [1,2]Chennai, India
ee19d702@smail.iitm.ac.in, aravind@ee.iitm.ac.in, hema@cse.iitm.ac.in
[2]srikrishnansridharan@gmail.com

ABSTRACT

In Carnatic music concerts, taniāvartanam is a solo percussion segment that showcases intricate and elaborate extempore rhythmic evolution through a series of homogeneous sections with shared rhythmic characteristics. While taniāvartanam segments have been segmented from concerts earlier, no effort has been made to analyze these percussion segments. This paper attempts to further segment the taniāvartanam portion into musically meaningful segments. A taniāvartanam segment consists of an abhiprāya, where artists show their prowess at extempore enunciation of percussion stroke segments, followed by an optional korapu, where each artist challenges the other, and concluding with mohra and korvai, each with its own nuances. This work helps obtain a comprehensive musical description of the taniāvartanam in Carnatic concerts. However, analysis is complicated owing to a plethora of tāla and naḍe. The segmentation of a taniāvartanam section can be used for further analysis, such as stroke sequence recognition, and help find relations between different learning schools. The study uses 12 hours of taniāvartanam segments consisting of four tāla-s and five naḍe-s for analysis and achieves 0.85 F1-score in the segmentation task.

1. INTRODUCTION

Carnatic music (CM) is a South Indian music tradition considered an ancient form of Indian art music (IAM). A typical CM concert features a lead artist, typically a vocalist, accompanied by a violinist and percussion instrument artists. The lead percussion instrument in this ensemble is usually the *mridangam*, while additional percussion instruments like the *ghatam*, *khanjira*, and *morsing* may also be present. A CM concert includes a solo percussion performance known as *taniāvartanam*, or *tani* for short. Tani is a structured sequence of rhythmic elaborations performed at a fixed metric tempo and bound to a metric cycle (*tāla*). This study attempts to study the elaborations in tani, segment them using a culture-specific approach, and assigns semantically meaningful labels.

© Gowriprasad R, Srikrishnan Sridharan, R Aravind and Hema A Murthy. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Gowriprasad R, Srikrishnan Sridharan, R Aravind and Hema A Murthy, "Segmentation and Analysis of Taniāvartanam in Carnatic Music Concerts", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

Audio recordings of concert performances available online often lack detailed metadata and annotations regarding section boundaries and other information, particularly in the context of IAM. With the increasing availability of music collections and digital devices, there is growing interest in accessing music based on its characteristics. The paucity of editorial metadata has necessitated the development of music information retrieval (MIR) techniques to extract music's characteristic properties from audio recordings automatically. The paper is organized as follows. The taniāvartanam structure is described, followed by the task objectives, challenges, and dataset description. Domain-specific feature engineering is done, and the task is addressed for different cases. The experimental results are analyzed and discussed with culture-specific explanations.

1.1 Taniāvartanam Description

The tani is a highly structured and elaborate percussion performance that is a prominent feature of CM, showcasing the rhythmic skills and creativity of the percussionist. The main percussion instrument is the mridangam, occasionally accompanied by ghatam (clay pot), khanjira, and morsing (Jew's Harp). Since tani is part of a main item, it is performed in the same tāla, and metrical tempo as the main item. The intricacies are based on the precise mathematical calculations of the metric cycle.

The duration of the tani is divided among the mridangam and accompanying percussion to showcase individual artistry, e.g., if mridangam and ghatam are present, the structural framework of the tani is typically as follows: The mridangam always starts first by playing *sarvalaghu* (SV) patterns (indicators of basic tāla structure), and the complex patterns are introduced gradually. These elaborations are performed in a particular rhythm structure called *naḍe* (usually in *chaturaśra* at first) for a few rhythm cycles. These elaborations on a particular rhythmic theme are termed as *abhiprāya*. The literal meaning is "opinion", i.e, the artists' viewpoint of that particular rhythm structure. Ghatam follows and tries to keep the same theme built by the mridangam in the first cycle by playing in the same naḍe [1]. In the second cycle, the mridangist usually may change the naḍe (to *tiśra*, for example) and elaborates. The ghatam usually follows in the same naḍe or switches to a different naḍe (khanḍa). These may or may not continue for more than two cycles, usually owing to time constraints. Each abhiprāya ends with a pattern called *korvai*, which is repeated thrice to arrive at downbeat.

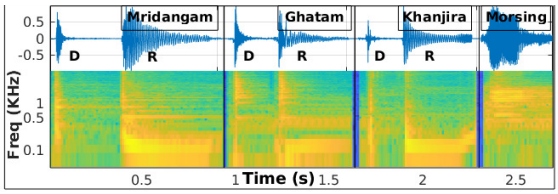


Figure 1. Spectral Illustration of a few Carnatic Percussion Strokes. D: Damped Strokes, R: Resonant Strokes

These abhiprāya-s are followed by the *korapu*, usually seen as a question-answer between mridangam and ghatam. Here it starts with multiple cycles of rhythmic patterns by the mridangam followed by ghatam, where each artist challenges the other. The duration of the rhythm patterns in korapu keeps reducing progressively from full cycle, half cycle, quarter cycle until it finally reduces to a single beat. It can be translated as "rhythmic descent" or "step-by-step reduction". The artist(s) then start playing together playing faster with crisp strokes (*farans*), building up the necessary momentum for playing the last parts of the tani called *mohra* and longer korvai [1]. Each of these has a specific composition structure upon which the artist builds. This structure holds even if only the mridangam is present, except that the korapu part might be absent. Summarizing the sequence of sections in a *tani* segment can be listed as sarvalaghu patterns abhiprāya in a specific naḍe → change of naḍe → back to starting naḍe → korapu → farans → mohra → final korvai [2].

1.1.1 Aspects of timbre and spectral differences among the Carnatic percussion

In Indian music tradition, accompanying instruments are relatively tuned according to the main melodic instrument or voice. The percussion instruments are also categorized on the sonic aspect. Figure 1 illustrates the damped (D) strokes and resonant (R) strokes of Carnatic percussion instruments. Two-sided percussion, mridangam has both the low-frequency and mid-frequency spectra covered. The ghatam occupies a little over the mid-frequency band, and morsing predominantly spreads over the high-mid frequency spectrum and has a larger resonance. Khanjira occupies a low-frequency spectrum a bit less than the left side of the mridangam. This explains the aesthetic quality of the percussion instruments that have been traditionally in use for CM concerts. The tonal nature also enhances the entire concert when played harmoniously.

1.2 Problem Objective and Challenges

This work addresses three primary tasks: (1) Diarization of the audio into mridangam, khanjira, and ghaṭam sections when multiple instruments are present, (2) Estimation of section boundaries using musical attributes, and (3) Classification of segments into broad categories such as abhiprāya, korapu, farans, mohra, and korvai. To achieve these goals, the paper applies techniques from well-researched music genres while also considering the culture-specific characteristics of tani. To improve readability and clarity, several terms are defined in Table 1.

Identifying and understanding the segments in tani is difficult for most CM audiences, except for professionally trained and practicing percussionists. However, this challenge can be addressed if we have a reliable system that can classify the primary segments in tani from audio recordings. Such a system would not only aid in appreciating the art form for a broader audience, but also serve as a valuable learning tool for beginner-level percussion students.

Segment	Audio fragment between any two adjacent detected boundaries that may or may not cover a complete section.
Section	A primary portion of the taniāvartanam. A section can contain multiple compositions and multiple segments.
Naḍe	A modifier to tāla that decides the number of strokes per beat, The subdivision structure within a beat in CM Chaturaśra, Tiśra, Khanḍa are different kinds of naḍe-s
Abhipraya (AB)	A rhythmic elaboration in a particular naḍe during tani.
Korapu (KP)	A musical dialogue between the musicians during performance.
Farans (FA)	The first part of the conclusion in tani where the percussionists play fast to gain momentum toward the end.
Mohra (MO)	Popular rhythmic structure played after the farans hinting the climax of taniāvartanam.
Korvai (KO)	Stroke patterns that are played three times, concluding the tani.

Table 1. Definitions of terms relevant to this paper

Coming to the challenges, tani is very diverse and extempore. The number of percussions may vary across the concerts. The duration of the tani also varies, influencing the number of possible segments. Additionally, the presence of the korapu section is contingent on the number of percussions, which is rare when only mridangam is played. Each rhythmic structure is presented at multiple speeds. This is reflected in the boundary within a single abhiprāya due to sudden tempo changes. The rendition also has small pauses, which may be part of the rhythmic elaboration or due to the artist's presentation style. As a result, the tani segmentation task presents unique challenges to existing audio segmentation methods. Listening to the entire audio carefully to mark the segment boundaries is time-consuming. This underscores the need to develop systems for automatic segmentation and annotations.

1.3 Dataset Description

Experimenting with various shades of tani requires a diverse collection of annotated audio data. As there is no properly annotated dataset available for this task, we collected diverse recordings of tani and labeled them. All the audio data used in this work is a subset of the Charsur Carnatic [3,4], Sangeethapriya [5] datasets along with two audios from [6]. The tani part from the main concerts is extracted by marking the start and end points. Professional performers listened and annotated the boundaries of primary sections in the tani. By doing so, we collected around 12 hours of annotated tani audio. The duration of each tani in the dataset ranges from 6 minutes to 29 minutes, with 11 minutes of mean duration.

The dataset details are described in Table 2. The considered audios comprises of tani played in four major tāla-s of CM [7, 8], namely ādi, miśra chāpu, khanḍa chāpu, and rupaka. The annotations consist of tāla labels, boundary instances, and labels of primary sections of tani. The multiple percussion audios considered in this work have only two instruments along with additional labelings of the instrument name for their respective segments. The dataset is heterogeneous with artist variability (22 mridangam, >12 ghatam, >8 khanjira), tonic, and tempo variability.

	No. of Abhiprāya	No. of Concerts	Duration ~ (hrs:mins)
Mridangam	51	16	02:24
Mrid + Ghat	86	18	04:56
Mrid + Khanj	94	21	05:47
Total	231	55	12:08

Table 2. Dataset Description.

1.4 Related Work

Segmentation and metadata labeling of a music recording have a fairly good research history both in Western [9–11] and IAM traditions [12, 13]. Various acoustic and temporal parameters were used for the segmentation task [9, 14]. Foote et al. [15] proposed a self-distance matrix method to determine the boundary between contrasting musical characteristics. The changes in musical features in Pop and Rock music were used to train the boosted decision stump [16]. Lately, [17] explored neural networks for structural segmentation, spanning various genres [18].

In the context of IAM, different approaches were explored for segmenting the main concert audios in the Dhrupad [13, 19, 20], Hindustani [21, 22], and Carnatic [12, 23–25] music traditions. For instrumental concerts, Vinutha et al. [22] considered the segmentation of sitar and sarod concerts using reliable tempo detection [26]. The analysis of rhythm/percussion in IAM has primarily focused on stroke onset detection [27, 28], stroke recognition [6, 29–33], and sequence modeling [34, 35] percussion pattern identification [36]. Ajay Srinivasamurthy [37] worked on tracking the "downbeat," provided the tāla is known. Tani diarization was also attempted in [4]. Further, mridangam artist identification from tani audio was attempted [38]. Parallel to [38], tabla gharānā recognition from the tabla solo was addressed in [39, 40].

Nevertheless, no attempts have been reported on the structural analysis of Indian solo percussion. This paper attempts to include additional meta-information to the tani portion of a concert, where the audio is segmented based on musical attributes. This can help identify the tāla and enable the association of the cycle of strokes with that of the lyrics of the main composition in CM. The outcomes can help in the concert summarization task and for further MIR studies in the field of percussion, which is crucial as it can give insights into the rhythm of the main item of the concert. Combined with works on meter tracking [7], percussion source separation [41], and stroke recognition [6], this could lead to additional metadata that could be important to an ardent listener or performer.

2. AUDIO FEATURE ENGINEERING

The raw concert audios have to be pre-processed for further analysis. Since each concert is unique in the choice of metric tempo, tonic, and compositional structure, the features used should be based on concert-specific characteristics. At the same time, it should scale inter-concert. We address the tasks by computing relevant features considering the culture-specific musicological perspectives. Initially, the raw audio is pre-processed by computing the Hilbert envelope of the linear prediction (LP) residual on the raw audio [27]. Then the onset detection function (ODF) is computed

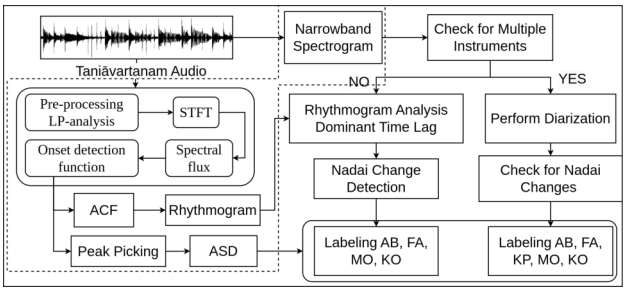


Figure 2. Flow Diagram for Segmentation and Labeling

using the spectral flux method [42]. It is shown to perform on par with state-of-the-art machine learning-based onset detection algorithms on percussion instruments [27]. The computed onset locations are considered for further rhythm analysis. While we have used LP analysis, any onset detection technique could have been used.

2.1 Rhythm and Tempo Features

The change in the rhythm structure or the tempo is a prominent indicator of the section transitions. In the case of percussion instruments, rhythm pattern refers to the aspects of stroke patterns. A rhythm representation can be obtained by considering the stroke ODF (sampled at 10 ms) over a suitably long window and computing the auto-correlation function (ACF). The periodicity analysis using the ACF of the ODF represents the audio in terms of rhythm called rhythmogram [43–45], where rhythm/tempo alone is emphasized.

The ACF of the ODF is computed frame-wise with a frame length of 4 seconds and a frameshift of 0.5 seconds up to a lag of 1 second. The dimension of each frame of the rhythmogram is $p = 100$, corresponding to a 1-second lag. The window length must be large enough to contain sufficient strokes for computing the ACF. Even while playing a slower tempo, we observe at least more than 8-10 strokes (sufficient to calculate the periodicity) in a window length of 4-5 seconds. A uniform window size of 4s is chosen to accommodate variability in rhythm. The peaks along the lag axis of the rhythmogram depict the periodicity of the surface rhythm, indicating surface tempo [22].

The tempo estimation using the product of ACF-DFT [46] is often prone to tempo octave errors due to uneven stroke distribution. We compute the number of strokes in each 4 seconds frame and divide by 4 to get the stroke density at every frame instance. The feature is named average stroke density (ASD), as the averaging is done over 4 seconds frame. The ASD is robust to tempo octave errors and is representative of surface tempo [13]. The mean and std. deviation of strokes per second, as obtained in the entire dataset, are 8.6 and 3.8, respectively. The variance of ASD depicts the tempo diversity in the dataset. Figure 4(c) shows the evolution of ASD over time.

2.2 Spectral Feature

From Section 1.1.1, it is clear that each of the Carnatic percussion instruments has distinct spectral properties, and the spectral features can serve as potential features for instru-

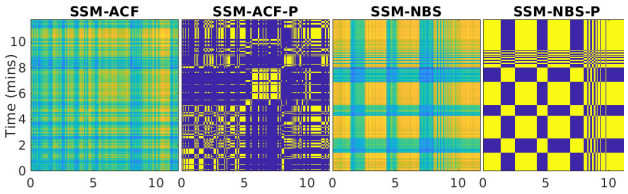


Figure 3. Self-Similarity Matrices on Different Features

ment classification. In this work, we need to localize the segments as coming from one of the percussion. To get the complete spectral aspects of a particular instrument, the spectrum must be computed over a window with almost all kinds of strokes. Thus we computed a narrow band spectrogram (NBS) with a window size of $4s$ and a hop size of $0.5s$. From Section 2.1, we know that the mean ASD is eight strokes per second. Thus in a four-second frame, we can expect at least one resonant stroke. We can clearly distinguish mridangam and ghatam segments from NBS in Figure 4(a).

2.3 Spectral and Rhythm Posteriors

The high-dimensional NBS and ACF rhythmogram represent the spectral and rhythmic-tempo homogeneity within the segment and the changes between the adjacent segments. This allows us to use Gaussian mixture models (GMM) to model the instrument's spectral and temporal homogeneity. The NBS and ACF vectors are converted to spectral and rhythm posteriors (NBS-P, ACF-P), representing class conditional probabilities.

We use two mixtures GMM to represent NBS feature vectors with the intuition that each instrument property is modeled by one mixture. Interestingly we find that each mixture corresponds to a different timbre. We also tried a third mixture to represent the portion where both the instruments play together (FA, MO, KO). This failed due to the volume dominance of mridangam and gave false alarms. The posterior feature computed on NBS is depicted in Figure 4(d). The posteriors from the rhythmogram are computed with five mixture components, each representing a particular speed. The GMM is fit only on the NBS and ACF vectors from a particular concert. The number of Gaussians is determined by the different speeds and naḍe-s expected in a concert.

3. TANI SEGMENTATION AND LABELING

Since tani may contain only mridangam or multiple instruments, we first need to detect if a particular tani audio has multiple instrument or not. The abhiprāya region segmentation task is slightly different in both cases. Locating the abhiprāya boundaries is based on detecting a change in the instrument itself (in case of multiple instrument) and the local rhythmic structure of segments at the highest timescale (in case of solo mridangam). Figure 2 shows the overall steps involved in the task. Each of the segmentation and labeling steps is described here.

3.1 Multiple Instrument Detection

From Sections 1.1.1, we know that different Carnatic percussion instruments differ in their sonic and timbral aspects

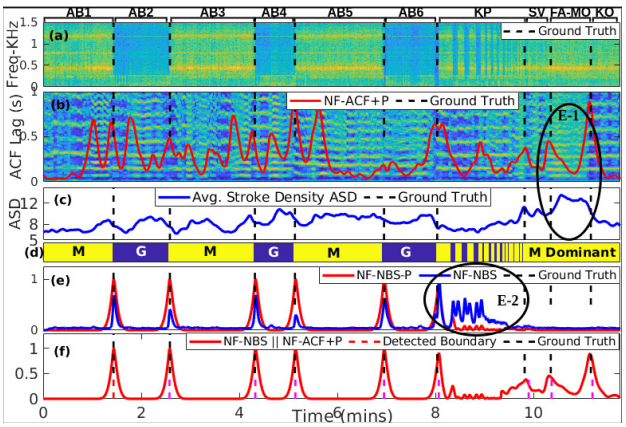


Figure 4. Eg: Multiple Instrument Tani: Segment labels on top (a) NBS feature (b) ACF Rhythmogram with NF-ACF+P overlay-ed (c) ASD evolution over time (d) Posteriors computed on NBS (e) NF-NBS-P (red) obtained from $(15s \times 15s)$ kernel, NF-NBS (blue) from $(3s \times 3s)$ kernel (f) NF-NBS-P replaced with NF-ACF+P in last 2.5 min indicating FA, MO, KO boundaries, and ground truth

and occupy different frequency bins in the spectrum. We use the NBS extracted in Section 2.2 from all the available audios. We built a Gaussian Mixture Model (GMM) on NBS with five mixtures, one for each class – mridangam, ghatam, khanjira, mrid-ghat, mrid-khan. If the ratio of the number of frames from any two classes to the total number of frames in a concert is greater than 20%, then that concert is classified as having multiple instruments. Otherwise, we verify if most frames are from mridangam (at least 80%) and classify it as single instrument mridangam. We performed GMM classification on MFCC features as well. Both methods gave 100% classification accuracy in detecting multiple percussion instruments in a recording.

3.2 Novelty Function Computation

The aim is to get an NF whose peaks indicate the desired segment boundaries. Given the ACF, ACF-P, NBS, and NBS-P feature vectors, the Self-Similarity Matrices (SSM) are computed on each of them using L_2 distance measure [10]. The SSM obtained on the ACF, ACF-P, NBS, and NBS-P are displayed in Figure 3. The homogeneous segments of length L frames possibly appear as $(L \times L)$ blocks. The section change points with high contrast in SSM are captured by convolving a checker-board kernel along the diagonal of SSM [15]. The 1D output obtained is called a novelty function (NF). The peaks of the NF indicate the segment boundary instances having high contrast in SSM. The obtained NFs are (1) the average of NF-ACF, NF-ACF-P (Figure 4(b), Figure 5(a)), (2) NF-NBS, and NF-NBS-P (Figure 4(e).

NFs are computed by convolving $(15s \times 15s)$ kernel with SSM of different features. Peak picking is performed by maintaining a minimum distance between adjacent peaks as 5s. We experimented with smaller kernel sizes such as $(3s \times 3s)$, and $(5s \times 5s)$, resulting in noisy NFs. This decreased the precision due to a lot of false positives. Though much larger kernel sizes, such as $(50s \times 50s)$, made the NFs smoother, they compromised in

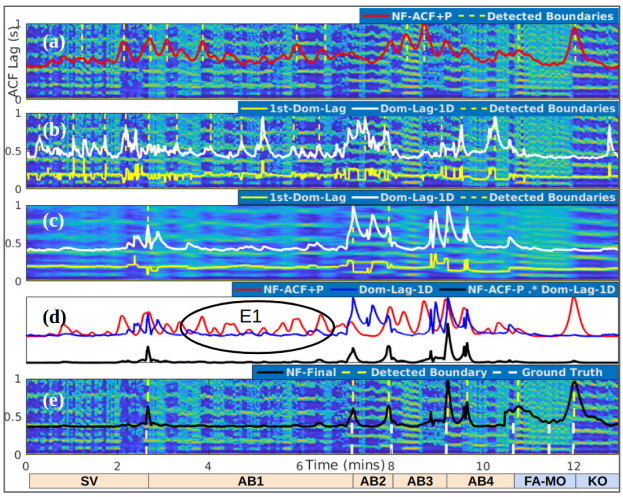


Figure 5. Eg: Solo Mridangam Tani, (a) ACF Rhythmogram with NF-ACF+P overlay-ed along with detected peaks (b) First Dominant peak along the Lag axis FDL (yellow), and its $1st$ diff. FDL-1D highlighting the discontinuities (c) FDL computed on the Gaussian smoothed ACF (yellow) and its $1st$ diff. FDL-1D (white) (d) NF-FDL-ACF (black) is a point-wise product of NF-ACF+P (red) and FDL-1D (blue) (e) NF-FDL-ACF replaced with NF-ACF+P in last 2.5 min indicating FA, MO, KO boundaries along with ground truth, and the segment labels below

resolving the closer boundaries. All the features and NFs in this work are computed at the resolution of 0.5 seconds.

3.3 Case1: Multiple Instrument Tani

In the case of multiple instrument tani, each round of individual percussion elaboration is considered one abhiprāya (one thematic development). Thus instrument change point detection is necessary and sufficient for getting the abhiprāya boundaries. Since the instrument change points are visually evident from the NBS, we used NF-NBS and NF-NBS-P to get the boundaries. A NF obtained from a smaller kernel enhances the rapid instrument change in the KP section, useful in localizing the KP section but creating false positives during segmentation. The first portion of the KP section is fairly large. A larger kernel emphasizes only the start instance of KP by suppressing the rapid instrument change. Thus we used NF obtained from a larger $(15s \times 15s)$ kernel for the segmentation task and the smaller $(3s \times 3s)$ kernel NF for localizing the KP section.

The FA, MO, and KO are always played toward the end of the tani and the FA has a higher ASD. As we see in the Figure 4(e), NF-NBS and NF-NBS-P do not capture these change points. Thus we replace the last two and a half minutes of NF-NBS-P with the average of NF-ACF and NF-ACF-P (NF-ACF+P). This gives the final NF ('red' curve in Figure 4(g)) in the case of multiple instrument tani. We empirically choose the last 2.5 mins as the FA, MO, KO are always found in the last 2.5 mins in the entire dataset.

3.4 Case2: Solo Mridangam Tani

Computing the AB boundaries on solo mridangam tani is a tough task, as the AB change needs to be detected based on the rhythm (naḍe) change. Naḍe change detection is pivotal in getting the AB boundaries, especially in the case of solo mridangam tani. Relying only on the raw rhythmogram features (NF-ACF+P) creates false alarms due to multiple tempo changes and irregularities within a single AB segment. This necessitates the computation of a robust function to tempo octave changes but also captures the non-octave tempo changes that indicate the naḍe changes.

We initially set to track the first peak along the lag axis of the rhythmogram over time, and the change in the peak lag apart from doubling and halving is expected to indicate the naḍe change. But this is also found to be noisy ('yellow' curve in Figure 5(b)). Thus, we perform horizontal Gaussian smoothing on the rhythmogram to mask the irregularities, then pick the first dominant lag peak (FDL). This fetched a smoother curve ('yellow' curve in Figure 5(c)) having discontinuities around the naḍe change with less tempo octave errors. The peaks on the first difference of this curve (FDL-1D) gave fairly good naḍe change estimates, along with a few false positives. We can observe that the peaks of both NF-ACF+P and FDL-1D (Figure 5(d)-E1) coincide around the naḍe change instances but not elsewhere. Thus we perform *"AND"* operation by multiplying NF-ACF+P and FDL-1D to get a NF which is an indicator of naḍe change. We can observe that the false positives are considerably reduced. Again we can see that towards the last FA-MO-KO portion, this NF is not indicating FA-MO-KO boundaries. Thus, we replace the last two and a half minutes of NF-FDL-ACF with NF-ACF+P, similar to Case1. This gives the final NF in the case of solo mridangam tani ('black' curve in Figure 5(e)).

3.5 Section Classification and Labeling

Given the hypothesized segment boundaries, the task is to classify each segment with appropriate labels. Each section, AB, KP, FA, MO, and KO, has unique structural, positional, and duration characteristics common across the concerts. We use the characteristic musical cues to classify and label the segments. For the multiple instrument tani, a NF obtained from a smaller kernel $(3s \times 3s)$ gives multiple peaks in the KP portion. The hypothesized segment having multiple peaks is labeled as KP [Figure 4(e)(E-2)]. The segments before the KP are classified broadly as AB. We compute the mean of ASD in each segment. As the ASD is high during FA-MO, the segment after KP having the highest mean-ASD is labeled FA [Figure 4(c)(E-1)], followed by KO at last. Labeling of FA, MO, and KO is the same for solo mridangam concerts as well. Korapu is not present if only mridangam is present. All the segments before FA are broadly labeled as AB for solo mridangam concerts. Thus the algorithm with a set of rules based on the structure of tani and the domain knowledge performs classification and labeling. Implementation, annotations, and dataset details are shared for research purposes [1].

4. ANALYSIS OF RESULTS AND DISCUSSION

The tani structural segmentation task is approached as a boundary detection task, where the presence or absence of

[1] https://bit.ly/3XIJfMa

Case	Section	Precision	Recall	F1-Score
Multiple Percussion	AB	0.92	0.99	0.96
	KP-FA-MO-KO	0.82	0.89	0.86
	Overall	0.87	0.94	**0.91±0.03**
Single Percussion	AB	0.7	0.82	0.74
	FA-MO-KO	0.82	0.86	0.83
	Overall	0.75	0.84	**0.79±0.05**

Table 3. Segmentation Results

a boundary is examined in uniformly spaced feature frames of 0.5 seconds. Unlike stroke onset detection, the task is addressed at a larger time scale and thus has a tolerance duration in "seconds" rather than milliseconds [27, 47]. A true-positive detection is one where the prediction boundary falls within ± 3 seconds of the ground truth boundary, while a false-positive detection is one where it does not. Precision, recall, and F1-scores are used for evaluation. Evaluation is performed on the entire dataset, as the proposed method is unsupervised, and no model training is done.

The segmentation evaluation scores for each case and individual sections are tabulated in Table 3. The recall is good in all cases, indicating that the system successfully detects the desired boundaries considerably. We can observe that the precision is consistently less than recall, indicating false positives. The change in local rhythm structure, which may be both gradual and abrupt, causes peaks in the NFs. The gradual change in rhythm structure can be seen often in the AB section as it is extempore.

In Case 1, the AB boundaries are identical to the instrument switching instances, and the NF-NBS/NF-NBS-P captured it well with a 0.96 F1-score. The KP-FA-MO-KO section performance is slightly lower, as the rapid instrument switching caused false positives. The end of the KP section is not always evident as the cycle duration reduces to one beat. A small SV pattern may also exist after KP while moving towards FA, making boundary detection challenging. Since MO is played along with or immediately follows the FA, the FA-MO boundary is often missed, reducing recall.

In Case 2, the AB boundaries are not straightforward. The local variations, tempo doubling and halving cause false positives when the NF-ACF and NF-ACF-P are used. These local variations also cause the first dominant lag on the ACF to be noisy. The horizontal averaging of the rhythmogram aided in noise-free first dominant lag tracing and considerably reduced false positives, but still, the false alarms persisted. The naḍe changes are also very gradual in many cases, which are not evident with tempo-related ACF analysis. For example, while transiting from 6 to 5 strokes per beat, the change is hardly noticeable when the metric tempo is fast. A few of the AB boundaries are also missed during smoothing. The performance on the FA-MO-KO is similar to Case:1, as the NF-ACF+P is used in the last 2.5 mins for both cases. Case 2 has more variance in F1-Score than Case 1. The average F1-score for both cases combined is 0.83.

We also experimented with ± 5s and ± 1s tolerance windows. The overall recall increased by 0.2 with a marginal increment in precision for the ± 5s case. The ± 1s case reported a drop of precision and recall by 0.4 and 0.3, respectively. This is evident as 1s corresponds to only two feature frames in this work, and many boundaries are missed.

Section classification performance is evaluated by considering the ground truth markings. We quantify the performance of calculating the ratio of correctly classified frames to the total number of frames in a tani. The weighted average of correctly classified frames in the entire dataset considering the lengths of each tani is 92%. That is, given 10m of segmented tani, around 9m-15s of the frames are correctly labeled as AB, KP, FA, MO, KO.

5. CONCLUSIONS

This work has addressed an unexplored problem, structural segmentation, and labeling of tani audios. We motivate the problem and present different facets and challenges in the task. From the experiments performed, it is clear that individual features alone are inadequate for segmentation. A culture-specific approach is clearly required, both in feature choice and modeling. Timbre is used when it is required to detect if multiple instruments are present in the tani, and MFCC features were found to be adequate. On the other hand, detecting AB sections required analysis of both timbre and rhythmogram to detect boundaries. Identifying AB sections when two percussion instruments are present is quite easy. In contrast, determining AB sections in a solo percussion instrument is difficult as naḍe changes/speed changes are difficult to determine. The hope is that such a task will aid in including additional meta-data w.r.t a concert.

The major contributions of this work are as follows: (i) curating a diverse dataset of tani recordings of around 12 hours having section boundary information along with primary section labels, (ii) evaluating the existing MIR techniques with culture-specific adaptation for a musicologically important task, segmentation and labeling of tani, (iii) formulating average stroke density (ASD) feature (a representative of surface tempo), which is robust to tempo octave errors, (iv) formulating the class-conditional probability features from the rhythmogram, and spectral features, and (v) exploring the combination of different NFs obtained from different features to achieve the task. Finally, this work provides an example of adapting available MIR methods to genre-specific problems by performing appropriate feature engineering.

6. ACKNOWLEDGMENTS

The authors are grateful to the percussion maestros V Selvaganesh, Patri Satish Kumar and Giridhar Udupa for their support and help. We are thankful to Ajay Srinivasamurthy for his support and timely guidance. We thank Jom Kuriakose for sharing the dataset audios.

7. REFERENCES

[1] U. Giridhar. (2020) Description of tani avartanam. [Online]. Available: https://www.ghatamudupa.com/

[2] E. N. Sunil, *Resounding Mridangam: The Majestic South Indian Drum*. Erickavu N Sunil, March 2021. [Online]. Available: https://www.youtube.com/c/erickavunsunil

[3] Charsur digital workstation. [Online]. Available: https://musicbrainz.org/label/3e188240-9eb5-4842-b7b9-d6c2393211b7

[4] N. Dawalatabad, J. Kuriakose, C. C. Sekhar, and H. A. Murthy, "Information bottleneck based percussion instrument diarization system for taniavartanam segments of carnatic music concerts." in *INTERSPEECH*, 2018.

[5] Sangeethapriya – indian fine arts. [Online]. Available: https://www.sangeethapriya.org/

[6] J. Kuriakose, J. C. Kumar, P. Sarala, H. A. Murthy, and U. K. Sivaraman, "Akshara transcription of mrudangam strokes in carnatic music," in *Twenty First National Conference on Communications (NCC) 2015*.

[7] A. Srinivasamurthy, A. Holzapfel, A. T. Cemgil, and X. Serra, "Particle filters for efficient meter tracking with dynamic bayesian networks," in *Proc. 16th International Society for Music Information Retrieval (ISMIR), Málaga, Spain. Canada*, 2015.

[8] A. Srinivasamurthy, G. K. Koduri, S. Gulati, V. Ishwar, and X. Serra, "Corpora for music information research in indian art music," in *Proc. International Computer Music Conference, ICMC/SMC; Athens, Greece.*, 2014.

[9] R. B. Dannenberg and M. Goto, "Music structure analysis from acoustic signals," in *Handbook of signal processing in acoustics*. Springer, 2008, pp. 305–331.

[10] J. Paulus, M. Müller, and A. Klapuri, "State of the art report: Audio-based music structure analysis." in *Proc. 11th International Society for Music Information Retrieval (ISMIR)*. Utrecht, 2010, p. 625–636.

[11] O. Nieto, "Discovering structure in music: Automatic approaches and perceptual evaluations," Ph.D. dissertation, New York University, 2015.

[12] S. Padi and H. A. Murthy, "Segmentation of continuous audio recordings of carnatic music concerts into items for archival," *Sādhanā*, vol. 43, no. 10, pp. 1–20, 2018.

[13] P. Rao, T. P. Vinutha, and M. A. Rohit, "Structural segmentation of alap in dhrupad vocal concerts," *Transactions of the International Society for Music Information Retrieval*, vol. 3, no. 1, 2020.

[14] P. Grosche, M. Müller, and F. Kurth, "Cyclic tempogram—a mid-level tempo representation for musicsignals," in *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2010.

[15] J. Foote, "Automatic audio segmentation using a measure of audio novelty," in *Proc. International Conference on Multimedia and Expo. (ICME)*. IEEE, 2000.

[16] D. Turnbull, G. R. Lanckriet, E. Pampalk, and M. Goto, "A supervised approach for detecting boundaries in music using difference features and boosting." in *Proc. 8th International Society for Music Information Retrieval (ISMIR)*, 2007.

[17] K. Ullrich, J. Schlüter, and T. Grill, "Boundary detection in music structure analysis using convolutional neural networks." in *ISMIR*, 2014, pp. 417–422.

[18] 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 *Proc. 22nd International Society for Music Information Retrieval (ISMIR)*, 2011.

[19] M. A. Rohit and P. Rao, "Structure and automatic segmentation of dhrupad vocal bandish audio," *Unpublished technical report*, 2020.

[20] M. A. Rohit, T. P. Vinutha, and P. Rao, "Structural segmentation of dhrupad vocal bandish audio based on tempo," in *Proc. International Society for Music Information Retrieval (ISMIR)*, 2020.

[21] P. Verma, T. P. Vinutha, P. Pandit, and P. Rao, "Structural segmentation of hindustani concert audio with posterior features," in *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE*, 2015.

[22] T. P. Vinutha, S. Sankagiri, K. K. Ganguli, and P. Rao, "Structural segmentation and visualization of sitar and sarod concert audio." in *Proc. 17th International Society for Music Information Retrieval (ISMIR)*, 2016.

[23] K. S. PV, S. Sankaran, and H. Murthy, "Segmentation of carnatic music items using kl2, gmm and cfb energy feature," in *Proc. Twenty Second National Conference on Communication (NCC)*. IEEE, 2016.

[24] H. Ranjani and T. Sreenivas, "Hierarchical classification of carnatic music forms," in *Proc. 14th International Society for Music Information Retrieval (ISMIR)*, 2013.

[25] B. Thoshkahna, M. Müller, V. Kulkarni, and N. Jiang, "Novel audio features for capturing tempo salience in music recordings," in *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015.

[26] T. P. Vinutha, S. Sankagiri, and P. Rao, "Reliable tempo detection for structural segmentation in sarod concerts," in *Proc. Twenty Second National Conference on Communication (NCC)*. IEEE, 2016.

[27] R. Gowriprasad and K. S. R. Murty, "Onset detection of tabla strokes using lp analysis," in *Proc. International Conference on Signal Processing and Communications (SPCOM)*. IEEE, 2020.

[28] P. A. M. Kumar, J. Sebastian, and H. A. Murthy, "Musical onset detection on carnatic percussion instruments," in *Proc. Twenty First National Conference on Communications (NCC)*, 2015.

[29] O. Gillet and Richard, "Automatic labelling of tabla signals," in *Proc. 4th International Society for Music Information Retrieval (ISMIR)*, 2003.

[30] P. Chordia, "Segmentation and recognition of tabla strokes." in *Proc. 6th International Society for Music Information Retrieval (ISMIR)*, 2005.

[31] K. Samudravijaya, S. Shah, and P. Pandya, "Computer recognition of tabla bols," Technical report, Tata Institute of Fundamental Research, Tech. Rep., 2004.

[32] M. A. Rohit, A. Bhattacharjee, and P. Rao, "Four-way classification of tabla strokes with models adapted from automatic drum transcription," in *Proc. 22nd International Society for Music Information Retrieval (ISMIR)*, 2021.

[33] A. Anantapadmanabhan, A. Bellur, and H. A. Murthy, "Modal analysis and transcription of strokes of the mridangam using non-negative matrix factorization," in *IEEE International Conference on Acoustics, Speech and Signal Processing, 2013*.

[34] P. Chordia, A. Sastry, and S. Şentürk, "Predictive tabla modelling using variable-length markov and hidden markov models," *Journal of New Music Research*, vol. 40, no. 2, pp. 105–118, 2011.

[35] P. Chordia, A. Sastry, T. Mallikarjuna, and A. Albin, "Multiple viewpoints modeling of tabla sequences." in *Proc. 11th International Society for Music Information Retrieval (ISMIR), 2010*.

[36] S. Gupta, A. Srinivasamurthy, M. Kumar, H. A. Murthy, and X. Serra, "Discovery of syllabic percussion patterns in tabla solo recordings," in *Proc. 16th International Society for Music Information Retrieval (ISMIR); 2015*.

[37] A. Srinivasamurthy, "A data-driven bayesian approach to automatic rhythm analysis of indian art music," Ph.D. dissertation, Universitat Pompeu Fabra, 2017.

[38] K. Gogineni, J. Kuriakose, and H. A. Murthy, "Mridangam artist identification from taniavartanam audio," in *Proc. Twenty Fourth National Conference on Communications (NCC)*. IEEE, 2018.

[39] R. Gowriprasad, V. Venkatesh, H. A. Murthy, R. Aravind, and K. S. R. Murty, "Tabla Gharana Recognition from Audio Music recordings of Tabla Solo performances," in *Proc. 22nd International Society for Music Information Retrieval Conference*, 2021.

[40] R. Gowriprasad, V. Venkatesh, and S. R. Murty K, "Tabla gharana recognition from tabla solo recordings," in *Proc. National Conference on Communications (NCC)*, 2022.

[41] 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.

[42] S. Dixon, "Simple spectrum-based onset detection," *MIREX 2006*, p. 62, 2006.

[43] K. Jensen, "Multiple scale music segmentation using rhythm, timbre, and harmony," *EURASIP Journal on Advances in Signal Processing*, vol. 2007, pp. 1–11, 2006.

[44] 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, 2010.

[45] K. K. Jensen, "Rhythm-based segmentation of popular chinese music," in *Proc. 6th International Society for Music Information Retrieval (ISMIR)*, 2005.

[46] G. Peeters, "Template-based estimation of time-varying tempo," *EURASIP Journal on Advances in Signal Processing*, vol. 2007, pp. 1–14, 2006.

[47] J. P. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and M. B. Sandler, "A tutorial on onset detection in music signals," *IEEE Transactions on speech and audio processing*, vol. 13, no. 5, pp. 1035–1047, 2005.

TRANSFER LEARNING AND BIAS CORRECTION WITH PRE-TRAINED AUDIO EMBEDDINGS

Changhong Wang[1] **Gaël Richard**[1] **Brian McFee**[2]
[1]LTCI, Télécom Paris, Institut Polytechnique de Paris, France
[2]Music and Audio Research Laboratory, New York University, USA
changhong.wang@telecom-paris.fr

ABSTRACT

Deep neural network models have become the dominant approach to a large variety of tasks within music information retrieval (MIR). These models generally require large amounts of (annotated) training data to achieve high accuracy. Because not all applications in MIR have sufficient quantities of training data, it is becoming increasingly common to transfer models across domains. This approach allows representations derived for one task to be applied to another, and can result in high accuracy with less stringent training data requirements for the downstream task. However, the properties of pre-trained audio embeddings are not fully understood. Specifically, and unlike traditionally engineered features, the representations extracted from pre-trained deep networks may embed and propagate biases from the model's training regime.

This work investigates the phenomenon of bias propagation in the context of pre-trained audio representations for the task of instrument recognition. We first demonstrate that three different pre-trained representations (VGGish, OpenL3, and YAMNet) exhibit comparable performance when constrained to a single dataset, but differ in their ability to generalize across datasets (OpenMIC and IRMAS). We then investigate dataset identity and genre distribution as potential sources of bias. Finally, we propose and evaluate post-processing countermeasures to mitigate the effects of bias, and improve generalization across datasets.

1. INTRODUCTION

Transfer learning generally refers to the concept of adapting a model for one task to solve another task. Often, this is achieved by extracting the internal representation (an *embedding*) of input data from a pre-trained neural network, and providing it as input features to some (often simpler) *downstream* model for the target task. While this approach is increasingly common and effective, pre-trained embedding models may encode and propagate implicit biases which can have detrimental and disparate population-dependent effects. Biases have caught wide attention from research fields such as natural language processing (NLP) [1–3], cognitive science [4], and computer vision [5], while in music information retrieval (MIR), bias of pre-trained audio embeddings, is under-explored.

In this paper, we identify and address the bias of different pre-trained audio embeddings for transfer learning on the task of instrument recognition. We summarize the contributions as following. (1) We study the within- and cross-domain performance of three pre-trained audio embeddings (VGGish, OpenL3, YAMNet) on two instrument datasets (IRMAS and OpenMIC-2018). (2) We demonstrate that this approach can propagate bias by producing classifiers which are sensitive to the source domain (dataset). (3) Based on the performance variation in cross-domain generalization, we investigate dataset identity and genre distribution as potential sources of bias. (4) We propose a post-processing countermeasure to mitigate unwanted bias in the representation. We experiment different bias correction strategies, and analyze the robustness of each pre-trained audio embedding. The proposed strategies make use of relatively little additional information, and generally produce a modest improvement to cross-domain accuracy for the instrument recognition task. Our code for all experiments is publicly available [1] .

2. RELATED WORK

Pre-trained embeddings are becoming increasingly used in transfer learning for audio-related tasks. Choi et al. [6] presented a transfer learning approach for music classification and regression tasks using the internal activations of a pre-trained convolutional network as features. The network was trained on the source task of music tagging, and the learned representation was then transferred to five target tasks, including genre classification, vocal/non-vocal classification, emotion prediction, speech/music classification, and acoustic event classification. Other well-known audio embedding models include OpenL3 [7], VGGish [8], and YAMNet [2] . The OpenL3 is a 512-dimensional embedding model that results from self-supervised training of the look-listen-learn (L3)-Net for audiovisual correlations. VGGish (128-dimensional) and YAMNet (1024-

© Changhong Wang, Gaël Richard, and Brian McFee. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Changhong Wang, Gaël Richard, and Brian McFee, "Transfer Learning and Bias Correction with Pre-trained Audio Embeddings", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

[1] https://github.com/changhongw/audio-embedding-bias
[2] https://github.com/tensorflow/models/tree/master/research/audioset/yamnet

dimensional) are both embeddings derived from classification models trained on AudioSet [8]. Although these embedding models differ in the architecture of the network, source data, and training regime, they have each demonstrated good and comparable generalization performance for a variety of tasks.

Despite that embedding models are normally trained on large amounts of data, it inevitably encodes biases due to the limitation of collected data. This problem can be especially prevalent in models trained via unsupervised or self-supervised strategies, where there may be no incentive for the model to learn invariances or equivalencies in the data beyond what is required for by the training objective. As a result, pre-trained embeddings may propagate unwanted biases to downstream tasks. Different types of biases and bias correction methods are explored in the NLP literature, such as gender [9], race and religion [10]. A general approach for addressing gender bias in word embedding was proposed by Bolukbasi et al., following three steps: identify bias direction, remove bias by projecting out the bias direction, and equalize pairs [9].

Besides field-specific biases, dataset bias is a general type of bias that could happen in any application domain. Tommasi et al. [5] investigated dataset bias in visual recognition with a cross-dataset testbed comprising 12 different datasets. Ganin et al. [11] proposed adversarial training for domain adaptation to reduce sensitivity to data drawn from similar but different distributions. When detecting depression, a mental health disorder, from speech, Bailey and Plumbley [12] found that biases in dataset could result in skewed classification performance.

The approach we take in this paper is most similar to those of Bolukbasi et al. [9] and Ganin et al. [11]. While Bolukbasi et al.'s method requires numerous paired examples to identify a subspace which encodes undesirable bias, our proposed method works at the level of collection statistics rather than individual correspondence, and may be easier to apply for audio applications. Similarly, Ganin et al.'s method requires adversarial training of the initial model to produce a representation which cannot discriminate well between subsets of data that should be treated equivalently. Our approach is implemented as a post-processing step, and can be applied to any pre-trained model. While we focus in this work on dataset identity as a concrete source of bias, we emphasize that the method should be generally applicable to other scenarios in which audio representations exhibit unwanted sensitivity to identifiable attributes.

3. METHODS

We consider embedding bias from the perspective of *domain adaptation*. Unlike transfer learning, which relates to the *output* of the model, domain adaptation refers to the behavior of a model (classifier, regressor, etc.) under changes to the distribution of *input data*. This is closely related to *representation bias*, which is one among many forms of bias known to impact machine learning systems as enumerated by Mehrabi et al. [13]. If a classifier is trained on a sample of (labeled) data which is not representative of the target population, then we expect the model to generalize poorly. The degree to which a pre-trained audio embedding is sensitive to differences between populations of interest—e.g., between a dataset annotated for instrumentation, compared to other collections of music—is therefore of principal interest [14].

3.1 Domain sensitivity

We investigate the domain sensitivity of three pre-trained embeddings (OpenL3, VGGish, and YAMNet) in transfer learning for the downstream task of instrument recognition. Each embedding is evaluated in both within-domain and cross-domain setting. For within-domain evaluation, we train and test the embedding in a single dataset; while for cross-domain case, we investigate the domain adaptation capability of the embedding models across datasets, i.e. training and testing the downstream classifier using data from different datasets. As a study case, we consider two well-known datasets for instrument recognition, i.e. IRMAS [15] and OpenMIC-2018 [16] (see Section 4.1 for dataset details).

Fig. 1 (a) and (d) visualize the within-domain (IRMAS–IRMAS and OpenMIC–OpenMIC) recognition results in terms of area under the receiver operating characteristic curve (ROC-AUC) using the three embeddings above for each of the ten instrument classes. [3] All three embeddings achieve comparable results, although there is a loose performance ranking of YAMNet $>$ OpenL3 $>$ VGGish for most instrument classes.

When generalizing across domains, performance degradation happens for both cross-dataset pairs, as shown in Fig. 1 (b) IRMAS–OpenMIC and (c) OpenMIC–IRMAS. The performance ranking of the three embedding does not persist either. Comparing the results when testing on OpenMIC, i.e. (b) and (d), only *voice, piano,* and *guitar* retain close results. For the remaining instrument classes, all three embeddings exhibit diminished performance. Similar trends take place in the comparison between (a) and (c) where the test set is IRMAS.

Surprisingly, a dramatic performance drop happens for the *organ* class. Examining this class in both datasets, we notice a large distribution difference on genre, as shown in Fig. 2. Organ in IRMAS is confined to *pop/rock* and *jazz/blue* genres, suggesting that examples mostly contain electric organ sounds (e.g. Hammond B3). The distribution of organ in OpenMIC is more balanced, but dominated by classical recordings which are more likely to contain pipe organ than electric. These differences aside, we generally expect the instrument labels to refer to similar sounds across domains.

3.2 Quantifying domain bias

To quantify the effect of domain bias, we first obtain the domain separation direction vector $\boldsymbol{w} \in \mathbb{R}^D$ by fitting a linear discriminant analysis (LDA) model to discriminate

[3] We report AUC because it is invariant to overall class proportions and decision thresholds—which vary between datasets—and thereby allows us to focus on the separating directions identified for each class.

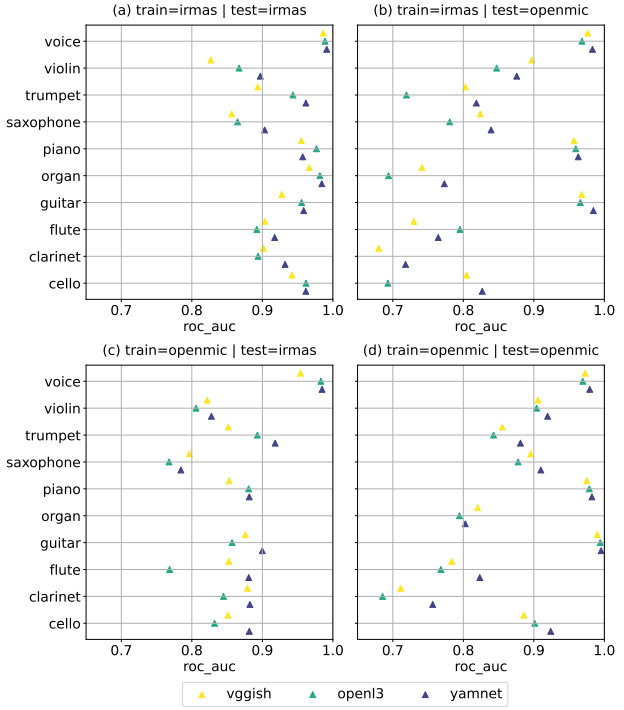


Figure 1. Within-domain (a, d) and cross-domain (b, c) performance of pre-trained audio embeddings (VGGish, OpenL3, and YAMNet) on instrument recognition in the IRMAS and OpenMIC datasets. ROC-AUC refers to area under the receiver operating characteristic curve.

between the OpenMIC and IRMAS datasets in each representation (VGGish, OpenL3, and YAMNet). $\boldsymbol{v}_k \in \mathbb{R}^D$ is the instrument separation direction vector, i.e. the coefficient vector of the trained downstream classifier, for the k-th instrument. $k = 1, 2, ..., K$ is the instrument class index and D is the dimension of pre-trained embedding. We measure the correlation between the domain separation and downstream classification using the cosine similarity between $\boldsymbol{w}$ and $\boldsymbol{v}_k$:

$$\boldsymbol{c}_k(\boldsymbol{w}, \boldsymbol{v}) = \frac{\langle \boldsymbol{w}, \boldsymbol{v}_k \rangle}{\|\boldsymbol{w}\| \times \|\boldsymbol{v}_k\|} \tag{1}$$

Large (in magnitude) $\boldsymbol{c}_k$ indicates that the instrument classifier is sensitive to dataset identity.

Fig. 3 top shows the absolute correlation value for each instrument class, when the classifier is trained on the training set of (a) IRMAS and (b) OpenMIC dataset, respectively. The mean correlation value over all instruments for each embedding is displayed in the legend. It clearly shows that YAMNet is the least sensitive to dataset bias; OpenL3 is also relatively stable while VGGish is the most sensitive to dataset bias. The relatively large correlation value for the organ class matches our analysis in Section 3.1 that genre distribution might be also a potential source of bias (see Fig. 2). Although the sensitivity of different embeddings to dataset bias are different, bias cannot generally be removed by simply using different pre-trained embeddings. As we will demonstrate, explicitly correcting for dataset bias can potentially improve domain adaptation performance for each choice of embedding.

3.3 Bias correction

To mitigate domain bias, we propose a post-processing countermeasure on the pre-trained embeddings which does not interact with the training process of embeddings. Importantly, the proposed method requires only samples of data which should behave similarly for the downstream task, but it does not require these samples to be *labeled* for the downstream task.

Continuing our instrument classification example, given that both datasets contain examples from each of the instrument categories of interest, we should expect that a well-formed linear classifier should behave independently of the domain from which data is drawn. Concretely, this means that the separating direction $\boldsymbol{v}_k$ should be orthogonal to any direction $\boldsymbol{w}$ which separates the two datasets in the embedding space, resulting in $\boldsymbol{c}_k(\boldsymbol{w}, \boldsymbol{v}_k) = 0$. While Ganin et al. [11] use this intuition to adversarially train the representation, this approach is impractical when using pre-trained embeddings which are presumed to be fixed in advance. Instead, we approach this problem by post-processing the embedding to project out the direction $\boldsymbol{w}$ which separates domains that should be indistinguishable for the downstream task.

Concretely, if $\boldsymbol{w} \in \mathbb{R}^D$ is the domain-separating direction (normalized to unit length, $\|\boldsymbol{w}\| = 1$), we project this dimension out of the space by applying the following transformation to input data $\boldsymbol{x} \in \mathbb{R}^D$:

$$\boldsymbol{x}_{\mathrm{P}} := \left(\mathbf{I} - \boldsymbol{w}\boldsymbol{w}^{\mathsf{T}}\right) \boldsymbol{x} \tag{2}$$

where $\mathbf{I}$ is the $D \times D$ identity matrix. The new embedding $\boldsymbol{x}_{\mathrm{P}}$ is the input to the classifier.

3.4 Multiple bias correction

While the above strategy is defined for binary bias correction, e.g. where there are two domains to be reconciled, it does generalize to more complex settings. In the instrument recognition example, we may also consider differences between genres across datasets as a source of bias. Even if two datasets both consist of examples in the same genre categories, this does not necessarily mean that the genre terms are used consistently between datasets.

To consider the influence of genre distribution, we propose also *multiple bias correction*, where we extract the dataset separation direction in the genre subspace. That is, for each pair of matched genre labels, e.g. pop/rock in IRMAS and pop/rock in OpenMIC, we fit a binary LDA to separate them. Then for each genre category $g = 1, 2, \ldots G$ (for $G \geq 1$ genres), we obtain a dataset separation direction vector $\boldsymbol{w}_g$ which depends only on examples from genre g. Collecting all $\boldsymbol{w}_g$ into a matrix $W \in \mathbb{R}^{D \times G}$ defines a basis for a subspace of the embedding of dimension at most G. Note that W may not be an orthogonal basis, as different $\boldsymbol{w}_g$ may correlate with each other.

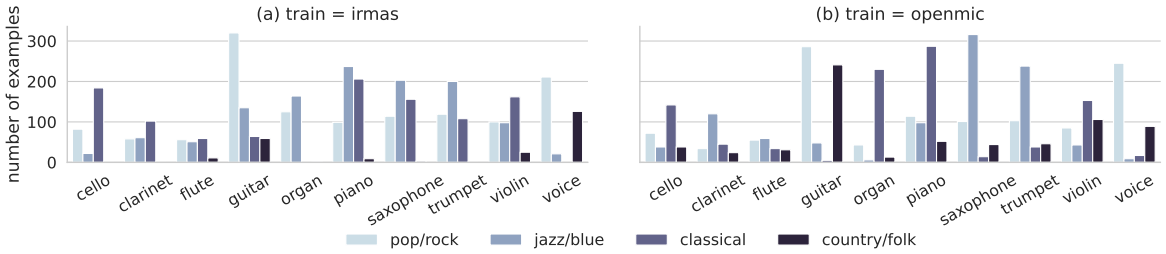


Figure 2. Number of genre examples for each instrument in the training set of IRMAS and OpenMIC datasets. We align the genre labels according to those in the IRMAS dataset: pop/rock, jazz/blue, classical, and country/folk.

We therefore derive an orthogonal basis by factorizing W via the reduced singular vector decomposition (SVD):

$$W = U\Sigma V^{\mathsf{T}} \tag{3}$$

where Σ is a $G \times G$ diagonal matrix of singular values, and $U \in \mathbb{R}^{D\times G}$ and $V \in \mathbb{R}^{D\times G}$ are the left- and right-singular vectors. We use the right singular vectors as an orthogonal basis for the domain-separating subspace, resulting in the following generalization of Eq. (2):

$$\boldsymbol{x}_{\mathrm{P}} := \left(\mathbf{I} - VV^{\mathsf{T}}\right)\boldsymbol{x} \tag{4}$$

In applying Eq. (4), it is important to verify that W is full rank (G), e.g. by verifying that all singular values Σ are sufficiently large, as Eq. (4) would otherwise remove a larger than necessary subspace from the representation. In all cases studied in this work, W was full rank.

3.5 Nonlinear bias correction

The above methods are based on two assumptions: 1) that the downstream model will be linear, and 2) that the domains are linearly separable. These assumptions may be restrictive in practice, so we generalize the method above by transforming the embeddings to a higher dimensional space using kernel methods. While both logistic regression and linear discriminant analysis support kernel generalizations [17], the subspace projection method described above is less directly adaptable. [4]

Instead of using implicit kernel representations, we will use approximate, i.e. explicit kernel approximation. That is, instead of replacing inner products $\langle \boldsymbol{w}, \boldsymbol{v}\rangle$ by nonlinear kernel function calculations $k(\boldsymbol{w}, \boldsymbol{v})$, we apply an explicit nonlinear transformation $f : \mathbb{R}^D \to \mathbb{R}^{D'}$ such that

$$\langle f(\boldsymbol{w}), f(\boldsymbol{v})\rangle \approx k(\boldsymbol{w}, \boldsymbol{v}) \tag{5}$$

We then apply the previously defined bias correction methods on the transformed data $f(\boldsymbol{w})$, which results in projecting out the dataset-separating direction(s) after applying f but prior to fitting the downstream (instrument) classifiers.

There are several choices to be made here when selecting the kernel k and the approximating map f. In this work, we use a standard radial basis function (Gaussian) kernel and the "random Fourier features" approximation method [18]. However, we note that other choices (e.g., the Nyström method) are readily available in scikit-learn [19], and may work just as well.

In total, we have four bias-correction strategies: linear bias correction (*LDA*), linear multiple bias correction (*mLDA*), nonlinear bias correction in the kernelized embedding space (*KLDA*), nonlinear multiple bias correction in the kernelized embedding space (*mKLDA*).

4. EXPERIMENTS

4.1 Datasets and experimental details

The datasets we use are two well-known datasets with instrument annotations, IRMAS [15] and OpenMIC-2018 [16]. The former comprises 20,000 examples of 10-second excerpts, partially labeled for the presence or absence of 20 instrument classes; and the latter contains 6705 audio files of 3-second clips, of which only the predominant instrument were annotated. Since there are 20 instrument classes in the OpenMIC dataset and 11 in the IRMAS, we focus only on the 10 mutual classes: *cello, clarinet, flute, guitar, organ, piano, saxophone, trumpet, violin,* and *voice*. For the sake of consistency, *electric guitar* and *acoustic guitar* in the IRMAS dataset have been merged into a single class: guitar.

To investigate the impact of genre, we also align the genres in the two datasets. Each audio sample in the IRMAS dataset is labeled with one of the five genres: *pop/rock, jazz/blue, classical, country/folk,* and *latin/soul*; while samples in the OpenMIC datase has multiple labels from around 130 genres. We consider four genres (pop/rock, jazz/blue, classical, country/folk) as the latin/soul genre has few examples in both datasets. The genre labels of the OpenMIC dataset are merged into those of IRMAS with name intersections. For example, we merge the genres—*Rock, Loud-Rock, Noise-Rock, Psych-Rock*, et. al.—in OpenMIC into one genre label: pop/rock. Multiple genre labels in the OpenMIC dataset are reduced to a single label by the first activation from the four considered genres or the first of the original labels otherwise.

With the embedding features extracted using pre-trained VGGish, OpenL3, and YAMNet models, we train a logistic regression classifier for each instrument class using IRMAS and OpenMIC training data. The input to the

[4] One could achieve a similar effect by adding a linear constraint $\langle \boldsymbol{w}, \boldsymbol{v}\rangle = 0$ to the logistic regression problem, but this would require a custom solver and limit the general utility of the approach.

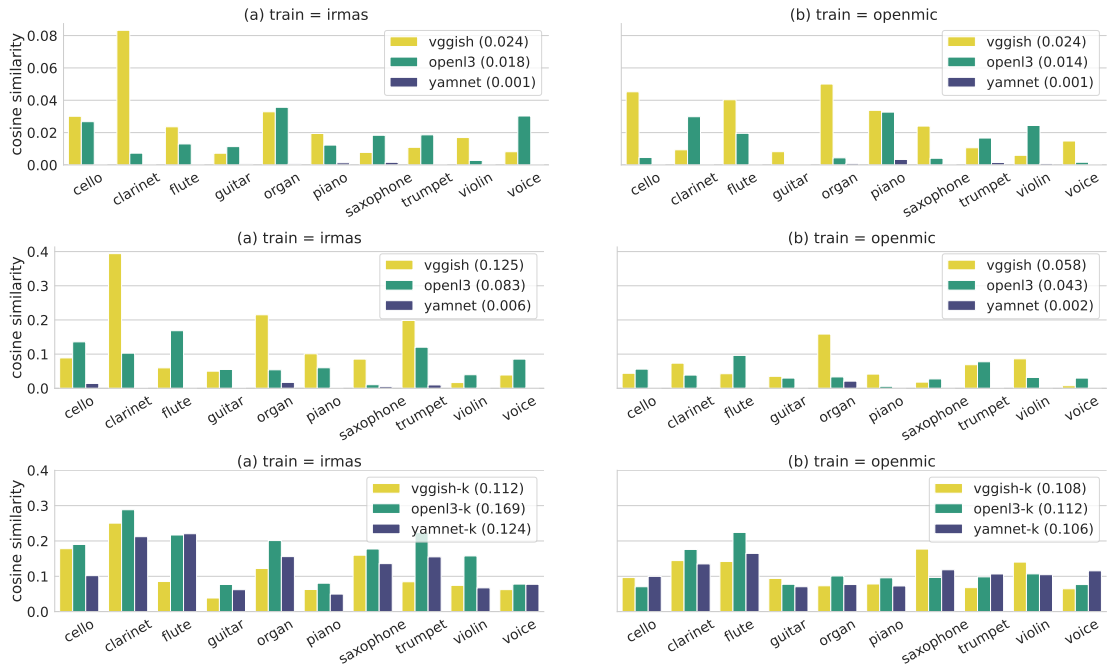


Figure 3. Correlation between domain separation and instrument classification for each instrument in the IRMAS and OpenMIC training set. *(Top)*: correlation in the original embedding space with domain separation direction extracted using only dataset identity; *(middle)*: correlation in the original embedding space with domain separation direction extracted class-wise; *(bottom)*: same as middle but in the kernelized embedding space. Mean value is given in parentheses.

classifier is the mean frame embedding of each audio example. For OpenMIC dataset, we follow the train-test split in [16], with a ratio of 3:1. For fair comparisons, we create a new partition with the same train-test ratio on IRMAS dataset which takes into account of the class-balance and non-track overlap between training and test sets. To focus only on distribution shift, we use the same number of samples per class for both datasets during training, following the lower one.

For the nonlinear method, we first standardize the embedding features using z-score normalization with the training-set statistics. Then we approximate the kernels for the embeddings with a fixed dimension D' of four times the dimension (D) of the original embeddings. Finally, we tune the hyper-parameter for the logistic regression classifier, i.e. the inverse of regularization strength C, by cross-validation with a grid of $10^{-8:1:4}$.

4.2 Results

Table. 1 lists the instrument classification performance of the debiasing methods discussed in Section 3 in terms of mean ROC-AUC over the 10 instrument classes. To compare the performance of using only dataset identity as additional information and that uses also class-labels, we present two sets of results: *global bias correction* and *class-wise bias correction*. We first present some observations that are common to both cases and then discuss the comparison. For the original embeddings (in *italic*), large performance drop shows for all cross-domain cases. OpenL3 is most sensitive to distribution shift, with a drop of 12.7% and 7% when testing on IRMAS and OpenMIC dataset, respectively. Yet, from the cosine similarity values in Fig. 3 top and middle, OpenL3 does not embed the most domain bias. This may indicate that for the task at hand, other more significant distribution shifts that OpenL3 is sensitive to may exist. For all embeddings, projecting to the higher dimensional space (debiasing methods with "K") almost never substantially hurts the within-domain performance and sometimes improves the performance.

Interestingly, when comparing linear debiasing ("-LDA" and "-mLDA") with nonlinear debiasing ("-KLDA" and "-mKLDA") for all embeddings, we find that kernelization does not help for VGGish while YAMNet only works in the kernelized embedding space. This explains the relative increase of cosine similarity values for YAMNet after kernelization as compared to the other two embeddings (see Fig. 3 bottom). Both linear and nonlinear debiasing exhibit performance improvement for OpenL3. In terms of global bias correction, almost no improvement for VGGish except LDA for OpenMIC->IRMAS; OpenL3 yields some boost for both cross-domain cases. YAMNet improves the results only for OpenMIC->IRMAS. It is expected that the class-wise bias correction achieves better performance as we extract the domain bias for the target instrument exactly. This is also verified by the more noticeable cosine similarity values in the middle subfigure as compared to the top of Fig. 3. VGGish and OpenL3 yields slight improvement for most linear debiasing. All nonlinear debiasing improves the results of OpenL3 for IRMAS->OpenMIC and YAMNet for OpenMIC->IRMAS. Although the overall improvement is not significant, we observe large improvements for some instrument classes.

Debiasing method	Global bias correction				Class-wise bias correction			
	Within-domain		Cross-domain		Within-domain		Cross-domain	
	IR-IR	OP-OP	OP-IR	IR-OP	IR-IR	OP-OP	OP-IR	IR-OP
VGGish	*91.6*	*87.95*	*82.82*	*83.81*	*91.60*	*87.95*	*82.82*	*83.81*
VGGish-LDA	91.60	87.99	82.99 (**+0.18**)	83.82 (0.0)	91.60	87.94	82.93 (**+0.12**)	83.85 (+0.03)
VGGish-mLDA	91.45	87.98	82.70 (-0.11)	83.30 (-0.51)	91.56	87.87	83.13 (**+0.31**)	83.66 (-0.16)
VGGish-K	92.24	88.08	82.57 (-0.25)	83.67 (-0.14)	92.24	88.08	82.57 (-0.25)	83.67 (-0.14)
VGGish-KLDA	92.24	88.08	82.58 (-0.24)	83.67 (-0.14)	92.22	88.07	82.70 (-0.12)	83.78 (-0.04)
VGGish-mKLDA	92.22	88.15	82.42 (-0.39)	83.70 (-0.11)	92.26	88.08	82.70 (-0.11)	83.76 (-0.05)
OpenL3	*93.26*	*87.16*	*80.56*	*80.13*	*93.26*	*87.16*	*80.56*	*80.13*
OpenL3-LDA	93.26	87.16	80.56 (+0.01)	80.15 (+0.02)	93.24	87.18	80.59 (+0.04)	80.38 (**+0.26**)
OpenL3-mLDA	93.11	87.16	80.67 (**+0.12**)	79.93 (-0.20)	93.09	87.23	80.57 (+0.02)	80.62 (**+0.50**)
OpenL3-K	93.89	87.91	79.46 (-1.09)	81.23 (**+1.11**)	93.89	87.91	79.46 (-1.09)	81.23 (**+1.11**)
OpenL3-KLDA	93.89	87.84	79.03 (-1.53)	81.23 (**+1.11**)	93.96	87.91	79.99 (-0.57)	81.79 (**+1.66**)
OpenL3-mKLDA	93.88	87.88	79.56 (-1.00)	81.20 (**+1.07**)	94.04	87.83	79.97 (-0.59)	81.32 (**+1.19**)
YAMNet	*94.65*	*89.74*	*85.01*	*85.47*	*94.65*	*89.74*	*85.01*	*85.47*
YAMNet-LDA	94.65	89.74	85.01 (0.0)	85.47 (0.0)	94.65	89.74	85.02 (0.0)	85.47 (0.0)
YAMNet-mLDA	94.65	89.74	85.01 (0.0)	85.47 (0.0)	94.65	89.74	85.02 (0.0)	85.46 (0.0)
YAMNet-K	93.83	89.24	85.87 (**+0.86**)	84.56 (-0.91)	93.83	89.24	85.87 (**+0.86**)	84.56 (-0.91)
YAMNet-KLDA	93.83	89.23	85.87 (**+0.86**)	84.56 (-0.91)	93.63	89.24	86.00 (**+0.99**)	84.76 (-0.70)
YAMNet-mKLDA	93.79	89.19	85.72 (**+0.71**)	84.43 (-1.04)	93.79	89.34	85.53 (**+0.51**)	84.60 (-0.87)

Table 1. Mean ROC-AUC (%) of global bias correction and class-wise bias correction on instrument classification in IRMAS (IR) and OpenMIC (OP) datasets. VGGish, OpenL3, and YAMNet (in *italic*) refers to the original embedding; the other cases, i.e. with -LDA, -mLDA, -LDA, and -mKLDA, correspond to linear, linear-multiple, nonlinear, and nonlinear-multiple debiasing strategies; cases with -K are the kernelized embeddings. Values in parenthesis are the performance boost (>0.1 are **bolded**) or degradation as compared to the original embedding (the closest *underlined* above).

5. DISCUSSION

We notice two important factors for transfer learning with pre-trained audio embeddings: the training regime of the embeddings, and the class vocabulary alignment between the source task and downstream task.

The better generalization performance of YAMNet and VGGish in a transfer setting may be attributed to their training regime. YAMNet and VGGish are derived from supervised training while OpenL3 is from self-supervised training and more prone to overfitting a domain. As a result, YAMNet and VGGish have both been incentivized to learn invariances within specific categories (including musical instrumentation), while OpenL3 has no such incentive as it is only designed to predict audio-visual correspondence. Moreover, YAMNet was specifically trained for sound classification using a vocabluary that broadly subsumes that of our downstream task (instrumentation). This likely contributes to its high performance and cross-domain stability overall.

The class vocabulary alignment is related to *label shift*, an under-explored type of distribution shift in the domain-adaptation field [20]. The labelling scheme difference between the two datasets complicates the debiasing as the IRMAS dataset only contains labels for the predominant instrument while all active instruments are annotated in the OpenMIC dataset. Aligning these two datasets is nontrivial as it involves label shift besides *covariate shift*. We propose multiple-bias correction, i.e. debiasing in the genre subpsace, to deal with this problem. Yet, it does not resolve the conditional probability shift that happens due to unbalanced relationships between instrumentation and genre, e.g. the strong dependence between organ and pop/jazz in IRMAS, and in this specific case an argument could be made that the classification task is closer to transfer learning than domain adaptation.

A notable limitation of the presented experiments is the small amount of functional data. Although OpenMIC dataset is relatively large with 14915 samples for training, only a small portion is actually used in the binary classification of each instrument. After equalizing the number of samples per class in both datasets, there are only 288, 221, 177, 578, 290, 551, 476, 427, 385, and 358 samples for the 10 instrument classes in the binary classification. Most classes have number of samples less than the dimension of OpenL3 (512) and all of them are below that of YAMNet (1024).

6. CONCLUSION

The method proposed in this work addresses one specific form of bias that can arise in transfer learning scenarios. Correctly applying this method requires identifying subsets of data that should be treated equivalently, i.e., be indistinguishable under the chosen representation. We stress that this notion of equivalence ultimately depends on the choice of the downstream task, and caution should be exercised when identifying populations to treat as equivalent. For the case study presented here—domain adaptation and instrument recognition—we argue that the downstream task ought to be generally independent of the source domain, though we recognize that this will not always be true in practice. We therefore urge practitioners to critically investigate all assumptions of equivalence and independence when applying bias correction methods.

7. ACKNOWLEDGMENTS

This work was partly funded by the European Union (ERC, HI-Audio, 101052978). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them. We thank Michel Olvera for the discussions on domain adaptation.

8. REFERENCES

[1] I. Garrido-Muñoz, A. Montejo-Ráez, F. Martínez-Santiago, and L. A. Ureña-López, "A survey on bias in deep NLP," *Applied Sciences*, vol. 11, no. 7, p. 3184, 2021.

[2] E. Ntoutsi, P. Fafalios, U. Gadiraju, V. Iosifidis, W. Nejdl, M.-E. Vidal, S. Ruggieri, F. Turini, S. Papadopoulos, E. Krasanakis *et al.*, "Bias in data-driven artificial intelligence systems—an introductory survey," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 10, no. 3, p. e1356, 2020.

[3] N. Garg, L. Schiebinger, D. Jurafsky, and J. Zou, "Word embeddings quantify 100 years of gender and ethnic stereotypes," *Proceedings of the National Academy of Sciences*, vol. 115, no. 16, pp. E3635–E3644, 2018.

[4] M. G. Haselton, D. Nettle, and D. R. Murray, "The evolution of cognitive bias," *The handbook of evolutionary psychology*, pp. 1–20, 2015.

[5] T. Tommasi, N. Patricia, B. Caputo, and T. Tuytelaars, "A deeper look at dataset bias," in *Domain adaptation in computer vision applications*. Springer, 2017, pp. 37–55.

[6] K. Choi, G. Fazekas, M. B. Sandler, and K. Cho, "Transfer learning for music classification and regression tasks," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2017, pp. 141–149.

[7] A. Cramer, H. Wu, J. Salamon, and J. Bello, "Look, listen, and learn more: Design choices for deep audio embeddings," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 3852–3856.

[8] A. Jansen, J. F. Gemmeke, D. P. Ellis, X. Liu, W. Lawrence, and D. Freedman, "Large-scale audio event discovery in one million YouTube videos," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 786–790.

[9] T. Bolukbasi, K.-W. Chang, J. Y. Zou, V. Saligrama, and A. T. Kalai, "Man is to computer programmer as woman is to homemaker? debiasing word embeddings," *Advances in neural information processing systems*, vol. 29, 2016.

[10] 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.

[11] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *The journal of machine learning research*, vol. 17, no. 1, pp. 2096–2030, 2016.

[12] A. Bailey and M. D. Plumbley, "Gender bias in depression detection using audio features," in *Proceedings of the IEEE European Signal Processing Conference (EUSIPCO)*, 2021, pp. 596–600.

[13] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, "A survey on bias and fairness in machine learning," *ACM Computing Surveys (CSUR)*, vol. 54, no. 6, pp. 1–35, 2021.

[14] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira, "Analysis of representations for domain adaptation," *Advances in neural information processing systems*, vol. 19, 2006.

[15] J. J. Bosch, J. Janer, F. Fuhrmann, and P. Herrera, "A comparison of sound segregation techniques for predominant instrument recognition in musical audio signals." in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2012, pp. 559–564.

[16] E. Humphrey, S. Durand, and B. McFee, "Openmic-2018: An open data-set for multiple instrument recognition." in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2018, pp. 438–444.

[17] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, *The elements of statistical learning: data mining, inference, and prediction*. Springer, 2009.

[18] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," *Advances in neural information processing systems*, vol. 20, 2007.

[19] 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 E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[20] K. Azizzadenesheli, A. Liu, F. Yang, and A. Anandkumar, "Regularized learning for domain adaptation under label shifts," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.

COLLABORATIVE SONG DATASET (COSOD): AN ANNOTATED DATASET OF MULTI-ARTIST COLLABORATIONS IN POPULAR MUSIC

Michèle Duguay[1] **Kate Mancey**[1] **Johanna Devaney**[2]

[1] Department of Music, Harvard University
[2] Brooklyn College and Graduate Center, City University of New York
mduguay@fas.harvard.edu, johanna.devaney@brooklyn.cuny.edu

ABSTRACT

The Collaborative Song Dataset (CoSoD) is a corpus of 331 multi-artist collaborations from the 2010–2019 *Billboard* "Hot 100" year-end charts. The corpus is annotated with formal sections, aspects of vocal production (including reverberation, layering, panning, and gender of the performers), and relevant metadata. CoSoD complements other popular music datasets by focusing exclusively on musical collaborations between independent acts. In addition to facilitating the study of song form and vocal production, CoSoD allows for the in-depth study of gender as it relates to various timbral, pitch, and formal parameters in musical collaborations. In this paper, we detail the contents of the dataset and outline the annotation process. We also present an experiment using CoSoD that examines how the use of reverberation, layering, and panning are related to the gender of the artist. In this experiment, we find that men's voices are on average treated with less reverberation and occupy a more narrow position in the stereo mix than women's voices.

1. INTRODUCTION

As far back as the 1960s, *Billboard* charts have featured collaborations between independent acts. In recent years, however, the number of songs featuring a collaboration between artists has skyrocketed [1]. Part of this is due to the rising popularity of hip-hop in the 1980s, in which collaboration between different artists is a fixture. The 1986 version of "Walk This Way" by Aerosmith and Run DMC is an oft-cited example of such a collaboration. As Rose notes, the success of a collaboration between a hip-hop group (Run DMC) and a rock group (Aerosmith) "brought [hip-hop's] strategies of intertextuality into the commercial spotlight" [2, p. 51–52]. The 1990 success of "She Ain't Worth It" by Glenn Medeiros ft. Bobby Brown marked the first time a sung and rapped collaboration reached #1 on *Billboard*'s "Hot 100." Molanphy notes that during this period, multi-artist collaborations crystallized into two different frameworks: the "featured bridge rapper," and the "featured hook singer" [3]. Subsequently, tracks with one or more guest artist(s) have become a mainstay on the charts.

By 2021, over a third (39%) of the songs in *Billboard*'s "Hot 100" year-end chart credited more than one artist. Consider for instance "Save Your Tears," by singers The Weeknd & Ariana Grande, which occupied second place on the chart. A solo version of the song originally appeared on The Weeknd's album *After Hours* (2020). While this version achieved commercial success, the remix with Ariana Grande became a #1 single on the *Billboard* "Top 100" in May 2021 and became the longest-charting collaboration in *Billboard* "Hot 100" history. In the remix, Grande performs approximately half of the vocals, transforming the solo song into a dialogue between two characters. The collaboration between the two artists is responsible for the popularity of the remix, inviting both Grande's and The Weeknd's fans to stream, buy, and otherwise engage with the song. Several musicological studies have examined this relationship between collaborative songs and commercial success [4–6]. Other work has provided in-depth explorations of the musical characteristics of collaborative songs, with a particular focus on hip-hop [7–9].

Given the popularity of multi-artist collaborations, a more systematic exploration of their musical features is warranted. In this paper, we introduce the Collaborative Song Dataset (CoSoD), an annotated dataset that facilitates the study of various musical features in multi-artist collaborations. CoSoD provides metadata and analytical data for 331 multi-artist collaborations appearing on the *Billboard* "Hot 100" year-end charts between 2010 and 2019. The dataset also provides timed annotations on the song's formal structure, artists' gender, vocal delivery and pitch, and vocal production (reverberation, panning, and layering). As detailed in Section 2, the range of features included in the dataset makes it more broadly applicable for MIR research tasks. These include structural segmentation, vocal mixing, automatic music production, and examinations of gender in popular music. After outlining the contents of the dataset and the annotation methodology in Section 3, we present an experiment in Section 4 that examines the relationship between vocal production parameters and the gender of the performer in a subset of CoSoD.

© M. Duguay, K. Mancey, and J. Devaney. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** M. Duguay, K. Mancey, and J. Devaney, "Collaborative Song Dataset (CoSoD): An annotated dataset of multi-artist collaborations in popular music", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

2. RELATED WORK

CoSoD complements the growing list of annotated datasets that provide information on song structure in various popular music genres, e.g., [10–17], and is the first dataset to exclusively contain data on collaborative songs between independent acts. It can thus be used for training and evaluating structural segmentation tasks and for studying the specific structural characteristics of collaborative songs. CoSoD also complements existing datasets for multi-track mixing/analysis [18–23] and vocal analysis [24–26] by providing analytical annotations on the treatment of the voice in a mix.

In recent years, several studies have proposed tools and methods to automate the mixing of multi-track recordings [27, 28]. Such automatic production methods have various artistic and creative applications. One framework has been suggested to remix early jazz recordings, which are preprocessed using source separation then remixed with automatic production tools [29]. [30] proposes a prototype for an automatic DJ mixing system allowing for cross-fading via beat and tempo adjustment between songs. Studies on automatic mixing can be enhanced by knowledge of common mixing practices for specific instruments or sound sources. For instance, one study uses mixing practices that are consistent between mixing engineers to create a model that automatically mixes multiple drum tracks [31]. By focusing on vocals, which are a salient component of the mix in popular music [32], CoSoD provides a complementary approach to these studies on automated production. By providing annotations based on close listening of specific vocal mixing parameters in the different formal sections of a song, the dataset allows for the identification of trends in panning, layering, and use of artificial reverberation as they are applied to vocals in commercially successful post-2010 popular music. It enables the direct comparison of how various mixing parameters are applied to individual artists' voices within and across songs. In addition to facilitating the modeling of voice mixing, CoSoD also allows musicologists to ask questions about the way different voice types and individuals are mixed.

Finally, CoSoD facilitates the study of the relationship between gender and popular music. A number of previous studies have examined music programming and streaming services, exploring for instance how listeners tend to stream male artists more than women and mixed-gender groups [33]. Watson discusses gender inequality and low programming of women's music in country music radio [34]. Other work addresses how a listener's declared gender impacts automatic music recommendation [35] and musical preferences [36]. Additionally, various studies have addressed race and gender, along with sexist and racist discourses and practices, as they impact the music industry in general and the *Billboard* charts in particular [37–43]. By providing data on musical features, gender, and the role of these parameters within the formal structure of a song, CoSoD offers a new and complementary angle for the study of gender as it directly relates to the musical content of post-2010 popular collaborations.

3. COLLABORATIVE SONG DATASET (COSOD)

CoSoD [1] consists of metadata and analytical data of a 331-song corpus comprising all multi-artist collaborations on the *Billboard* "Hot 100" year-end charts published between 2010 and 2019. Each song in the dataset is associated with two CSV files: one for metadata and one for analytical data. We assembled the corpus by identifying every song on the charts that featured collaborations between two or more artists who usually perform independently from one another.

3.1 Annotation of Musical Features

The following analytical data is provided for each song in the dataset:

1. **Index number:** 1 to 33
2. **Time stamps:** In seconds (start of new section)
3. **Formal section label:** *Introduction, Verse, Pre-chorus, Chorus, Hook, Dance Chorus [44], Link, Post-chorus, Bridge, Outro, Refrain* or *Other*
4. **Name of artist(s):** Full name of the artist performing in each section. If all artists credited on the *Billboard* listing perform in a section, the label *both* or *all* is used.

Songs were assigned at random to one of two annotators, who generated time stamps at the onset of each formal section with Sonic Visualiser. [2] The annotators provided formal labels according to their analysis of the song. In case of ambiguity in the formal sections, both annotators discussed the analysis and agreed upon an interpretation.

For each formal section performed by *one artist only*, the following analytical data on the voice is provided:

1. **Gender of artist:** *M* (Man), *W* (Woman), *NB* (Non-binary)
2. **Function of artist:** *Feat* (Featured artist), *Main* (Main artist), *Neither*, *Uncredited*
3. **Style of vocal delivery:** *R* (Rapped vocals), *S* (Sung vocals), *Spoken*
4. **Minimum pitch value:** In Hz
5. **First quartile pitch value:** In Hz
6. **Median pitch value:** In Hz
7. **Third quartile pitch value:** In Hz
8. **Maximum pitch value:** In Hz
9. **Environment value:** On a scale of E1 to E5
10. **Layering value:** On a scale of L1 to L5
11. **Width (panning) value:** On a scale of W1 to W5

[1] https://github.com/duguay-michele/CoSoD

[2] The first annotator (first author) has a doctorate in music theory, while the second (second author) is a doctoral candidate in the same field.

The annotators determined the name of the artist(s) performing in each section by ear, and using song lyric website Genius.com to validate their hearing. In cases where an artist only provides minimal background vocals (a few words) in a particular formal section, their name is not included. One annotator then provided analytical data on each formal section performed by one artist only. Data on gender was gathered from media interviews and social media statements from the artists, and matches the artist's gender identity at the time of the dataset creation. This methodology yielded three categories: man, non-binary, and woman. We understand these labels as umbrella terms that encompass a variety of lived experiences that intersect with race, sexuality, and other power structures. The style of vocal delivery was determined by ear. The distinction between rapping and singing is porous, with many vocalists adopting ambiguous modes of vocal delivery. We consider any formal section containing a melodic line performed with sustained pitches as sung.

The pitch data was obtained by first isolating the vocals from the full mix using Open-Unmix [45] and then running the pYIN Smoothed Pitch Track transform [46] on the isolated vocal file. The minimum, first quartile, median, third quartile, and maximum pitch points in each formal section were calculated and recorded in the dataset. [3]

The Environment, Layering, and Width values were determined by the first annotator to ensure consistency. Rather than attempting to reconstruct the mixing process itself, the annotations for these parameters represent the way a listener might perceive the final mix upon listening to it on stereo speakers. The Environment of a voice is the space in which the voice reverberates. Environment values were determined via an aural analysis of the full track by using the following scale [4]:

E1: The voice's environment sounds flat. There might be minimal ambiance added to the voice, but there is no audible echo or reverberation.

E2: The last word or syllable of most musical phrases is repeated through an echo or reverberation effect.

E3: The vocal line is repeated in one clear layer of echo. This added layer may be dry or slightly reverberant and has a lower amplitude than the main voice.

E4: The main voice is accompanied by a noticeable amount of reverberation. There is no clear echo layer, but rather a sense that the main voice is being reverberated across a large space.

E5: The main voice is accompanied by two or more layers of echo. The echo layers may be noticeably reverberant, similar in amplitude to the main voice, and difficult to differentiate from one another.

[3] The accuracy of the F_0 estimates used to calculate this feature is impacted by the quality of the vocal source separation. A more accurate isolated vocal file would allow for more precise pitch data. Additionally, since pYIN Smoothed Pitch Track can only track a single melodic line, the accuracy of the pitch data is lessened in sections that feature multiple vocal layers with different pitch content.

[4] The scales were initially published in [9].

The Layering of a voice refers to the additional vocal tracks that are dubbed over a single voice. Layering values were determined via an aural analysis of the full track by using the following scale:

L1: The voice is presented as solo. Occasionally, a few words may be doubled with another vocal track for emphasis. Double-tracking is often used in the mixing process to create a fuller sound, with a final result sounding like a single vocal layer. Such cases fall into this category.

L2: The voice is presented as solo, but additional vocal layers are added at the end of musical phrases for emphasis.

L3: The main voice is accompanied by one or two layers. Layers might provide minimal harmonies or double the main voice. The layers have a noticeably lower amplitude than the main voice.

L4: The main voice is accompanied by two or more layers. These layers are close copies of the main voice, sharing the same pitch and similar amplitude.

L5: The main voice is accompanied by two or more layers. These layers add harmonies to the main voice, creating a thick and multi-voiced texture.

The Width of a voice refers to the breadth it occupies on the stereo stage. The Width was analyzed aurally with the aid of panning visualisation tool MarPanning [47]. The annotator simultaneously listened to the isolated vocal audio and observed the MarPanning visualization generated from the isolated vocals to determine the Width value. Since Open-Unmix occasionally omits reverberated components of the voice from the isolated file, the analyst then listened to the full track to confirm the Width value. Width values were determined according to the following scale:

W1: The voice occupies a narrow position in the center of the stereo stage.

W2: The voice occupies a slightly more diffuse position in the center of the stereo stage.

W3: The main voice occupies a narrow position in the center of the stereo stage, but some of its components (echo, reverberation, and/or additional vocal tracks) are panned toward the sides. These wider components have a lower amplitude than the main voice.

W4: The main voice occupies a slightly more diffuse position in the center of the stereo stage, and some of its components (echo, reverberation, and/or additional vocal tracks) are panned toward the sides. These wider components have a lower amplitude than the main voice.

W5: The main voice and its associated components (echo, reverberation, and/or additional vocal tracks) are panned across the stereo stage. All components have a similar amplitude.

3.2 Metadata

The following metadata is provided for each song in the dataset:

1. **Index number:** From 1 to 331
2. **Year of first appearance on *Billboard* "Hot 100" year-end charts**
3. **Chart position:** As it appears on the *Billboard* "Hot 100" year-end charts
4. **Song title:** As it appears on the *Billboard* "Hot 100" year-end charts
5. **Name of artists:** As it appears on the *Billboard* "Hot 100" year-end charts
6. **Collaboration type:**
   - *Lead/featured:* Collab. with lead artist(s) and featured artist(s)
   - *No lead/featured:* Collab. with no determined lead
   - *DJ/vocals:* Collab. between a DJ and vocalist(s)
7. **Gender of artists:**
   - *Men:* Collab. between two or more men
   - *Women:* Collab. between two or more women
   - *Mixed:* Collab. between two or more artists of different genders
8. **Collaboration type + gender:**
   - *Collab M:* Collab. between men, no determined lead
   - *Collab M and W:* Collab. between men and women, no determined lead
   - *Collab NB and W:* Collab. betwen women and non-binary artists, no determined lead
   - *Collab W:* Collab. between women, no determined lead
   - *DJ with M:* Collab. between male DJ and male vocalist
   - *DJ with Mix:* Collab. between male DJ and mixed-gender vocalists
   - *DJ with NB:* Collab. between male DJ and non-binary vocalist
   - *DJ with W:* Collab. between male DJ and female vocalist
   - *M ft. M:* Men featuring men
   - *M ft. W:* Men featuring non-binary artist(s)
   - *W ft. M:* Women featuring men
   - *W ft. W:* Women featuring women
9. **MusicBrainz URL:** Link to the song on open music encyclopedia MusicBrainz

Each song in the dataset is labeled with an index number from 1 to 331. Songs are numbered in reverse chronological order, beginning with the 2019 charts and ending with 2010. One annotator obtained the metadata on year, chart position, title, and artists from the information available on the *Billboard* charts. Within years, songs are organized according to their position on the chart, from highest to lowest. Some songs appear on the charts two years in a row. In such cases, we only include the data for the earliest appearance.

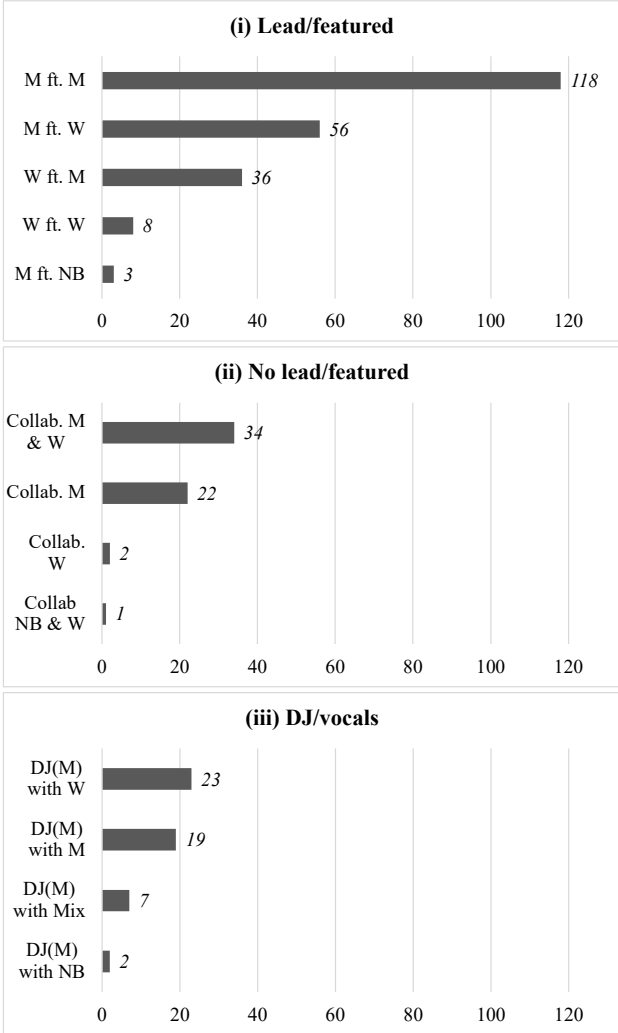


Figure 1. Summary of the gender distribution across different types of multi-artist collaborations. Subplot (i) shows gender counts for collaborations with lead and featured artists, subplot (ii) shows collaborations with no determined lead or featured artist, and subplot (iii) shows collaborations between DJs and vocalist(s).

3.3 Corpus Statistics

The dataset can be divided into three categories (shown in Figure 1): (i) collaborations between the lead artist(s) and featured artist(s), which account for 221, or 66.7% of the tracks, (ii) collaborations with no determined lead or featured artist, which account for 59, or 17.8%, of the tracks, and (iii) collaborations between a DJ and a vocalist, which account for 51, or 15.4% of the tracks. In category (i), the lead artist usually performs the majority of vocals. For example, in "No Limit" (2018) by G-Eazy ft. A$AP Rocky & Cardi B, G-Eazy performs most of the vocals. A$AP

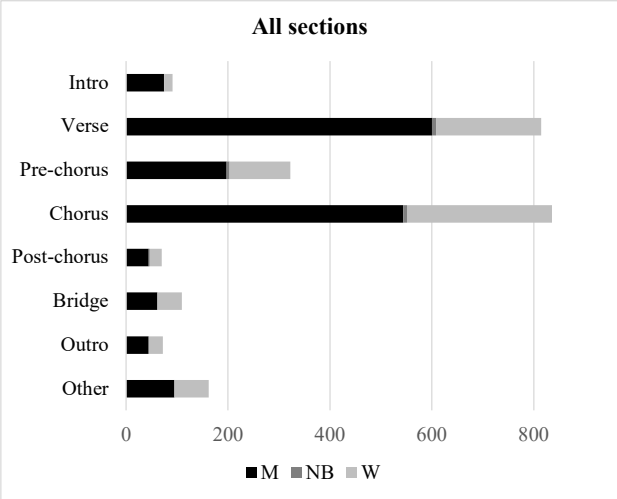


Figure 2. Number of formal sections performed by a single artist (main, featured, or neither), categorized according to formal section type

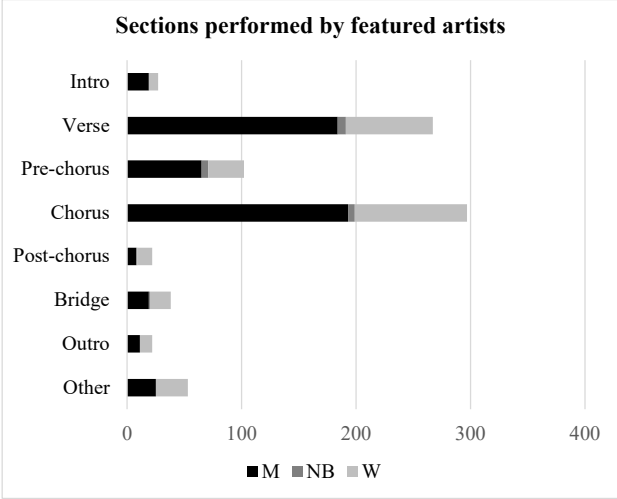


Figure 3. Number of formal sections performed by featured artists, categorized according to formal section type

Rocky accompanies him in the chorus and Cardi B raps the second verse. In category (ii), the performance of the vocals is often more equally distributed. Such collaborations are often billed as "duets," and the artists' names are separated by a "+", a "&", or a comma on the *Billboard* charts. For example,"Something' Bad" (2014) is labeled as a "Miranda Lambert Duet With Carrie Underwood." Both vocalists perform approximately equal portions of the song. In category (iii), the DJ does not provide vocals. In "Sweet Nothing" (2012), for instance, only the featured Florence Welch sings. The voice of DJ Calvin Harris is not heard.

Mixed-gender collaborations (including any combination of non-binary, women, and men artists) frequently appear on the Billboard charts and account for 162, or 49%, of the tracks in the dataset. Collaborations between two or more men account for 159 tracks, or 48% of the dataset. Finally, collaborations between women account for 10, or 3%, of the tracks. In six of the ten years under study–2011, 2012, 2015, 2017, 2018, and 2019–no collaborations between women reached the *Billboard* "Hot 100" year-end chart. Conversely, songs with two or more male vocalists were a consistent fixture on the charts. Mixed-gender collaborations, with any combination of men, women, and non-binary artists within the same track, also frequently appear on the charts.

Figure 2 shows the number and type of sections performed by individual artists in the corpus, categorized according to gender. This figure includes identical sections (such as choruses) that are repeated within a song. Sections in which more than one artist performs are not included. More sections are performed by men than by women and non-binary artists, which is to be expected given the over-representation of men in the dataset as a whole (Figure 1). Figure 3 displays the number and type of sections performed by *featured* artists only.

4. EXPERIMENT: VOCAL PRODUCTION FEATURES AND GENDER

This section examines the relationship between the gender of an artist and the treatment of their voice, as characterized by three of the annotated musical features in the dataset: Environment, Layering, and Width. For the purposes of statistical power in the experiment, only songs with men and/or women artists were included. We only included tracks that contained verse and chorus sections to remove section types that occur in only a few tracks. In order to avoid over-representations of tracks with repeated sections (i.e., several instances of the same chorus), we sampled the first verse and chorus performed by a single artist from each track. [5] This method resulted in the inclusion of two sections from 287 of the 331 dataset tracks in the experiment.

We analyzed the data with three separate logistic regressions–one for each feature–using the statsmodels package in Python. We encoded the different levels of the parameter scales (defined in Section 3.1) with one-hot encoding in order to allow us to examine whether there is a correspondence between specific parameter scale levels and gender.

Of the three logistic regressions, Environment (R^2_{McFadden} (4, N = 574) = 0.028, $p < 0.0001$) and Width (R^2_{McFadden} (4, N = 574) = 0.035, $p < 0.0001$) were statistically significant, while Layering (R^2_{McFadden} (4, N = 574) = 0.0036, p = 0.64) was not. The McFadden R^2 values for both Environment and Width were very low. This was not surprising since we did not anticipate that these features, particularly in isolation, would be explanatory. We were instead interested in exploring whether there is a significance between these features with respect to the man/woman gender binary in these collaborations.

For Environment, there were significant effects ($p <$

[5] If the first verse of a song was performed by two artists simultaneously, while the second verse was only performed by one, we sampled the second verse.

0.0001) for E1 (β=-1.18, 95%CI [-1.49, -0.87]), E2 (β=-1.12, 95%CI [-1.56, -0.69]), and E3 (β=–0.78, 95%CI [-1.14, -0.42). There was a significant negative effect for the lower/mid-level environment values and gender, meaning that men's voices are more likely to be set in less reverberant spaces than women's voices. For Width, there were significant effects at all of the levels: W1 (β=-1.84, 95%CI [-2.50, -1.17]), W2 (β=-1.58, 95%CI [-2.39, -0.77]), W3 (β=–1.13, 95%CI [-1.51, -0.75]), W4 (β=-0.47, 95%CI [-0.77, -0.17), and W5 (β=-0.60, 95%CI [-0.95, -0.25]).

The Width results are harder to interpret than the Environment ones because the coefficient values are smaller and all negative. This is likely due to the imbalance between men and women in featured artist roles, both in the dataset (see Figure 1) overall and in the sample used in this experiment (404 of the included sections featured men while only of 170 featured women). However, the overall trend is similar to the one in the Environment experiment: lower-level values are more common for men than women. Men's voices are more likely to occupy a narrow, centered position on the stereo stage, while women's voices are more likely to occupy a wider space. These results were expected given that high Environment values tend to be associated with high Width values, as the reverberated components of a voice are generally panned across the stereo stage.

The lack of significant results for Layering indicates that there are no differences in the ways in which this parameter is applied to men's and women's voices. Since textural variation (such as the addition of vocal layers) is a standard feature of verse-chorus form, it is possible that Layering is linked to the type of formal section rather than to the gender of the vocalist. The significant results for the Environment and Width parameters can be interpreted in light of Brøvig-Hanssen's and Danielsen's work on technological mediation [48]. The authors establish a distinction between transparent and opaque technological mediation in recorded music. Transparent mediation, on one hand, is meant to create a recorded product that sounds natural and unaltered. Low Environment and Width values, for instance, are closer to transparent mediation because they sound closer to a real-life performance that is unmediated with artificial reverb or panning. Opaque mediation, on the other hand, highlights the use of technology by making it obvious to the listener. High Width and Environment values, with their clearly audible artificial reverberation and wide panning, are examples of opaque mediation. The results of the experiment therefore suggest that men's voices are more likely to be mixed to sound "transparent" and natural while women's voices are more likely to be mixed to sound "opaque" and technologically mediated.

Overall, this experiment demonstrates that within verse and chorus sections in CoSoD, there is a significant difference between the treatment of men's and women's vocals in terms of Environment and Width. This suggests that some mixing parameters contribute to the sonic differentiation of men's and women's voices in popular music.

5. CONCLUSION

CoSoD is a 331-song corpus of all multi-artist collaborations for faciliating appearing on the 2010–2019 *Billboard* "Hot 100" charts. Each song in the dataset is annotated with metadata, formal sections, and aspects of vocal production (including reverberation, layering, panning, and gender of the artists). As outlined in Section 2, CoSoD has several implications for MIR research. It provides annotated data for structural segmentation tasks and a listener-centered perspective on vocal mixing that could be useful for automatic music mixing tasks. The dataset could also be used to determine how these parameters interact with song form. Further study could also examine the relationship between the vocal range of an artist in a given section, their type of vocal delivery (rapped, spoken, or sung), and mixing parameters. Finally, the dataset also allows for the examination of the ways in which Environment, Layering, and Width values tend to be grouped together to create specific vocal production effects.

The dataset also facilitates musicological study of multi-artist collaborations post-2010 and gender norms. The experiment in Section 4 demonstrates this, as its results suggest that, for the chorus and verse data sampled from 287 songs in the dataset, men's voices are more likely to be narrow and less reverberated than women's. Opportunities for future research include examining whether there is a significant difference in the way Environment, Width, Layering, or other parameters are applied to women's and men's voices *within* collaborations that feature mixed- and same-gender vocalists. In other future work, we plan on expanding the annotations in the dataset with time-aligned lyrics, harmonic analyses, and additional performance data for the voice extracted using AMPACT [49, 50]. These annotations will include both spectral features and semantic descriptors, and the data will be encoded in relation to vocal-line transcriptions, where possible [51]. We also plan on providing annotations on vocal production parameters in sections performed by multiple artists and examining how vocal production parameters correlate with mixing parameters such as panning.

Finally, while our dataset focuses on gender, we are also interested in encoding other aspects of identity, such as race, in order to provide an intersectional perspective on artists' identities. However, categorizing artists according to race proves to be more problematic than gender. Matthew D. Morrison writes that "white (and other non-black) people freely express themselves through the consumption and performance of commodified black aesthetics without carrying the burden of being black under white supremacist structures" [52, p. 791]. In other words, white and non-Black artists–such as rappers Iggy Azalea and G-Eazy, or singer Bruno Mars–often assume particular sonic characteristics that implicitly associate them with commodified notion of Blackness. By categorizing all white artists together, for instance, we would ignore this phenomenon and the way it is sonically realized. Further work needs to be done to understand how to best expand on CoSoD, or datasets in general, to account for this dynamic.

6. REFERENCES

[1] Anonymous, "In popular music, collaborations rock," *The Economist*, February 2018. [Online]. Available: https://www.economist.com/business/2018/02/03/in-popular-music-collaborations-rock

[2] T. Rose, *Black Noise: Rap Music and Black Culture in Contemporary America*. Hanover, NH: Wesleyan University Press, 1994.

[3] C. Molanphy, "Feat. don't fail me now: The rise of the featured rapper in pop music," *Slate*, July 2015.

[4] A. Ordanini, J. C. Nunes, and A. Nanni, "The featuring phenomenon in music: How combining artists of different genres increases a song's popularity," *Marketing Letters*, vol. 29, no. 4, pp. 485–499, Dec 2018.

[5] M. Silva and M. Moro, "Causality analysis between collaboration profiles and musical success," in *Proceedings of the 25th Brazillian Symposium on Multimedia and the Web*, 10 2019, pp. 369–376.

[6] G. P. Oliveira, M. O. Silva, D. B. Seufitelli, A. Lacerda, and M. M. Moro, "Detecting collaboration profiles in success-based music genre networks," in *Proceedings of the 21st International Society for Music Information Retrieval Conference*, 2020, pp. 726–732.

[7] R. Komaniecki, "Analyzing collaborative flow in rap music," *Music Theory Online*, vol. 23, no. 4, 12 2017.

[8] B. Duinker, "Song form and the mainstreaming of hip-hop music," *Current Musicology*, vol. 107, pp. 93–135, 1 2020.

[9] M. Duguay, "Analyzing vocal placement in recorded virtual space," *Music Theory Online*, vol. 28, no. 4, 2022.

[10] A. Berenzweig, B. Logan, D. Ellis, B. Whitman, and C. A, "A large-scale evaluation of acoustic and subjective music similarity measures," *Computer Music Journal*, vol. 28, 11 2003.

[11] C. Harte, "Towards automatic extraction of harmony information from music signals," Ph.D. dissertation, Queen Mary, University of London, august 2010.

[12] J. Burgoyne, J. Wild, and I. Fujinaga, "An expert ground truth set for audio chord recognition and music analysis." 01 2011, pp. 633–638.

[13] J. Smith, J. Burgoyne, I. Fujinaga, D. De Roure, and J. Downie, "Design and creation of a large-scale database of structural annotations," in *Proceedings of the 12th International Society for Music Information Retrieval Conference*, Miami, FL, USA, 01 2011, pp. 555–560.

[14] T. Bertin-Mahieux, D. P. Ellis, B. Whitman, and P. Lamere, "The million song dataset," in *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*, 2011.

[15] T. de Clerq and D. Temperley, "A corpus analysis of rock harmony," *Popular Music*, vol. 30, no. 1, pp. 47–70, 2011.

[16] F. Bimbot, G. Sargent, E. Deruty, C. Guichaoua, and E. Vincent, "Semiotic description of music structure: an introduction to the quaero/metiss structural annotations," in *Proceedings of the AES International Conference*, London, UK, 01 2014.

[17] 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*, Delft, The Netherlands, 2019, pp. 565–572.

[18] 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. 310–19, 2010.

[19] 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 International Society for Music Information Retrieval Conference*, Taipei, Taiwan, 10 2014.

[20] T.-S. Chan, T.-C. Yeh, Z.-C. Fan, H.-W. Chen, L. Su, Y.-H. Yang, and R. Jang, "Vocal activity informed singing voice separation with the ikala dataset," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brisbane, Australia, 2015, pp. 718–722.

[21] R. Bittner, J. Wilkins, H. Yip, and J. 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, NY, USA, 2016.

[22] B. D. Man and J. D. Reiss, "The mix evaluation dataset," in *Proceedings of the 20th International Conference on Digital Audio Effects*, Edinburgh, UK, 9 2017, pp. 436–442.

[23] Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimilakis, and R. Bittner, "The MUSDB18 corpus for music separation," Dec. 2017.

[24] R. Gong, R. C. Repetto, and X. Serra, "Creating an a cappella singing audio dataset for automatic jingju singing evaluation research," in *Proceedings of the 4th International Workshop on Digital Libraries for Musicology*, Shanghai, China, 2017, pp. 37–40.

[25] C.-i. Wang and G. Tzanetakis, "Singing style investigation by residual siamese convolutional neural networks," in *2018 IEEE International Conference on*

Acoustics, Speech and Signal Processing (ICASSP), Calgary, Canada, 2018, pp. 116–120.

[26] 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.

[27] B. D. Man and J. D. Reiss, "Ten years of automatic mixing," in *Proceedings of the 3rd Workshop on Intelligent Music Production*, Salford, UK, 2017.

[28] 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 *Proceedings of the 23rd International Society for Music Information Retrieval Conference*, 2022.

[29] D. Matz, E. Cano, and J. Abeßer, "New sonorities for early jazz recordings using sound source separation and automatic mixing tools," in *Proceedings of the 16th International Society for Music Information Retrieval Conference*, Málaga, Spain, 2015, pp. 749–755.

[30] 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*, Kobe, Japan, 01 2009, pp. 135–140.

[31] J. J. Scott and Y. E. Kim, "Instrument identification informed multi-track mixing," in *Proceedings of the 14th International Society for Music Information Retrieval Conference*, Curitiba, Brazil, 2013, pp. 305–310.

[32] A. L. Knoll and K. Siedenburg, "The optimal mix? presentation order affects preference ratings of vocal amplitude levels in popular music," *Music & Science*, vol. 5, pp. 1–12, 12 2022.

[33] A. Epps-Darling, H. Cramer, and R. Takeo Bouyer, "Artist gender representation in music streaming," in *Proceedings of the 21st International Society for Music Information Retrieval Conference*, 10 2020, pp. 248–254.

[34] J. Watson, "Programming inequality: Gender representation on canadian country radio (2005-2019)," in *Proceedings of the 21st International Society for Music Information Retrieval Conference*, 2020, pp. 392–399.

[35] G. Vigliensoni and I. Fujinaga, "Automatic music recommendation systems: Do demographic, profiling, and contextual features improve their performance?" in *Proceedings of the 17th International Society for Music Information Retrieval Conference*, New York, NY, USA, 2016, pp. 94–100.

[36] A. Laplante, "Improving music recommender systems: What can we learn from research on music tastes?" in *Proceedings of the 15th International Society for Music Information Retrieval Conference*, Taipei, Taiwan, 2014, pp. 451–456.

[37] C. L. Keyes, "Empowering self, making choices, creating spaces: Black female identity via rap music performance," *The Journal of American Folklore*, vol. 113, no. 449, pp. 255–69, 2000.

[38] T. J. Dowd and M. Blyler, "Charting race: The success of black performers in the mainstream recording market, 1940 to 1990," *Poetics*, vol. 30, pp. 87–110, 2002.

[39] R. N. Bradley, *Barbz and Kings: Explorations of Gender and Sexuality in Hip Hop*. Cambridge, England: Cambridge University Press, 2015, pp. 181–191.

[40] M. Lafrance, C. Scheibling, L. Burns, and J. Durr, "Race, gender, and the billboard top 40 charts between 1997 and 2007," *Popular Music and Society*, vol. 41, no. 5, pp. 522–538, 2018.

[41] K. J. Lieb, *Gender, Branding, and the Modern Music Industry: The Social Construction of Female Popular Music Stars*, 2nd ed. New York, NY: Routledge, 2018.

[42] J. E. Watson, "Gender on the billboard hot country songs chart, 1996–2016," *Popular Music and Society*, vol. 42, no. 5, pp. 538–60, 2002.

[43] C. Bauer and J. Devaney, "Constructing gender in audio: exploring how the curation of the voice in music and speech influences our conception of gender identity," in *Mediale Stimmentwürfe: perspectives of media voice designs*, ser. Schriftenreihe zur digitalen Gesellschaft NRW, M. Erbe, A. Riffi, and W. Zielinski, Eds. Munich, Germany: kopaed Verlag, 2022, vol. 7, pp. 83–100.

[44] A. Barna, "The dance chorus in recent top-40 music," *SMT-V*, vol. 6, no. 4, June 2020. [Online]. Available: http://doi.org/10.30535/smtv.6.4

[45] F.-R. Stöter, S. Uhlich, A. Liutkus, and Y. Mitsufuji, "Open-unmix - a reference implementation for music source separation," *Journal of Open Source Software*, 2019.

[46] M. Mauch and S. Dixon, "pyin: A fundamental frequency estimator using probabilistic threshold distributions," in *2014 Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Florence, Italy, 2014, pp. 659–663.

[47] K. McNally, G. Tzanetakis, and S. R. Ness, "New tools for use in the musicology of record production," *Unpublished Paper, University of Victoria.*, 2009.

[48] R. Brøvig-Hanssen and A. Danielsen, *The Impact of Digitization on Popular Music Sound*. Cambridge, MA, USA: The MIT Press, 2016.

[49] J. Devaney, M. I. Mandel, and I. Fujinaga, "A study of intonation in three-part singing using the automatic music performance analysis and comparison toolkit (ampact)." in *Proceedings of the 13th International Society for Music Information Retrieval Conference*, 2012, pp. 511–516.

[50] J. Devaney and M. Mandel, "Score-informed estimation of performance parameters from polyphonic audio using ampact," in *Extended abstracts for the Late-Breaking Demo Session of the 17th International Society for Music Information Retrieval Conference*, 2016.

[51] J. Devaney, "Using note-level music encodings to facilitate interdisciplinary research on human engagement with music," *Transactions of the International Society for Music Information Retrieval*, vol. 3, no. 1, 2020.

[52] M. D. Morrison, "Race, blacksound, and the (re)making of musicological discourse," *Journal of the American Musicological Society*, vol. 72, no. 3, pp. 781–823, 12 2019.

HUMAN-AI MUSIC CREATION: UNDERSTANDING THE PERCEPTIONS AND EXPERIENCES OF MUSIC CREATORS FOR ETHICAL AND PRODUCTIVE COLLABORATION

Michele Newman
University of Washington
mmn13@uw.edu

Lidia Morris
University of Washington
ljmorris@uw.edu

Jin Ha Lee
University of Washington
jinhalee@uw.edu

ABSTRACT

Recently, there has been a surge in Artificial Intelligence (AI) tools that allow creators to develop melodies, harmonies, lyrics, and mixes with the touch of a button. The reception of and discussion on the use of these tools - and more broadly, any AI-based art creation tools - tend to be polarizing, with opinions ranging from enthusiasm about their potential to fear about how these tools will impact the livelihood and creativity of human creators. However, a more desirable future path is most likely somewhere in between these two polar opposites where productive and ethical human-AI collaboration could happen through the use of these tools. To explore this possibility, we first need to improve our understanding of how music creators perceive and utilize these types of tools in their creative process. We conducted case studies of a range of music creators to better understand their perception and usage of AI-based music creation tools. Through a thematic analysis of these cases, we identify the opportunities and challenges related to the use of AI for music creation from the perspective of the musicians and discuss the design implications for AI music tools.

1. INTRODUCTION

In the past few years, there has been an increase in the creation of AI tools that support various musical activities. These activities are varied, including music recommendation/organization [1], sound synthesis [2], composition [3–5], and mixing [6, 7]. Current discourse on the use of AI-based tools in music production often presents two polarized perspectives: one that sees AI as an opportunity for innovation and progress [8, 9], while the other views it as a threat to the artistic creativity and livelihood of human creators [10–12]. However, a more nuanced and desirable approach entails a productive and ethical collaboration between humans and AI in the creative process, allowing both human creators and AI tools to create something neither could easily do alone.

© M. Newman, L. Morris, and J.H. Lee. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).
Attribution: 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. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

Discussion around the perception of AI and music creativity tends to be focused on evaluation of the product of the AI [13–15], legal issues [14, 16], or human-computer interaction [17], and not on the implications and connections these factors have on the creative thinking of musicians, though there is growing interest in this domain [18, 19]. Additionally, while there has been discussion within the MIR community around the ethical implications of AI in music creation [18, 20], the experience of using AI to perform songwriting tasks [17], and the perspectives of expert users in creative music information retrieval [21, 22], there is still a need to further understand how creative MIR tasks are impacted by AI tools based on creative context. Even within the ISMIR community, in the last decade, there were fewer than 20 publications that discussed AI music creation tools, and less than half of them considered the creator's perspective before developing the tool.

Musicians engage in creativity in many different ways through generating products such as compositions, analyses, and performances [23]. Our paper aims to address the impact of AI tools on the perception and work of one such domain: composition. Within composition we explore the impact of AI on what Peter R. Webster [23, p. 22] describes as "Creative Thinking," or "the mental processes associated with creative production." We will refer to those who engage in this act of creative thinking in composition as *creators*, their environment/creation goals as *creative context*, and the act of creative thinking as the *creative process*. This paper addresses three research questions: (1) In what way do creators perceive and envision the use of AI tools during their compositional process?, (2) How does their creative context influence their use of AI? and (3) What design implications can we derive to inform the creation of AI tools for music creators?

Our paper extends knowledge about how AI impacts the creative process of musical creators and adds to the discussion of expert users of creative MIR and human-AI collaboration in music creation acts. [19,21,22,24,25] To address these questions we conducted six case studies across a selection of creative contexts and our results are collected in a model that emphasizes the fluidity of roles that AI can play across creative thinking in composition and represents the start of future work aimed at building a dynamic model of human-ai musical creativity.

2. RELATED WORK

The idea of using computational means in composition is not a recent development. Over the course of music history, creators have considered various ways to develop algorithmic procedures to help with their process [26]. Since the early days of the computer, programmers and music creatives alike have utilized their skills to create programs that allowed them to continuing this tradition; creating computer-aided compositions (CAC) and computer-aided algorithmic compositions (CAAC). As a whole, CAC tools require user intervention - correction to misnoted parts, adjustment of autotuned voices, and creators choosing which electronic instruments to employ and when [27]. CAAC tools, unlike CAC tools, are intended to be used to help "make music with minimal human intervention" [28]. Popular examples of CAAC tools include programs such as Opusmodus [29] - a library for real-time computer-aided composition in Max [30] - and more. These tools are extremely distinct in their purpose and use, helping to algorithmically aid creators, with rhythmic trees, polymetric notation, and data visualizations of algorithmically-generated material. When defining algorithmic composition, Pearce et al. [31] state:

> Many who write programs for music composition are motivated by artistic goals: these programs are used to generate novel musical structures, compositional techniques and even genres of music...The composer may use an existing computer program or she may write a program herself: since identical motivations are involved, we count both of these as algorithmic composition. [31, p. 5]

In both cases, the question arises: *What is creativity and what does it mean for computers to be part of the creative process?* This question has been discussed in many ways in multiple fields, as scholars from the humanities [32, 33], the sciences [34–36] , HCI [37, 38] and MIR [21, 22] have speculated for years over how the use of computer systems changes the creation process.

Through their exploration of using AI to co-songwrite, Micchi et al. [17] list two potential ways in which AI tools could assist creators: through (1) automation and (2) AI as suggestion. They note that while AI as automation is more akin to the tasks given to AI outside of the artistic field, the idea of AI as suggesting solutions to compositional tasks, acting as a partner in the process, is unique to the use of AI within creative pursuits. As Tipei et al. [39] discuss in their paper where student composers utilized DISSCO (Digital Instrument for Sound Synthesis and Composition), compositions were still considered by users to be collaborative, as participants were able to add, modify, or reject contributions made by the software and other users. Researchers compared this interaction to the process of collective improvisation, with the software playing a key role as a collaborator and manager in this compositional process - "[the computer/software]...becomes part of the process not only by performing a vast number of operations very quickly, but also as a consequential contributor to the creative effort" [39]. More specifically, this implies that AI simultaneously acts as a collaborator in the process and as a tool, allowing the creator to explore different possibilities of how AI can be applied within their workflow.

AI as Collaborator	Democratization
Meaning of Creativity	Bias in AI
Creative Control	Corporatization of Art
Influence	Knowledge of AI
Mechanism	Creating Opportunities
Old vs. New AI	Sharing of Tools
Types of AI Contributions	Current State of AI

Table 1. Final Codebook for Interviews

3. STUDY DESIGN AND METHODS

We employed a exploratory, multi-subject case study method [40] to examine how creators perceive the use of AI tools within their compositional process. Using multiple-subject case studies allows us to better explore the phenomenon of AI within the compositional process across a variety of contexts in order to build a stronger basis of understanding and is useful for formulating concepts for theory construction [40, 41].

Our case selection strategy was focused on representing diverse cases within the varied creative contexts of both western art music and western popular/commercial music traditions [42]. Our cases included a classical/jazz music composer, a film and video game music composer, an interactive media composer, an electroacoustic composer, a sound artist, and a DJ. Due to the scope of the study and resources, we did not include case studies of programmers, listeners, or creators outside the western context, though these communities will be explored in further studies.

There were a total of six creators, one for each case, all of which were over eighteen years of age and recruited via email. Of the recruited participants, all had heard of AI tools and five worked actively with AI tools within their process. While all creators were actively working within the music field professionally, the film music creator and the intermedia creator were the only ones not affiliated with an academic institution as a student, though both had been trained within western academic music schools. All participants had been composing over five years at the time of the interview.

For each case, we conducted in-depth, semi-structured interviews between 60 and 90 minuets. Interview questions for this study were generated via a review of the existing literature on the use of AI in music composition and production, where we identified relevant themes and topics (e.g., definitions of AI, AI creativity, typical tools in music creation). Topics ranged from participants' experiences with AI-based tools in music production, their perceptions of the advantages and disadvantages of using AI, and their views on the ethical implications of AI in music creation. Both descriptions of the case contexts and interview questions can be found at

the url: `https://github.com/michelenewman/ISMIR23_supplemental_material`.

All interviews were conducted online over Zoom and fully transcribed and edited for clarity. We created the codebook using a mix of the inductive and deductive approach [43]. Initially, two of the authors created the first draft of the codebook using thematic analysis of the transcribed interviews. The codebook was iteratively refined by adjusting and aligning the themes that emerged from the interview data with those from existing literature. Using the final codebook, we first coded the interviews separately on the qualitative coding software ATLAS.ti, then came together to discuss any discrepancy with a goal of reaching an agreement and assigning final codes following the consensus model [44].

4. RESULTS

During analysis, 12 categories emerged which were grouped into two main sets: AI as Collaborator and Democratization of Music Creation. The themes were influenced by the current or lack of use of AI by the creators and the reasoning behind their choices. Themes that arose such as tool sharing are common practice among communities of creators, especially on the internet [45, 46], but within the this study, refers specifically the sharing of AI and ML tools.

Coding the interview transcripts led to the insight that creators had specific creative tasks with which they would or would not feel comfortable utilizing AI tools, as well as parts of the process in which they would consider the use of AI. The most common code within our analysis was "Types of AI Contribution" in which the creators reflected on how they would personally use AI within their own process. This included tasks such as creating repositories (P4) and mastering songs (P6). The least common code was "Sharing of Tools." As a whole, all participants had some knowledge of what AI was, and all but the jazz/classical creator had utilized it in some capacity within their workflow. Three of the creators used also used non-musical AI (such as text-based AI) in their process.

Based on our analysis, we present the ***Human-AI Creative Collaboration Model*** (Figure 1) to represent the use of AI tools throughout the compositional process of music creators situated within the western tradition of composition who may employ computer assisted tools. The model is comprised of three parts: Factors on Influences, AI Roles, and Creation as Process.

4.1 Factors on Influences

The far-most left section of our model represents the various contextual factors that impact creators' perception on where AI should fall within their creative process. These factors are broken into three parts within our model: personal context, social context, and creative goal. While all creation contexts are different, these are the three most common aspects that arose from our analysis.

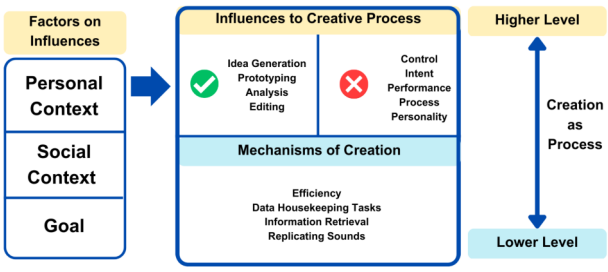


Figure 1. Human-AI Creative Collaboration Model

4.1.1 *Personal Context*

The *Personal Context* is defined by the individual creator's familiarity with their own creative process and their music literacy. When reflecting on whether or not participants thought that an AI tool would be helpful to them, they considered where it would fall into their process and how much control they would be able to maintain over the final product. Because our participants had been composing for over five years, they already had a strong idea of how their process worked and a familiarity with their personal context of musical creation. Creators commented on their desire to have AI tools that are flexible enough to work within their current creative process, are interoperable with existing compositional software, and have clear and concise interfaces to help with facilitating their adoption and use.

For example, the film music creator and DJ who work in more commercial settings, with tighter schedules, mentioned utilizing AI tools that were already integrated within software they used such as Ozone [7] and Logic's Drummer [47]. If integration is not possible, they suggested AI should be designed in a way that it does not interfere with the use of a primary creation tool. The electroacoustic creator, interactive music creator, and sound artist preferred to use older forms of AI due to developer transparency. The participants differentiated older models from the newer models, suggesting that newer models were hidden behind a corporate "black box" in order to work. These creators preferred supervised learning algorithms to unsupervised learning algorithms, so that they could change the open source code and exert more creative control (P5). As they had more experience with AI, they were more open to learning and working with AI tools. All participants also commented that current AI tools were not able to support their process in the ways that they wanted due to the lack of control and low-fidelity outputs.

4.1.2 *Social Context*

Creators also consider *Social Context*; this includes their current community of practice where they often converse and share their art with (other creators, their audience) and their educational and musical upbringing. Many aspects of the act of creativity are tied to the sociocultural aspects of making music [48]. Those whose communities were most open to using AI tools, often in more experimental creative contexts such as in academic experimental com-

munities, were much more willing to engage with the idea of AI in different parts of their process. The jazz/classical composer was adamant that they were very skeptical of AI in part because the community around them was also very skeptical, especially with a strong tradition of composition within the western art context.

Similarly, they decided on what was an appropriate use of AI by comparing the impact the tool had on others. Participants raised the issue of bias in AI and using the creative works of others. All creators noted that most large-scale models rely on Eurocentric training data that may not align with their individual artistic expressions or requirements, feeling that the AI would "flatten" their work with it has "biases and this kind of Eurocentric Westernization of aesthetics" (P5). P4 noted, "There's been a big problem in the past couple of weeks with people coming out talking about how that's not right, to be able to use someone's likeness and their voice however you want." This sentiment highlights the worry that there was oversight in the ways in which these models are being created and distributed, leading to potential harms in taking the intellectual property of others and using it to quickly make financial gains, a sentiment echoed by others in the music industry [15]. Furthermore, participants expressed worry about the impact of such rapidly generated artworks on not only their personal work, but also on the general public's perception of art as a whole.

Participants also talked about how AI tools opened up the potential for more types of creators to get involved in the music creation process. Our DJ participant discussed their use of the AI tool Ozone, which he uses to digitally master his songs. While the tool costs 50 USD to purchase, the participant could use it to process all of his songs and have them match the audio specifications needed to upload to streaming services such as Spotify within seconds, whereas if it was sent to a mastering engineer, each song would cost him hundreds of dollars to master. In the same vein, multiple participants mentioned that one of the potential abilities of AI music tools would be the ways it could potentially allow entry-level musicians to bypass some of the extensive music education they would need before being able to create music, including the cost and time investment of said education. "I think it can really accelerate the learning process, the process of studying music and experiencing all this music that we've documented...I think it can kind of build each person's personal vocabulary of what music is." (P1). These participants qualified their statements by clarifying that this would not mean users should bypass the whole process of learning the art of composition. Rather, the ability to create music without networking and funding as a necessity in the creative process is a kind of "freedom" one participant noted, one which begun with the advent of the personal computer and has only continued to expand as the process is simplified (P2).

4.1.3 Creative Goal

Lastly, *Goal* refers to creators' specific reason for composing (i.e. for a film, for a commission, a performance). The goal can put pressure on creation time, influence the social practices and expectations, and change the personal workflow of the creator.

4.2 Potential AI Roles: Influences and Mechanisms

At the center of the model, we present the different ways that AI could potentially be used in the creation process. Participants expressed an overall positive view towards the potential of AI tools as collaborators. Participants also personified the AI in their process stating it was similar to having a "second person" check over their work or a way to bounce off other ideas with the AI tools. While the list is not exhaustive, these represent the most common tasks that creators in our study talked about. These roles were primarily impacted by the concept of "control" of a creative output across the process. All participants agreed that computational creativity cannot supplant human creativity. While participants recognized that AI can "create something" and output a product that mirrors human creativity, such as P4 stating that they were "...sure AI could create something like a poem, for example, that would be really hard for me to tell if it was from a human or from an AI," they highlighted that AI lacks the deliberate decision-making of human creators, continuing they would have a hard time "emotionally connect[ing] with it."

In the former case, participants discussed engaging in a process of "play" with the AI, which allowed them to explore a variety of prompts and generate a collection of potential options that they could later modify or combine to achieve their artistic goals. P5 noted: "So sometimes when I'm stuck, I like to grab some of the models I pretrained and just ask it something." The creators used the tools to explore, both as a way to spark new ideas and as a way to generate a large repository of content to remix in their own way. Within this process though, the creators emphasized the AI does not make the final choice. The final decision was always made by the creator to maintain their artistic agency.

For all of the participants, within the context of their own compositional process, intention and choice was as important as the creative product. One participant stated, "You can have [AI] generate some sort of electronic music code for you and that sort of just skips for me a whole important step in the process, because in my process of creating live electronic music, there's sort of an interplay between my coding and my writing. I think a lot will be lost if you just take out an entire part of that process" (P3). Another participant remarked, "For me, creativity also involves the decision-making in a big way. And then to determine where to end things. It doesn't seem that my experience with AI so far affords these possibilities" (P6).

Elaborating on this idea of creative control, P6 noted: "I feel that it doesn't sound like me, or especially with music compositions and working with some of these AI that will give you a MIDI file, you know?", implying some kind of loss in the creator's personality in automatically generated music pieces. P1 stated that "AI seems to be something that's designed to do some of that channeling of an idea for

you. It seems like AI is kind of trying to be designed to do the human part of the process." Personality Theory related to intellectual property, put forward by thinkers like Emmanual Kant and George Hegel [49], suggests that a person's personality is incorporated into their creative work during the labor process, and is therefore an essential part of their work. When an AI takes that labor away from the creator, creators felt that the work now no longer has their personality, and thus is no longer their creation.

4.2.1 Influences to Creative Process

The top section *Influences to Creative Process* lists tasks that directly influence the final artistic product, allowing for integrity of expression by the creator. On the left side are *acceptable influences*. These tasks involve aspects that help prompt ideas or create inspiration. Tasks that fall within the *acceptable influences* do not need to be as integrated with the programs creators already use, though they should integrate with the overall creative process - especially in the ideation phase, where many creators felt AI influences fit best. These tools should allow for continuous reiteration, with understandable in-tool design signifiers that indicate the ways they can edit, change, and manipulate the AI's data before and after each iteration. Once this ideation phase is over, there should be a clear way to export their ideas into a new software or system, again allowing for the interoperability that is vital for music creator's process; this could be done in a number of ways such as using MIDI files, WAV files, or MusicXML.

Participants also discussed ways that AI tools could go beyond what humans are traditionally capable of, and in that way become a partner in the expansion of their compositional capabilities. One potential function as noted by a participant was the ability to use AI as a music analysis tool, helping users pinpoint things they were not aware of or even able to perceive with human hearing, such as "the sound field...expanding from the front to the back" (P6). Another participant described how their current use of AI as part of their process has changed how they see the world around them, gaining a new understanding around what could be used or turned into data which allows them to create patterns and connections within their music (P5). AI tools also helped many creators find relationships between sounds, found materials, words, and pictures. One participant explained it as a "feedback loop" (P6).

The right side displays aspects of the creative process where creators are not comfortable engaging in Human-AI collaboration. They felt using AI with these tasks negatively affect the creative process by taking away an essential component of their creations. This includes losing the ability to control their intent and choices, not being able to specify performance parameters/low-fidelity outputs, and interfering with their process and creative personality.

4.2.2 Mechanisms of Creation

The lower section is titled *Mechanisms of Creation*. It includes types of Human-AI collaborations where our participants had little issue if AI took over the process completely, often searching for and utilizing AI that could complete these tasks. In general, *mechanisms* are tasks that occur within the creative process that do not require direct decision-making by the music creators, including housekeeping tasks such as file naming, information retrieval tasks such as looking for electronic instruments, and replicating sounds. Many participants noted they would use AI to complete tasks to help speed up their process or complete tasks they did not want to do. These tasks often had to do with analyzing data in some way. For example, P3 noted "I have an idea that I want to do, and I just use the AI to make that idea happen faster."

4.3 Creation as Process

Lastly, on the far right side is a spectrum representing what we call "Creation as Process," emphasizing the role of iteration and thinking that happens during the process of writing music [23, 32]. For all of the participants, within the context of their own creative process, intention and choice was as important as the creative product. One participant stated, "You can have [AI] generate some sort of electronic music code for you and that sort of just skips for me a whole important step in the process, because in my process of creating live electronic music, there's sort of an interplay between my coding and my writing. I think a lot will be lost if you just take out an entire part of that process" (P3). Another participant remarked, "For me, creativity also involves the decision-making in a big way. And then to determine where to end things. It doesn't seem that my experience with AI so far affords these possibilities" (P6).

The spectrum represents the level of intellectual engagement needed in each task, ranging from highly intentional choices to mechanical and repetitive tasks. Within the process of creating, the given AI tasks may move to higher or lower levels along the spectrum, sometimes influencing the process more and other times receding to lower levels of impact. The creative process is fluid, meaning that both the factors *and* roles of the AI can change over the course of creation.

5. DISCUSSION

Although our focus on only six case-studies of music creators in specific creative contexts presents a limitation to our study, we believe that our focus allowed us to explore possible applications of AI tools to creators' needs, and allowed us to form initial insights into the perception of the use of AI tools in the creative process. Musical creativity is not a monolith and it is our belief that in order to understand how to design specific AI systems that support creative musical tasks, we need to know how creative thinking is conceived by those engaging in these types of musical activities. Our main contribution in this paper is to begin to situate certain AI tasks as potential helpful or potentially harmful to the creative process of those who create.

In this study, we argue that the discussion around the threat that AI poses to both the jobs of creators and artis-

tic integrity is of importance to creators; emphasizing that co-creation of music in the context of music composition is dependent not only on the larger creative context, but on the process of creative thinking as well. Our work suggest that Andersen and Knees [21, 127] notion of the importance of an "individual user[s'] models of music perception as well as a solid understanding of usage context" is not only needed for exploring dissimilarity in search, but also for understanding AI systems in the other forms creative endeavors. Knees et al.'s [50] consideration of the use of "strangeness" for artists recommendations is useful in AI systems as so far the AI is helping to generate new ideas for creators; though strangeness is one aspect of many needs that a creative engages with in AI music systems. Other tasks such as analysis and editing are also elements of creative MIR tasks that may be helpful to facilitating the creative process, though often can occur at different points or simultaneously with the task of idea generation. We argue that there is a need to understand how specific creative processes view and interact with AI at all stages. While there are some aspects of the use of AI that many of our cases agreed on, such as allow AI to take over tasks that have little to no control over the final product of creative thinking which is echoed in other literature [50, 51], our study also indicated that the role of AI is also dependent upon personal, social, and creative goal related factors that are constantly in flux. The Human-AI Collaboration Model demonstrates our belief that the role that AI plays on creative thinking is highly flexible within music creation and that without a clear understanding of how creators are thinking, AI systems can hurt musical creative practices of musicians.

Oliver Bown has warned against the possible negative affect that AI tools can have if it disrupts cultural applications and creation of music. [33]. If music AI systems are designed to limit creator control, intent, or process, they could potentially lead to Schröter's notion of the "(possible) automatization of artistic work" [52]. Full control over the final artistic product and an understanding of the creator's emphasis on their process are the most important aspects to developing tools that can support, instead of harm, human creativity - as noted by Knees, it is important that the user is given agency in the process of "co-creation" with high-level control of the generative process [19].

While it is true that the concept of "Explainable AI" [53–55] can help to educate those who worry about the role AI plays in future creative endeavors, it is not a full solution to the lack of user trust or changing user hesitancy in tool adoption. Recent fears over data misuse by generative AI, backed up by online discussions and even legal investigations into data scraping [56] and intellectual property [57] have made creators fear utilizing AI tools, with creators fearing that AI is trained on data that does not meet their personal artistic goals or actively hurts other artists. Increasing common knowledge about the functions of AI tools would create more trust in these systems and encourage users to integrate them more into their creative process [58], but there is also a need to design in systems in such a way that creators feel they can ethically use these systems in their own work. This means ethically sourcing material and allowing for the control of elements within AI systems.

Designers of AI tools for creators should consider what role they expect for their tool to play within specific creation processes and make choices that support this. The specific inputs, outputs, and needs of an AI system will change over the creative process. Because AI tasks can move up and down in importance, that means that it is highly possible to have a mismatch of the execution and evaluation of AI systems that may lead to less cohesion between the creator and the AI as they will continually need to reevaluate how these systems fit in their workflow.

6. CONCLUSION AND FUTURE WORK

Our findings support that creators have concerns surrounding the transparency of and lack control within AI tools, but that there is still much to do in relation to understanding the exact needs of creative users. In order to develop useful AI tools, designers must consider the specific creation context, existing processes of creators, control of creator, and the fluidity of creating. Our *Human-AI Creative Collaboration Model* is designed to help developers and researchers who create AI systems to consider the variety of factors and influences that exist on creative process and how they might intersect with a creators experience. We hope that this work encourages developers and other MIR researchers to continue to consider advancing Human-AI collaborations that align with music creators' needs. There are a variety of tasks that AI can perform, and considering if tasks are impacting creative thinking in a different phases of creation will allow for a more ethical and productive experience for music creators.

While we interviewed different creation contexts within our case studies, there is still a need for future work to consider how differences in cultural background, musical training, and experience with AI factor into Human-AI creative thinking. Composing music can happen in a variety of other contexts not explored in this study, including as part of music education and cultural situations. Yet, composition is only one form of creative thinking within music and future work might will continue work to identify the differences that arise when using AI systems within different forms of creation such as music analysis and performance. Creators may be utilizing all these forms of thinking across the creative process in non-linear ways. There is still much to learn about the impact that AI will have on music as an art; if designed and deployed ethically, AI offers the opportunity to enhance human creation and provide new avenues for creating and learning about music. But, in order for AI to support musicians in any form of creative thinking, we need to ensure we are designing AI tools with creators in mind.

7. REFERENCES

[1] "Cyanite ai-based music tagging system," https://cyanite.ai/, accessed: 2023-03-30.

[2] "VOCALOID ai engine," https://www.vocaloid.com/en/news/news_001/, accessed: 2023-03-30.

[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 6th International Workshop on Musical Metacreation*. Charlotte, United States: MUME, Jun. 2019, p. 7. [Online]. Available: https://doi.org/10.5281/zenodo.4285266

[4] "AIVA artificial intelligence composer," https://www.aiva.ai/, accessed: 2023-03-30.

[5] "Sony Flow Composer," https://www.flow-machines.com/history/projects/flowcomposer-composing-with-ai/, accessed: 2023-03-30.

[6] "LANDR mastering software," https://www.landr.com/online-audio-mastering/, accessed: 2023-03-30.

[7] "Ozone izotope mastering software," https://www.izotope.com/en/products/ozone.html, accessed: 2023-03-30.

[8] J. Hong, "Bias in perception of art produced by artificial intelligence," in *International Conference on Human-Computer Interaction: Interaction in Context*, 2018.

[9] C. Moruzzi, "Should human artists fear ai? : A report on the perception of creative ai," in *xCoAx 2020 : Proceedings of the Eighth Conference on Computation, Communication, Aesthetics & X*, M. Verdicchio, M. Carvalhais, L. Ribas, and A. Rangel, Eds. Porto: Universidade do Porto, 2020, pp. 170–185. [Online]. Available: https://2020.xcoax.org/xCoAx2020.pdf

[10] M. Mazzone and A. Elgammal, "Art, creativity, and the potential of artificial intelligence," *Arts*, vol. 8, no. 1, p. 26, 2019.

[11] J. Hong, Q. Peng, and D. Williams, "Are you ready for artificial mozart and skrillex? an experiment testing expectancy violation theory and ai music," *New Media & Society*, vol. 23, no. 7, pp. 1920–1935, 2021.

[12] H. Zulić, "How ai can change/improve/influence music composition, performance and education: Three case studies," *INSAM Journal of Contemporary Music*, vol. 1, no. 2, pp. 100–114, 2019.

[13] D. Zlatkov, J. Ens, and P. Pasquier, "Searching for human bias against ai-composed music," in *Artificial Intelligence in Music, Sound, Art and Design: 12th International Conference, EvoMUSART 2023, Held as Part of EvoStar 2023, Brno, Czech Republic, April 12–14, 2023, Proceedings*. Berlin, Heidelberg: Springer-Verlag, 2023, p. 308–323. [Online]. Available: https://doi.org/10.1007/978-3-031-29956-8_20

[14] F. Tigre, F. Moura, and C. Maw, "Artificial intelligence became beethoven: how do listeners and music professionals perceive artificially composed music?" *Journal of Consumer Marketing*, vol. 38, no. 2, pp. 137–146, 2021.

[15] K. Lee, G. Hitt, E. Terada, and J. Lee, "Ethics of singing voice synthesis: Perceptions of users and developers," in *Proc. of the 23rd Int. Society for Music Information Retrieval Conf.*, Bengaluru, India, 2022, pp. 733–740.

[16] E. Drott, "Copyright, compensation, and commons in the music ai industry," *Creative Industries Journal*, vol. 14, no. 2, pp. 190–207, 2021.

[17] G. Micchi, L. Bigo, M. Giraud, R. Groultand, and F. Levé, "I keep counting: An experiment in human/ai co-creative songwriting," *Transactions of the International Society for Music Information Retrieval*, vol. 4, no. 1, p. 263–275, 2021.

[18] M. Rohrmeier, "On creativity, music's ai completeness, and four challenges for artificial musical creativity," *Transactions of the International Society for Music Information Retrieval*, vol. 5, no. 1, p. 50–66, 2022.

[19] P. Knees, M. Schedl, and M. Goto, "Intelligent user interfaces for music discovery," *Transactions of the International Society for Music Information Retrieval*, vol. 3, no. 1, p. 165–179, 2020.

[20] 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*, vol. 4, no. 1, p. 105–113, 2021.

[21] P. K. Kristina Andersen, "Conversations with expert users in music retrieval and research challenges for creative mir," in *17th International Society for Music Information Retrieval Conference*, 2016, pp. 122–128.

[22] C.-E. Cella, "Music information retrieval and contemporary classical music: A successful failure." *Transactions of the International Society for Music Information Retrieval*, vol. 3, no. 1, pp. 126–136, 2020.

[23] P. R. Webster, "Creativity as creative thinking," *Music Educators Journal*, vol. 76, no. 9, pp. 22–28, 1990.

[24] A.-M. Gioti, "From artificial to extended intelligence in music composition," *Organised Sound*, vol. 25, no. 1, p. 25–32, 2020.

[25] P. Pasquier, A. Eigenfeldt, O. Bown, and S. Dubnov, "An introduction to musical metacreation," *Comput. Entertain.*, vol. 14, no. 2, jan 2017. [Online]. Available: https://doi.org/10.1145/2930672

[26] G. Nierhaus, "Historical development of algorithmic procedures," in *Algorithmic Composition: Paradigms of Automated Music Generation*. Vienna, Austria: Springer Vienna, 2008, pp. 7–66.

[27] D. Bouche, J. Nika, A. Chechile, and J. Bresson, "Computer-aided composition of musical processes," *Journal of New Music Research*, vol. 46, no. 1, pp. 3–14, 2017.

[28] A. Alpern, "Techniques for algorithmic composition of music," *Hampshire College*, 1995.

[29] Janusz Podrazik, "Opusmodus." [Online]. Available: https://opusmodus.com/

[30] Cycling 74, "Max." [Online]. Available: https://cycling74.com/products/max

[31] M. Pearce, D. Meredith, and G. Wiggins, "Motivations and methodologies for automation of the compositional process," *Musicae Scientiae*, vol. 6, no. 2, pp. 119–147, 2002. [Online]. Available: https://doi.org/10.1177/102986490200600203

[32] D. Cope, *Computer models of musical creativity*. MIT Press, 2005.

[33] O. Bown, "Sociocultural and design perspectives on ai-based music production: Why do we make music and what changes if ai makes it for us?" in *Handbook of Artificial Intelligence for Music*, E. R. Miranda, Ed. Cham: Springer, 2021.

[34] 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, no. 14, pp. 111–222, April 2020.

[35] C. Lamb, D. G. Brown, and C. L. A. Clarke, "Evaluating computational creativity: An interdisciplinary tutorial," *ACM Computing Surveys*, vol. 51, no. 2, feb 2018.

[36] F. Nake, "Creativity in algorithmic art," in *Proceedings of the Seventh ACM Conference on Creativity and Cognition*, Berkeley, California, USA, 2009, pp. 97–106.

[37] M. Lee, P. Liang, and Q. Yang, "Coauthor: Designing a human-ai collaborative writing dataset for exploring language model capabilities," in *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, ser. CHI '22. New York, NY, USA: Association for Computing Machinery, 2022. [Online]. Available: https://doi.org/10.1145/3491102.3502030

[38] G. A. Wiggins, "A preliminary framework for description, analysis and comparison of creative systems," *Know.-Based Syst.*, vol. 19, no. 7, p. 449–458, nov 2006. [Online]. Available: https://doi.org/10.1016/j.knosys.2006.04.009

[39] S. Tipei, A. Craig, and P. Rodriguez, "Using high-performance computers to enable collaborative and interactive composition with dissco." *Multimodal Technologies and Interaction*, vol. 5, no. 24, 2006.

[40] R. K. Yin, *Base Study Research and Applications : Design and Methods*. Sage Publications, 2017.

[41] B. E. White, S. J. Gandhi, A. Gorod, V. Ireland, and B. Sauser, "On the importance and value of case studies," in *2013 IEEE International Systems Conference (SysCon)*, 2013, pp. 114–122.

[42] J. Seawright and J. Gerring, "Case selection techniques in case study research: A menu of qualitative and quantitative options," *Political Research Quarterly*, vol. 61, no. 2, p. 294–308, 2008.

[43] J. Corbin and A. Strauss, *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*. Sage Publications, 2014.

[44] C. E. Hill, S. Knox, B. Thompson, E. Williams, S. Hess, and N. Ladany, "Consensual qualitative research: An update," *Journal of Counseling Psychology*, vol. 52, no. 2, p. 196–205, 2005.

[45] J.-P. Fourmentraux, "Internet artworks, artists and computer programmers: Sharing the creative process," *Leonardo*, vol. 39, no. 1, p. 44–50, 2021.

[46] Y. Kjus, "The use of copyright in digital times: A study of how artists exercise their rights in norway," *Popular Music and Society*, vol. 44, no. 3, pp. 241–257, 2021.

[47] "Logic Pro virtual drummer," https://support.apple.com/guide/logicpro/drummer-lgcpa4324884/mac, accessed: 2023-03-30.

[48] V. P. Glăveanu, "Creativity as a sociocultural act," *The Journal of Creative Behavior*, vol. 49, no. 3, pp. 165–180, 2015. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/jocb.94

[49] J. Hughes, "The philosophy of intellectual property," *Georgetown Law Journal*, vol. 77, p. 287, 1988.

[50] P. Knees, K. Andersen, and M. Tkal˘ci˘c, ""I'd like it to do the opposite": Music-Making Between Recommendation and Obstruction," in *Proceedings of the 2nd International Workshop on Decision Making and Recommender Systems*, vol. 1533. Charlotte, United States: MUME, 2015. [Online]. Available: CEUR-WS.org.

[51] D. Buschek, L. Mecke, F. Lehmann, and H. Dang, "Nine potential pitfalls when designing human-ai co-creative systems," *CoRR*, vol. abs/2104.00358, 2021. [Online]. Available: https://arxiv.org/abs/2104.00358

[52] J. Schröter, "Artificial intelligence and the democratization of art," in *The Democratization of Artificial Intelligence: Net Politics in the Era of Learning Algorithms*, A. Sudmann, Ed. Transcript publishing, 2019, pp. 297–311.

[53] C. Zednik, "Solving the black box problem: A normative framework for explainable artificial intelligence," *Philosophy & Technology volume*, vol. 34, p. 265–288, 2021.

[54] A. Holzinger, "From machine learning to explainable ai," in *2018 World Symposium on Digital Intelligence for Systems and Machines (DISA)*, 2018, pp. 55–66.

[55] W. Samek, G. Montavon, A. Vedaldi, L. K. Hansen, and K.-R. Müller, *Explainable AI: Interpreting, explaining and Visualizing Deep Learning*. Springer, 2019.

[56] C. Fiesler, N. Beard, and B. C. Keegan, "No robots, spiders, or scrapers: Legal and ethical regulation of data collection methods in social media terms of service," *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 14, p. 187–196, 2020.

[57] T. Zipper, "Mind Over Matter: Addressing Challenges of Computer-Generated Works Under Copyright Law," *Wake Forest Journal of Business and Intellectual Property Law*, aug 7 2022, https://jbipl.pubpub.org/pub/zn744tze.

[58] E. Ruane, A. Birhane, and A. Ventresque, "Conversational ai: Social and ethical considerations," in *Irish Conference on Artificial Intelligence and Cognitive Science*, 2019.

IMPACT OF TIME AND NOTE DURATION TOKENIZATIONS ON DEEP LEARNING SYMBOLIC MUSIC MODELING

Nathan Fradet[1,2] **Nicolas Gutowski**[3] **Fabien Chhel**[3,4] **Jean-Pierre Briot**[1]
[1] Sorbonne University, CNRS, LIP6, F-75005 Paris
[2] Aubay, Boulogne-Billancourt, France
[3] University of Angers, LERIA, 49000 Angers, France
[4] ESEO-TECH / ERIS, 49100 Angers, France
nathan.fradet@lip6.fr

ABSTRACT

Symbolic music is widely used in various deep learning tasks, including generation, transcription, synthesis, and Music Information Retrieval (MIR). It is mostly employed with discrete models like Transformers, which require music to be tokenized, i.e., formatted into sequences of distinct elements called tokens. Tokenization can be performed in different ways, and recent research has focused on developing more efficient methods. However, the key differences between these methods are often unclear, and few studies have compared them. In this work, we analyze the current common tokenization methods and experiment with time and note duration representations. We compare the performance of these two impactful criteria on several tasks, including composer classification, emotion classification, music generation, and sequence representation. We demonstrate that explicit information leads to better results depending on the task.

1. INTRODUCTION

Most tasks involving using deep learning with symbolic music [1] are performed with discrete models, such as Transformers [2]. To use these models, the music must first be formatted into sequences of distinct elements, commonly called tokens. For instance, a token can represent a note attribute or a time event. The set of all known tokens is commonly called the vocabulary, and each token is associated to a unique integer id. These ids are used as input and output of models.

Compared to text, tokenizing music provides greater flexibility, as a musical piece can be played by different instruments and composed of multiple simultaneous notes, each having several properties such as pitch, duration and velocity. As a result, it is necessary to represent these elements in conjunction with the time dimension. To achieve this, researchers have developed various methods of tokenizing music, which are introduced in the next section.

While these works offer model performance comparisons between tokenization strategies, their main differences or similarities are not always clearly stated. Few experiments have been conducted to compare model performances using different tokenization strategies. Additionally, these studies mostly focus on music generation, for which evaluations are performed on results obtained autoregressively, which accumulates biases [3] and is arguably difficult to evaluate [4].

This paper's primary contribution is a thorough and well-designed comparison of common tokenization techniques. Our focus is on two critical aspects: the representation of time and note duration. We believe that they are significant and impactful design choices for any music tokenization approach. Through experiments on composer classification, emotion classification, music generation, and sequence representation, we demonstrate that these design choices produce varying results depending on the task, model type, and inference process. Autoregressive generation benefits from explicit note duration and time shift tokens, while explicit note offset is more discriminating better suited for contrastive learning approaches.

We present next the related works, followed by an analysis of music tokenization, experimental results, and finally a conclusion. The source code is available for reproducibility. [1]

2. DECOMPOSING MUSIC TOKENIZATION

2.1 Related works

Early works using discrete models for symbolic music, such as DeepBach [5] or FolkRNN [6], rely on specific tokenizations often tied to their training data. Since then, researchers introduced more general representations applicable to any kind of music. The most commonly used are *Midi-Like* [7] and *REMI* [8]. The former tokenizes music by representing tokens as the same types of events from the MIDI protocol, while the latter represents time with *Bar* and *Position* tokens and note durations with explicit

[1] https://github.com/Natooz/time-duration-music-modeling

© 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, "Impact of time and note duration tokenizations on deep learning symbolic music modeling", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.*, Milan, Italy, 2023.

	Time		Note duration	
Tokenization	`TimeShift`	`Bar` + `Pos.`	`Duration`	`NoteOff`
MIDI-Like [7]	√	-	-	√
REMI [8]	-	√	√	-
Structured [17]	√	-	√	-
TSD [15]	√	-	√	-
Octuple [10]	-	√	√	-

Table 1: Time and note duration representations of common tokenizations. `Pos.` stands for Position.

Duration tokens. Additionally, *REMI* includes tokens with additional information such as chords and tempo.

More recently, researchers have focused on improving the efficiency of models with new tokenizations techniques: *Compound Word* [9], *Octuple* [10] and *PopMAG* [11] merge embedding vectors before passing them to the model; 2) LakhNES [12] and [13], SymphonyNet [14] and [15] use tokens combining several values, such as pitch and vocabulary.

2.2 Music tokenization design

When analyzing the possible designs of music tokenization, we can distinguish seven key dimensions:

- **Time**: Type of token representing time, either *TimeShift* indicating time movements, or *Bar* and *Position* indicating new bars and the positions of the notes within them. We can also consider the unit of *Time-Shift* tokens, either in beats or in seconds. [2]
- **Notes duration**: How notes durations are represented, with either *Duration* or *NoteOff* tokens.
- **Pitch**: Most works use tokens representing absolute pitch values, although recent work shed light on the expressiveness gain of representing as intervals instead [16];
- **Multitrack representation**: The representation of several music tracks in a sequence, i.e., how are the notes linked to their associated track.
- **Additional information**: Any additional information such as chords, tempo, rests, note density. Velocity can also falls in this category;
- **Downsampling**: How "continuous-like" features are downsampled into discrete sets, e.g. the 128 velocity values reduced to 16 values;
- **Sequence compression**: Methods to reduce the sequence lengths, such as merging tokens and embedding vectors.

As time and note duration can both be represented in two different ways, existing tokenizations can be easily classified based on these dimensions, as shown in table 1. However, other dimensions offer a broader spectrum of potential designs.

For instance multitrack can be represented by `Program` tokens [3] preceding notes as in FIGARO [18], distinct tracks sequences separated by `Program` tokens as in MMM [19], combined note and instrument tokens as LakhNes [12] and MuseNet [13], or merging `Program` embeddings with the associated note tokens (MMT [20], MusicBert [10]). One could even infer each sequence separately and lately model their relationships with operations aggregating their hidden states an in ColBERT [21].

The MIDI protocol supports a set of effects and metadata that can also be represented when tokenizing symbolic music, such as tempo, time signature, sustain pedal or control changes. Some works also include explicit `Chord` tokens, detected with rule-based methods. Nevertheless, only a few works experimented with such additional tokens so far ([8, 22]).

Previous works have mainly compared tokenization strategies by evaluating models with automatic and sometimes subjective (human) metrics, but often do not proceed to comparisons between the ways to represent one of the dimensions we introduced previously. [8] compared results for the generation task, for the use of `Bar` and `Position` tokens versus `TimeShift` in seconds and beats.

To the best of our knowledge, no comprehensive work and empirical analysis have fairly compared these possible tokenization choices. Conducting such an assessment would require an extensive survey. In this paper, we specifically focus on the time and note duration representations, as they are the two main characteristics present in every tokenization.

We want to highlight the importance of the explicit information carried by the token types, as they directly impact the performances of models. `TimeShift` tokens represent explicit time movements, and especially the time distances between successive notes. On the other hand, `Bar` and `Position` tokens bring explicit information on the absolute positions (within bars) of the notes, but not the onset distances between notes. One could assume that the former might help to model melodies, and the latter rhythm and structure. For note duration, `Duration` tokens intuitively express the absolute durations of the notes, while `NoteOff` tokens explicitly indicates the offset times. With `NoteOff`, a model would have to model note durations from the combinations of previous time tokens.

Our experiments aim to demonstrate the impact of different combinations of time and note duration tokens on model performance and which combinations are suitable for different tasks. Next, we introduce our methodology.

3. METHODOLOGY

3.1 Models and trainings

For all experiments, we use the Transformer architecture [2], with the same model dimensions: 12 layers, with di-

[2] In this paper we only treat of the beat unit. The MIDI protocol represents time in *tick* unit, which value is proportional to the time division (in ticks per beat) and tempo. Hence, working with seconds would require a conversion from ticks.

[3] Following the conventional programs from the MIDI protocol.

mension of 768 units, 12 attention heads and inner feed-forward layers of 3072.

For classification and sequence representation, it is first pretrained on 100k steps and a learning rate of 10^{-4}, then finetuned on 50k steps and a learning rate of 3×10^{-5}, with a batch size of 48 examples. An exception is made for the EMOPIA dataset, for which we set 30k pretraining steps and 15k finetuning steps, as it is fairly small. These models are based on the BERT [23] implementation of the Transformers library [24]. We use the same pretraining than the original BERT: 1) from 15% of the input tokens, 80% is masked with a special `MASK` token, and 20% is randomized; 2) half of the inputs have 50% of their tokens (starting from the end) shuffled and separated with a special `SEP` token, and the model is trained to detect if the second part is the next of the first.

For generation, the model is based on the GPT2 implementation of the Transformers library [24]: it uses a causal attention mask, so that for each element in the sequence, the model can only attend to the current and previous elements. The training is performed with teacher forcing, the cross-entropy loss is defined as: $\ell = -\sum_{t=1}^{n} \log p_\theta(x_t|\mathbf{x}_{\leq n})$.

All trainings are performed on V100 GPUs, using automatic mixed precision [25], the Adam optimizer [26] with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$, and dropout, weight decay and a gradient clip norm of respectively 10^{-1}, 10^{-2} and 3. Learning rates follow a warm-up schedule: they are initially set to 0, and increase to their default value during the first 30% of training, then slowly decrease back to 0.

10% of the data is used for validation during training, and 15% to test models. Inputs contains 384 to 512 tokens, and begin with a `BOS` (Beginning of Sequence) token and end with a `EOS` (End of Sequence) one.

3.2 Tokenizations

We investigate here the four combinations of possible time and note duration representation. In the results, we refer to them as *TS* (`TimeShift`), *Pos* (`Position`), *Dur* (`Duration`) and *NOff* (`NoteOff`). It is worth noting that *TS* + *Dur* is equivalent to *TSD* [15] and *Structured* [17], *TS* + *NOff* is equivalent to *MIDI-Like* [7], and *Pos* + *Dur* is equivalent to *REMI* (without additional tokens for chords and tempo).

We apply different resolutions for `Duration` and `TimeShift` token values: those up to one beat are downsampled to 8 samples per beat (spb), those from one to two beats to 4 spb, those from two to four beats to 2 spb, and those from four to eight beats to 1 spb. Thus, short notes are represented more precisely than longer ones. `Position` tokens are downsampled to 8 spb, resulting in 32 different tokens as we only consider the 4/* time signature. This allows to represent the 16th note. We only consider pitches within the recommended range for piano (program 0) specified in the General MIDI 2 specifications [4]: 21 to 108. We then deduplicate all duplicated notes. Velocities are downsampled to 8 distinct values. No additional token (e.g., `Chord`, `Tempo`) is used.

We perform data augmentation by creating variations of the original data with pitches increased and decreased by two octaves, and velocity by one value. Finally, following [15], we use Byte Pair Encoding to build the vocabularies up to 2k tokens for generation and 5k for other tasks. All these preprocessing and tokenization steps were performed with MidiTok [27].

4. GENERATION

For the generative task, we use the POP909 dataset [28]. The models start with prompt made of between 384 to 512 tokens, then autoregressively generate 512 additional tokens. Evaluation of generated results remains an open issue [4]. Previous work often perform measures of similarity of certain features such as pitch range or class, between prompts and generated results, alongside human evaluations. Feature similarity is however arguably not very insightful: a generated result could have very similar features to its prompts while being of poor quality. Human evaluations, while being more reliable on the quality can also induce biases. Besides, [8] already shows results on an experiment similar to ours.

Hence we choose to evaluate results on the ratios of prediction errors: Token Syntax Error (TSE) [15]. This metric is bias-free and directly linked to the design choices of the tokenizations. It allows us to measure how a model achieves to make reliable predictions based on the input context and the knowledge it learned.

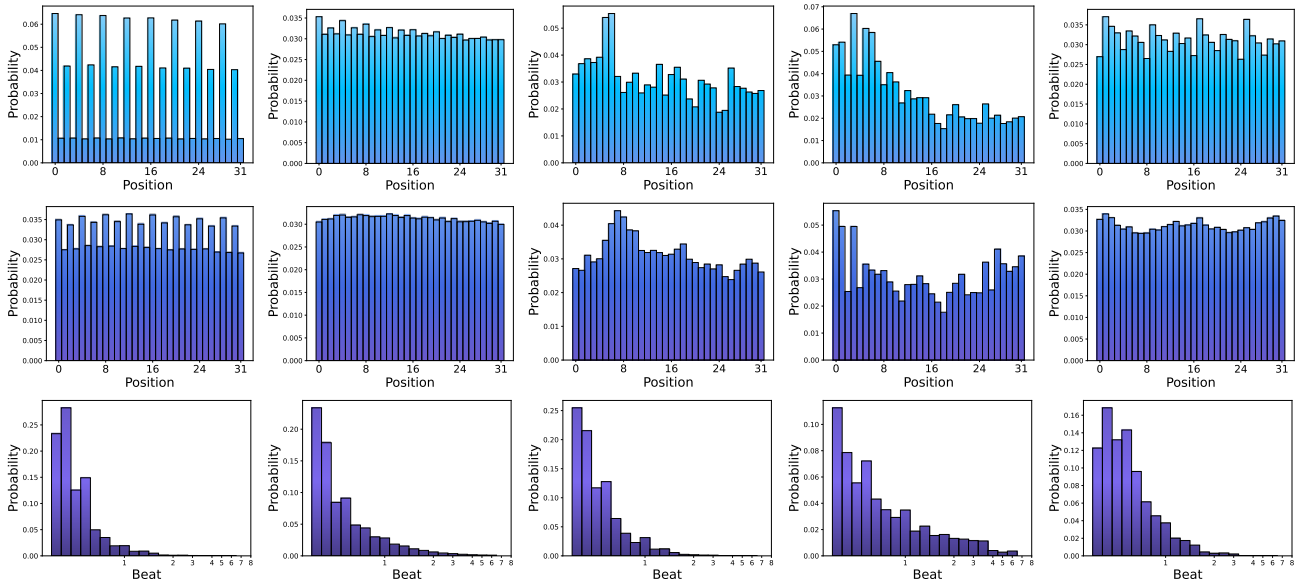
We use the categories from [15]:

- $\mathbf{TSE_{type}}$: an error of type, e.g., when the model predicts a token of an incompatible type with the previous one.
- $\mathbf{TSE_{time}}$: a wrong predicted `Position` value, that goes back or stay in time.
- $\mathbf{TSE_{dupn}}$ (duplicated note): a note predicted whereas it was already being played at the current time being.
- $\mathbf{TSE_{nnof}}$ (no NoteOff): a `NoteOn` token been predicted with no following `NoteOff` token to end it.
- $\mathbf{TSE_{nnon}}$ (no NoteOn): `NoteOff` token predicted whereas this note was not being played.

For each generated token, a rule-based function analyzes its type and value to determine if both are valid, or which type of error was made otherwise. The overall number of errors is normalized by the number of predicted tokens.

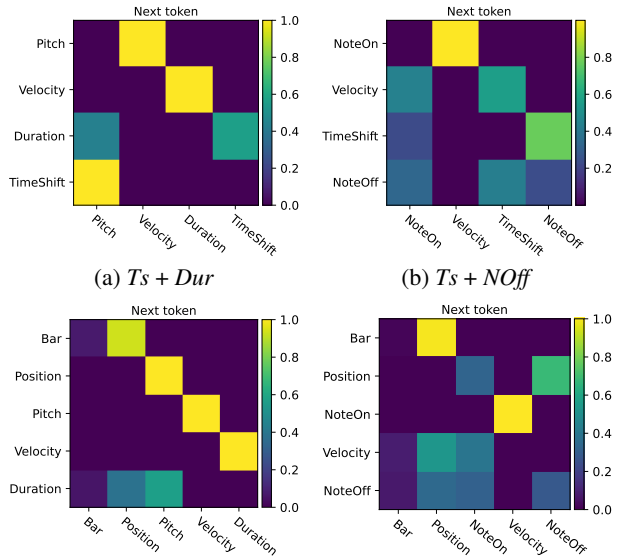
The results are reported in table 2. We first observe that the type error ratios are lower than in other categories. This is excepted since it is less computationally demanding to model the possible next types depending solely on the last one, rather than on the value of the predicted token, for

[4] Available on the MIDI Manufacturers Association website.



(a) *Ts + Dur* (b) *Ts + NOff* (c) *Pos + Dur* (d) *Pos + NOff* (e) POP909 dataset

Figure 1: Histograms of the note onset positions within bars (top-row), note offset positions within bars (middle-row) and note durations (bottom-row) of the generated notes. There are 32 possible positions within a bar, numerated from 0 (beginning of bar) to 31 (last 32th note). The durations are expressed in beats, ranging from a 32th note to 8 beats.



(a) *Ts + Dur* (b) *Ts + NOff* (c) *Pos + Dur* (d) *Pos + NOff*

Figure 2: Token type succession heatmaps of the generated results. The horizontal axis denotes the next token type per from the ones on the vertical axis. Each row is normalized to a sum of 1.

which the validity depends on a the whole previous context.

`Position` tokens bring almost no type errors, but a noticeable proportion of time errors. When decoding tokens to notes, this means that the time may go backward, and resulting in sections of overlapping notes.

Although `Duration` tokens seem to bring slightly more note duplication errors, the use of `NoteOn` and `NoteOff` tokens results in a considerable proportion of

Tokenization	$\mathbf{TSE_{type}} \downarrow$	$\mathbf{TSE_{time}} \downarrow$	$\mathbf{TSE_{dupn}} \downarrow$	$\mathbf{TSE_{nnon}} \downarrow$	$\mathbf{TSE_{nnof}} \downarrow$
TS + Dur	$< 10^{-3}$	-	0.014	-	-
TS + NOff	$< 10^{-3}$	-	0.001	0.109	0.040
Pos + Dur	0.002	0.113	0.032	-	-
Pos + NOff	0.002	0.127	0.005	0.095	0.066

Table 2: Prediction error ratios when performing autoregressive generation. - symbol stands for not concerned, and can be interpreted as 0.

note prediction errors. `NoteOff` tokens predicted while the associated note was not being played ($\mathbf{TSE_{nnon}}$) does not have undesirable consequences when decoding tokens to notes, but it pointlessly extends the sequence, reducing the efficiency of the model, and may mislead the next token predictions. Additionaly, `NoteOn` tokens predicted without associated `NoteOff` ($\mathbf{TSE_{nnof}}$) result in notes not properly ended. This error can only be handled by applying a maximum note duration after decoding. Explicit `Duration` tokens allows to specify in advance this information, for both short and long notes. Conversely, with `NoteOff` tokens, the note duration information is implicit and inferred by the combinations of `NoteOn`, `NoteOff` and time tokens. This can be interpreted as an extra effort for the model. Consequently, some uncertainty on the duration accumulates over autoregressive steps during generation. Based on these results, the best tradeoff ensuring good predictions seems to represent time with `TimeShift` tokens and note duration with `Duration` tokens.

In fig. 1 we observe the positions within bars and durations of the generated notes. In all cases, onset positions are more distributed at the beginning of the bars. This is especially the case with `Bar` and `Position` tokens, for which we may find unexpected rests at the end of bars,

when `Bar` tokens are predicted during the generation before that the current bar is completed. The *TS + Dur* combination places note onsets much more on even positions. The probability mass of *TimeShift* tokens (especially for short values) seems to be much higher. However, this is not the case for the *TS + NOff* combination, as `TimeShift` tokens have to be predicted to move the time on odd positions of note offsets. As shown in fig. 2, right after the model is likely to predict a next note, resulting in evenly distributed onset distribution.

Finally, the use of `NoteOff` tokens tends to produce longer note durations, especially when combined with `Position` tokens. In this last case, we can assume that the model might "forget" the notes currently being played, and that it struggles more to model their durations that have to be implicitly deduced from the past `Bar` and `Position` tokens.

Tokenization	Top-20 composers ↑	Top-100 composers ↑	Emotion ↑
TS + Dur	**0.973**	**0.941**	**0.983**
TS + NOff	0.962	0.930	0.962
Pos + Dur	0.969	0.927	0.963
Pos + NOff	0.963	0.925	0.956

Table 3: Accuracy on classification tasks.

5. CLASSIFICATION

For some classification tasks, symbolic music is arguably better suited than audio or piano roll. This is particularly true for classical music feature classification, such as composer [29]. Mono-instrument music with complex melodies and harmonies and no particular audio effect benefit from being represented as discrete for classification and modeling tasks. Given this, it felt important to us to conduct experiments on such task.

We choose to experiment with the GiantMIDI [30] dataset for composer classification and the EMOPIA [31] dataset for emotion classification. The results, as shown in table 3, indicate that there is very little difference between the various tokenization methods. However, the combination of `TimeShift` and `Duration` consistently outperforms the others by one point

The classification task involves modeling the patterns from data that are characteristic to composers or emotions. Here, it seems that the time distance between notes, and their explicit duration play a role in these task, more than note offsets or onset positions. This comes with no surprise for the composer classification task, considering that the data is largely composed of complex music with dense melodies and harmonies, featuring mostly short successive notes. Intuitively, patterns of note successions and chords are more easily distinguishable with explicit durations. With implicit note durations, the overall patterns must be deduced by the combinations of `NoteOn` and `NoteOff` tokens while keeping track of the time.

6. SEQUENCE REPRESENTATION

The last task that we wished to explore is sequence representation. It consists in obtaining a fixed size embedding representation of an input sequence of tokens $p_\theta : \mathbb{V}^L \mapsto \mathbb{R}^d$. Here $\mathbb{V} \subset \mathbb{N}$ denotes the token ids of the vocabulary $\mathcal{V}$, L is the variable input sequence length, and d the size of embeddings. In other words, the model learns to project an input token sequence into a embedding space, thus providing a universal representation. We find this task interesting and well-suited to assess model performances as it directly trains it to model the relationships between tokens within the input sequence and between different representations themselves. While the real-world applications of this task for symbolic music are currently limited, it serves as a useful benchmarking technique for measuring how tokenization impacts the learning of models.

This task has previously been addressed in natural language processing by SentenceBERT [32] or SimCSE [33]. We adopted the approach of the latter, which uses contrastive learning to train the model to learn sequence representations, for which similar inputs have higher cosine similarities. The sequence embedding is obtained by performing a pooling operation on the output hidden states of the model. We decided to use the last hidden state of the `BOS` token position, as it yielded good results with SimCSE [33] [5]. We trained the models with the dropout method: during training, a batch of n sequences $\mathcal{X} = \{\mathbf{x}_i\}_{i=0}^n$ is passed twice to the model, but with different dropout masks, resulting in different output sequence embeddings $\mathcal{Z} = \{\mathbf{z}_i\}_{i=0}^N$ and $\bar{\mathcal{Z}} = \{\bar{\mathbf{z}}_i\}_{i=0}^N$. Although the dropout altered the outputs, most of the input information is still accessible to the model. Hence, we expect pairs of sequence embeddings $(\mathbf{z}_i, \bar{\mathbf{z}}_i)$ to be similar, so having a high cosine similarity. To achieve this objective, we train the model with a loss function defined by the cross-entropy for in-batch pairwise cosine similarities (sim):

$$\ell_i = -\log \frac{e^{\mathrm{sim}(\mathbf{z}_i, \bar{\mathbf{z}}_i)/\tau}}{\sum_{j=1}^N e^{\mathrm{sim}(\mathbf{z}_i, \bar{\mathbf{z}}_j)/\tau}} \tag{1}$$

As a result, the model will effectively learn to create similar sequence embeddings for similar inputs, while pushing apart those with dissimilarities. We kept a 0.1 dropout value to train the models, and used the GiantMIDI dataset [30].

Evaluation of sequence representation is intuitively performed by measuring the distances and similarities of pairs of similar sequences. We resort to data augmentation by shifting the pitch and velocity of the sequences in order to get pairs of similar music sequences. The augmented data keeps most of the information of the original data. As such, the models are expected to produce similar embeddings for pairs of original-augmented sequence. Ideally, the cosine similarity should be high, yet not to be equal to 1, as this would indicate that the model fails to capture the differences between the two sequences. The results, presented in fig. 3, indicate that `Position`-based tokenizations per-

[5] SimCSE uses a `CLS` token which is equivalent to `BOS` in our case.

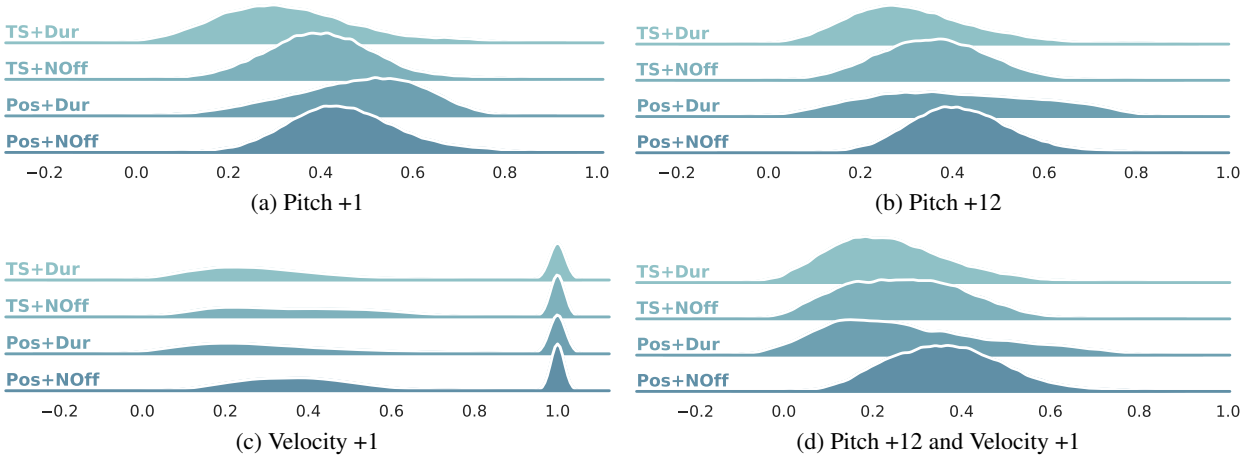


Figure 3: Density plots of cosine similarities between pairs of original and augmented token sequences.

form slightly better. Therefore, it appears that explicit note onset and offset positions information facilitates models to obtains a universal musical representation.

Unlike classification, the contrastive learning objective models the similarities and dissimilarities between examples in the same batch. In this context, note onset and offset positions appear to be helpful for the models to distinguish music.

We also note the contrasting results when augmenting the velocity. Increasing it by one unit, which would be equivalent to playing just a little bit louder, have arguably a very small impact. As a result, the models mostly produces embeddings that are almost identical for the original and the augmented sequences, but also exhibits uncertainty for a notable proportion of samples.

To complement these results, we estimated the isotropy of sets of sequence embeddings. Isotropy measures the uniformity of the variance of a set points in a space. More intuitively, in an isotropic space, the embeddings are evenly distributed. It has been associated with improved performances in natural language tasks [34–36], because embeddings are more equally distant proportionally to the density of their area, and are in turn more distinct and distinguishable. We choose to estimate it with the intrinsic dimension of the sets of embeddings. Intrinsic dimension is the number of dimensions required to represent a set of points. It can be estimated by several manners [37]. We choose Principal Component Analysis (PCA) [38], method of moments (MOM) [39], Two Nearest Neighbors (TwoNN) [40] and FisherS [41]. The results, reported in table 4, show that the embeddings created from the *Pos + Dur* combination tend to occupy more space across the dimension of the model, and are potentially better distributed.

Tokenization	lPCA ↑	MOM ↑	TwoNN ↑	FisherS ↑
TS + Dur	**213**	42.6	34.3	17.5
TS + NOff	161	43.7	32.7	17.5
Pos + Dur	146	39.1	33.1	17.1
Pos + NOff	177	**45.2**	**35.6**	**17.8**

Table 4: Intrinsic dimension of sequence embeddings, as an estimation of isotropy.

7. CONCLUSION

We have discussed the importance of different aspects of symbolic music tokenization, and focused on two major ones: the time and note duration representations. We showed that different tokenization strategies can lead to different model performances due to the explicit information carried by tokens, depending on the task at hand.

Explicitly representing note duration leads to better classification accuracy as it helps the models to capture the melodies and harmonies of a music. Modeling durations, when represented implicitly, adds an extra effort to the model. However, the note offset position information it brings have been found to be more discriminative and effective in our contrastive learning experiment.

For music generation, the time representation plays a significant role, for which the note onset and offsets distributions vary due to the successions of token types. Implicit note durations are less suited for the autoregressive nature of this task, from a prediction error perspective, and sometimes "forgetting" notes being played resulting in higher durations.

We did not explore music transcription, for which we can assume that implicit note durations (note onset and offset) might be better suited. When training with chunks of log-scaled mel-spectrograms as done by [42, 43], these may contain frequencies of unended or not begun notes. Specifying their original durations might approximate onsets might alter model performances.

Future research will further explore the other dimensions of music tokenization, such as multitrack or metadata, on transcription and other tasks analogous to natural language understanding.

8. REFERENCES

[1] J.-P. Briot, G. Hadjeres, and F.-D. Pachet, *Deep Learning Techniques for Music Generation*, ser. Computational Synthesis and Creative Systems. Springer International Publishing, 2020. [Online]. Available: https://www.springer.com/gp/book/9783319701622

[2] 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., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[3] A. Holtzman, J. Buys, L. Du, M. Forbes, and Y. Choi, "The curious case of neural text degeneration," in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=rygGQyrFvH

[4] L.-C. Yang and A. Lerch, "On the evaluation of generative models in music," *Neural Comput. Appl.*, vol. 32, no. 9, p. 4773–4784, 5 2020. [Online]. Available: https://doi.org/10.1007/s00521-018-3849-7

[5] G. Hadjeres, F. Pachet, and F. Nielsen, "DeepBach: a steerable model for Bach chorales generation," in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. PMLR, 8 2017, pp. 1362–1371. [Online]. Available: https://proceedings.mlr.press/v70/hadjeres17a.html

[6] B. L. Sturm, J. F. Santos, and I. Korshunova, "Folk music style modelling by recurrent neural networks with long short-term memory units," in *Extended abstracts for the Late-Breaking Demo Session of the 16th International Society for Music Information Retrieval Conference*, 2015. [Online]. Available: https://ismir2015.ismir.net/LBD/LBD13.pdf

[7] 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, p. 955–967, 2018. [Online]. Available: https://link.springer.com/article/10.1007/s00521-018-3758-9

[8] 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*, ser. MM '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 1180–1188. [Online]. Available: https://doi.org/10.1145/3394171.3413671

[9] 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," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 1, pp. 178–186, 5 2021. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/16091

[10] 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: Association for Computational Linguistics, 8 2021, pp. 791–800. [Online]. Available: https://aclanthology.org/2021.findings-acl.70

[11] 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*. Association for Computing Machinery, 2020, p. 1198–1206. [Online]. Available: https://doi.org/10.1145/3394171.3413721

[12] C. Donahue, H. H. Mao, Y. E. Li, G. W. Cottrell, and J. J. McAuley, "Lakhnes: Improving multi-instrumental music generation with cross-domain pre-training," in *Proceedings of the 20th International Society for Music Information Retrieval Conference, ISMIR 2019, Delft, The Netherlands, November 4-8, 2019*, 2019, pp. 685–692. [Online]. Available: http://archives.ismir.net/ismir2019/paper/000083.pdf

[13] C. Payne, "Musenet," 2019. [Online]. Available: https://openai.com/blog/musenet

[14] 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*. Bengalore, India: ISMIR, Dec. 2022. [Online]. Available: https://arxiv.org/abs/2205.05448

[15] N. Fradet, J.-P. Briot, F. Chhel, A. E. F. Seghrouchni, and N. Gutowski, "Byte Pair Encoding for symbolic music," 2023. [Online]. Available: https://arxiv.org/abs/2301.11975

[16] M. Kermarec, L. Bigo, and M. Keller, "Improving tokenization expressiveness with pitch intervals," in *Extended Abstracts for the Late-Breaking Demo Session of the 23nd International Society for Music Information Retrieval Conference*, 2022. [Online]. Available: https://ismir2022program.ismir.net/lbd_369.html

[17] G. Hadjeres and L. Crestel, "The piano inpainting application," 2021. [Online]. Available: https://arxiv.org/abs/2107.05944

[18] D. von Rütte, L. Biggio, Y. Kilcher, and T. Hofmann, "FIGARO: Controllable music generation using learned and expert features," in

The Eleventh International Conference on Learning Representations, 2023. [Online]. Available: https://openreview.net/forum?id=NyR8OZFHw6i

[19] J. Ens and P. Pasquier, "Mmm : Exploring conditional multi-track music generation with the transformer," 2020. [Online]. Available: https://arxiv.org/abs/2008.06048

[20] 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)*, 2023, pp. 1–5.

[21] O. Khattab and M. Zaharia, "Colbert: Efficient and effective passage search via contextualized late interaction over bert," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 39–48. [Online]. Available: https://doi.org/10.1145/3397271.3401075

[22] J. Ching and y.-h. Yang, "Learning to generate piano music with sustain pedals," 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/0000017.pdf

[23] 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: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. [Online]. Available: https://aclanthology.org/N19-1423

[24] 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 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, Oct. 2020, pp. 38–45. [Online]. Available: https://www.aclweb.org/anthology/2020.emnlp-demos.6

[25] P. Micikevicius, S. Narang, J. Alben, G. Diamos, E. Elsen, D. Garcia, B. Ginsburg, M. Houston, O. Kuchaiev, G. Venkatesh, and H. Wu, "Mixed precision training," in *International Conference on Learning Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=r1gs9JgRZ

[26] 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*, 2015. [Online]. Available: http://arxiv.org/abs/1412.6980

[27] 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://github.com/Natooz/MidiTok

[28] 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, ISMIR*, 2020. [Online]. Available: https://arxiv.org/abs/2008.07142

[29] Q. Kong, K. Choi, and Y. Wang, "Large-scale midi-based composer classification," 2020. [Online]. Available: https://arxiv.org/abs/2010.14805

[30] Q. Kong, B. Li, J. Chen, and Y. Wang, "Giantmidi-piano: A large-scale midi dataset for classical piano music," in *Transactions of the International Society for Music Information Retrieval*, vol. 5, 2021, pp. 87–98. [Online]. Available: https://transactions.ismir.net/articles/10.5334/tismir.80/#

[31] H. Hung, J. Ching, S. Doh, N. Kim, J. Nam, and Y. Yang, "EMOPIA: A multi-modal pop piano dataset for emotion recognition and emotion-based music generation," in *Proceedings of the 22nd International Society for Music Information Retrieval Conference, ISMIR 2021, Online, November 7-12, 2021*, J. H. Lee, A. Lerch, Z. Duan, J. Nam, P. Rao, P. van Kranenburg, and A. Srinivasamurthy, Eds., 2021, pp. 318–325. [Online]. Available: https://archives.ismir.net/ismir2021/paper/000039.pdf

[32] 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*. Association for Computational Linguistics, 11 2019. [Online]. Available: https://arxiv.org/abs/1908.10084

[33] T. Gao, X. Yao, and D. Chen, "SimCSE: Simple contrastive learning of sentence embeddings," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 6894–6910. [Online]. Available: https://aclanthology.org/2021.emnlp-main.552

[34] T. Wang and P. Isola, "Understanding contrastive representation learning through alignment and uniformity on the hypersphere," in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, H. D. III and A. Singh, Eds., vol. 119. PMLR, 13–18 Jul 2020, pp. 9929–9939. [Online]. Available: https://proceedings.mlr.press/v119/wang20k.html

[35] D. Biś, M. Podkorytov, and X. Liu, "Too much in common: Shifting of embeddings in transformer language models and its implications," 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, Jun. 2021, pp. 5117–5130. [Online]. Available: https://aclanthology.org/2021.naacl-main.403

[36] Y. Liang, R. Cao, J. Zheng, J. Ren, and L. Gao, "Learning to remove: Towards isotropic pre-trained bert embedding," in *Artificial Neural Networks and Machine Learning – ICANN 2021: 30th International Conference on Artificial Neural Networks, Bratislava, Slovakia, September 14–17, 2021, Proceedings, Part V*. Berlin, Heidelberg: Springer-Verlag, 2021, p. 448–459. [Online]. Available: https://doi.org/10.1007/978-3-030-86383-8_36

[37] J. Bac, E. M. Mirkes, A. N. Gorban, I. Tyukin, and A. Zinovyev, "Scikit-dimension: A python package for intrinsic dimension estimation," *Entropy*, vol. 23, no. 10, 2021. [Online]. Available: https://www.mdpi.com/1099-4300/23/10/1368

[38] K. Fukunaga and D. Olsen, "An algorithm for finding intrinsic dimensionality of data," *IEEE Transactions on Computers*, vol. C-20, no. 2, pp. 176–183, 1971. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/1671801

[39] L. Amsaleg, O. Chelly, T. Furon, S. Girard, M. E. Houle, K.-I. Kawarabayashi, and M. Nett, "Extreme-value-theoretic estimation of local intrinsic dimensionality," *Data Mining and Knowledge Discovery*, vol. 32, no. 6, pp. 1768–1805, 11 2018. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01864580

[40] E. Facco, M. d'Errico, A. Rodriguez, and A. Laio, "Estimating the intrinsic dimension of datasets by a minimal neighborhood information," *Scientific Reports*, vol. 7, no. 1, p. 12140, 9 2017. [Online]. Available: https://doi.org/10.1038/s41598-017-11873-y

[41] L. Albergante, J. Bac, and A. Zinovyev, "Estimating the effective dimension of large biological datasets using fisher separability analysis," in *International Joint Conference on Neural Networks (IJCNN)*, 7 2019, pp. 1–8. [Online]. Available: https://arxiv.org/abs/1901.06328

[42] C. Hawthorne, I. Simon, R. Swavely, E. Manilow, and J. H. 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. [Online]. Available: https://archives.ismir.net/ismir2021/paper/000030.pdf

[43] J. P. Gardner, I. Simon, E. Manilow, C. Hawthorne, and J. Engel, "MT3: Multi-task multitrack music transcription," in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=iMSjopcOn0p

MUSICAL MICRO-TIMING FOR LIVE CODING

Max Johnson[1], **Mark Gotham**[1,2]

[1] Department of Computer Science and Technology, University of Cambridge
[2] Department of Computer Science, Durham University

ABSTRACT

Micro-timing is an essential part of human music-making, yet it is absent from most computer music systems. Partly to address this gap, we present a novel system for generating music with style-specific micro-timing within the Sonic Pi live coding language. We use a probabilistic approach to control the exact timing according to patterns discovered in new analyses of existing micro-timing data (jembe drumming and Viennese waltz). This implementation also required the introduction of musical metre into Sonic Pi. The new metre and micro-timing systems are inherently flexible, and thus open to a wide range of creative possibilities including (but not limited to): creating new micro-timing profiles for additional styles; expanded definitions of metre; and the free mixing of one micro-timing style with the musical content of another. The code is freely available as a Sonic Pi plug-in and released open source at `https://github.com/MaxTheComputerer/sonicpi-metre`.

1. INTRODUCTION

1.1 Metre *Versus* Rhythm

Metre is distinct from rhythm in that it primarily concerns a kind of mental representation for processing events in musical time; a common analogy casts metre as a "grid" or "template" for categorising rhythmic events [1–3].

Although metre often involves familiar notions such as "the beat", and definitions often emphasise intuitive ideas like regular periodicity, a clear-cut definition of metre is surprisingly hard to pin down. This is especially so when trying to capture the extremely wide range of musical-cultural contexts for which some concept of metre might be relevant. Nevertheless, notwithstanding the complexities of these terms, and putting any more specific definition of these terms to one side, it is reasonable to argue that some form of both "rhythm" and "metre" feature in almost all known musics: "rhythm" in the sense of events occurring in time, and "metre" in some form of semi-regular cycle of event expectation.

© M. Johnson and M. Gotham. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** M. Johnson and M. Gotham, "Musical Micro-Timing for Live Coding", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

1.2 On Micro-Timing in Theory and Practice

"Micro-timing", in turn, refers to the specific timing of both those actual rhythmic "events" and of the metrical "grid" positions. While rhythm and metre are sometimes *modelled* in terms of a completely regular underlying pulse (e.g., 1-1-1-1-) and small integer combinations thereof (e.g., 2-1-1-), it is impossible in practice for a human performer to play with the mechanical precision of identical gaps between successive events. [1] Moreover, musicians make a virtue of this. A close look at the micro-timings in human performance reveals deeply sophisticated, style-specific micro-timing strategies, a.k.a. "groove". [2]

Even when distinguishing between a mental "grid" for events (metre) and the actual placement of those events (rhythm), it is appropriate to discuss micro-timing for both the rhythm and the metre. This distinction is sometimes cast in terms of a difference between "categorical" and "expressive" timing where events may be expressively altered from their expected (categorical) position [5], but note that the "categorical" position itself is also subject to micro-timing strategies because the "expected" positions are not spaced with equal, 1:1 regularity. In short, although micro-timing is sometimes described in terms of "small deviations" from simple (natural number) pulse relations, it is necessary also to consider micro-timing as part of the metre itself. To continue the "metre as grid" analogy: the gaps between grid lines are not evenly spaced.

In practice, these micro-timing durations are too short to learn in a declarative fashion (verbal or mathematical). Partly for that reason, they are also typically neglected by notation systems (including Western staff notation). Nonetheless, these micro-timing strategies clearly *are* taught and learned in the way that most music has been passed on: through listening, playing, and embodiment.

Computers enable us to achieve a level of timing regularity beyond our human capability. As ever, the technology not only extends what we can do, but invites us to consider new techniques, questions, and aesthetics. And the human's micro-timing can of course be combined with the computer's extreme timing precision, as when an MC raps over a beat. Computers are also used to perform the micro-timing *analysis* discussed here. However, computers are currently less exploited as a tool to help us engage in *creative uses* of micro-timing strategies.

[1] This has been systematically studied since Seashore [4].
[2] See, for example, Justin London's "many metres" [2].

2. RELATED WORK

A substantial research field has grown to analyse the micro-timing strategies in different musical styles. This has included work on the Viennese waltz (from Bengtsson and Gabrielsson's landmark 1977 work to Yang's 2022 analysis of 'The Blue Danube' [6, 7]), jembe music from Mali (notably through Rainer Polak's career-long focus on this repertoire, [8–10]), and a recent surge of work on Afro-Cuban and Latin musics (see, for instance, [11, 12]).

Creating music with synthetic micro-timing has also been explored as part of the broad field of MIR, but has concentrated on attempting to model human-like expressive timing [13]. For example, Flossman et al. use probabilistic models for expressive performance rendering [14, 15]. There has been very little academic work on integrating style-specific timing analyses into computational settings. The closest examples are in the commercial sphere: Ableton Live, for instance, features "grooves" which shift MIDI events from quantised positions according to micro-timing styles, including a probabilistic element and the option to create new "grooves" from any human performances (via MIDI).

Musical live coding is a way of creating and performing music by writing and modifying code in real time. While there may or may not be pre-made materials, live manipulation of the material is a given. Given the inherent "liveness" of live coding, [3] it is arguably an ideal part of the computer music pantheon to integrate human micro-timing. Yet most live coding languages lack not only micro-timing functionality, but even a full representation of musical metre, typically encoding only events in time, or at most an anaemic representation for beats and/or time signatures. For example, the commercial Max/MSP language uses its `transport` object to allow access to bar and beat numbers for the current time signature. An exception is McLean's open-source Tidal Cycles which uses a cyclic notion of time that can be subdivided to achieve more complex hierarchies [17].

In summary, although there has been much research into the analysis of micro-timing in different musical styles, and the application of expressive timing to computer-generated music, we still lack implementations of style-specific micro-timing in most computer-music software, and even foundational notions of musical metre in most live coding environments. This project seeks to address those issues through an implementation of both metre and style-specific micro-timing for Sonic Pi: a popular live coding language and IDE designed to support a range of creative possibilities while being simple enough to use as an educational tool for use in schools [18]. [4] We aim to improve not only how "life-like" the generated music sounds in general (and thus, arguably the "liveness" of that live coding), but also to do this in a style-specific way. In this paper we report on implementation of two case studies as well as a general framework for integrating further styles.

[3] This is discussed in Chapter 5 of [16], for instance.

[4] Sonic Pi's domain-specific language is written in Ruby and uses the SuperCollider sound synthesis server to produce sounds [19].

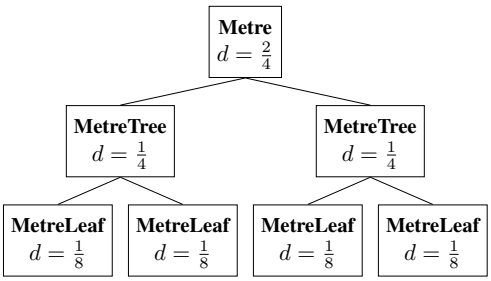


Figure 1: An example of how MetreTree and MetreLeaf objects are nested to construct a metrical hierarchy for $\frac{2}{4}$. The total duration d of each node is also displayed, and the duration of a parent node is the sum of the durations of its children [22, 23].

3. IMPLEMENTING METRE

This section describes the model of metre we have implemented for Sonic Pi. We argue that this is useful for a range of applications including (but not limited to) use as a basis for micro-timing as described below (§4). As discussed, metre is a very widespread phenomenon in general but specific aspects differ. We can broadly distinguish here between the specifically *hierarchical aspects* (important for some styles but not all) and the more general notion of *categorical positions* in a metrical cycle (much more widespread, and axiomatic for the kind of micro-timing systems discussed and implemented here).

3.1 Modelling Metrical Hierarchy with Trees

A favoured method for encoding metrical hierarchy in a data structure is through trees. See Forth [20] for a detailed mathematical treatment of trees used in this context and the `music21` Python library [21] for a popular implemenation. Our approach shares some high-level ideas with `music21` (and indeed Forth, and others), but differs in the specific implementation.

The tree structure is implemented by the `MetreLeaf` and `MetreTree` classes. Figure 1 shows the default tree structure formed by these objects and their durations for a Western $\frac{2}{4}$ time signature. Note how the duration of a parent node is the sum of the durations of its children.

To model tree data structures of any depth with a succinct, finite representation, we follow the Western notational assumption of diving each MetreLeaf into two equal parts to get the next level where not specified otherwise. [5] Users can specify the full depth of a tree as necessary against this assumption.

3.2 The MetreLeaf Class

A MetreLeaf object is the leaf node of the metrical tree structure. It has an instance variable `fraction` which represents the duration of the MetreLeaf as a fraction of a whole note. For example, a leaf node with the duration of one quarter note will have the value $\frac{1}{4}$.

[5] See [24] for discussion of this point, of metrical "well-formedness", and the notion of "binary", "ternary" and wider metrical structures.

The class contains a `subdivide()` method, which divides the MetreLeaf by two a given number of times, s. It returns a new MetreTree with 2^s MetreLeaf children, each of value $f/2^s$ where f is the fraction of the original MetreLeaf.

3.3 The MetreTree Class

A MetreTree object represents the hierarchical tree or subtree of a metre. The instance variable `sequence` is an ordered list representing this node's children and contains any combination of MetreLeaf objects and other MetreTree objects. For example, the hierarchy in Figure 1 could also be written in list form as:

$$\left[\left[\frac{1}{8}, \frac{1}{8}\right], \left[\frac{1}{8}, \frac{1}{8}\right]\right]$$

Each list is a MetreTree, and each fraction is a MetreLeaf. The MetreTree class contains several methods for manipulating and extracting information from the metrical hierarchy it represents. The two most important of these are explained in more detail below.

3.3.1 Getting Metrical Levels

Metrical level refers to the depth level of a metrical hierarchy. Here, we base our representation on the *beat level*, which is divided to get *division levels*, and grouped to get *grouping levels*.[6] The `get_level()` method allows a user to "flatten" the tree structure to a specified depth, accessing the sequence of events at a given metrical level.

For flattening to a division level ($l > 0$) or to the beat level ($l = 0$), we perform a recursive depth-first search on the tree. For each child in the sequence list, if it is a MetreTree, the method is recursively called until the base case of $l = 0$ is reached. At this point, all the children of that node are combined into one MetreLeaf equal to the sum of their durations. If the child is instead a MetreLeaf, it is subdivided l times to reach the desired metrical level.

For a grouping level ($l > 0$), we find an estimate of the structure of higher metrical levels by clustering nodes together. It is an estimate because this information is not in the MetreTree's representation of the metre, so is just one possibility for the higher structure. The algorithm recursively clusters nodes until the desired metrical level l is reached. The number of nodes combined in each cluster is determined by the smallest prime factor of the number of nodes at the level below. For example, if level $l + 1$ has four nodes, they will be clustered in groups of two. If it has nine nodes, they will be clustered in groups of three.

Some examples of the output of the flattened tree for the following complex hierarchy are shown in Table 1:

$$\left[\left[\frac{1}{8}, \frac{1}{8}\right], \left[\frac{1}{16}, \frac{3}{16}\right], \frac{1}{8}, \left[\frac{1}{4}, \left[\frac{5}{16}, \frac{3}{16}\right]\right]\right]$$

[6] Centring the beat level in this way reflects the psychology of metre better than alternative "top down" and "bottom up" approaches.

l	`get_level`(l)
-2	$\left[\frac{11}{8}\right]$
-1	$\left[\frac{1}{2}, \frac{7}{8}\right]$
0	$\left[\frac{1}{4}, \frac{1}{4}, \frac{1}{8}, \frac{3}{4}\right]$
1	$\left[\frac{1}{8}, \frac{1}{8}, \frac{1}{16}, \frac{3}{16}, \frac{1}{16}, \frac{1}{16}, \frac{1}{4}, \frac{1}{2}\right]$

Table 1: Examples of the output of `get_level`(l) at different metrical levels l for an example hierarchy. Note how level $l = -1$ is formed by the clustering of level $l = 0$.

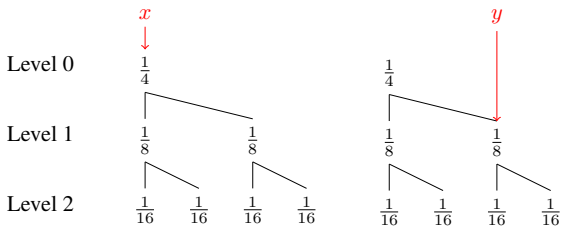


Figure 2: An example metrical hierarchy for $\frac{2}{4}$ showing those metrical events at each level which coincide with offsets x and y.

3.3.2 Getting Exact Metrical Events

We define an *offset* as a position in the metric cycle represented by the quarter length duration to have elapsed since the beginning of the cycle.[7] The `metrical_level_indices()` method of the MetreTree class finds any metrical events occurring at a given offset, and returns their index.

Consider the example shown in Figure 2. Offset x occurs on the first event of all three levels, so the function would return $L_0(x) = L_1(x) = L_2(x) = 0$, where $L_l(x)$ is the index of an event at level l that offset x occurs on. Offset y occurs only on the last event of Level 1 and the second-to-last event of Level 2, so the function would return $L_1(y) = 3, L_2(y) = 6$.

This method is important because it is used later to determine the "categorical" position to link an event to and, through that, which micro-timing probability distribution to apply.

3.4 Bar Class

The Bar class is a representation of a single metrical cycle,[8] and each instance of it has an associated metre. A Bar object is responsible for:

[7] "Quarter length" is a semi-standard measurement for a length of time in symbolic values where the unit length is one "quarter note" duration (UK: "crotchet").

[8] A more precise definition would account for *hypermetre* where bars occur at the beat level [25], but the simple definition is sufficient for our purposes. Note that "bar" is the UK English equivalent of "measure" (USA).

```
play :C4
sleep(1)
play :E4
sleep(1)
play :G4
sleep(0.5)
play :E4
sleep(0.5)
play :C4
```

(a) Old (above left)

```
use_metre '4/4'

bar do
    add_note :C4, 0, 1
    add_note :E4, 0, 1
    add_note :G4, 1, 1
    add_note :E4, 1, 1
    add_note :C4, 0, 1
end
```

(b) New (above right)

(c) Western musical notation:

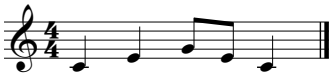


Figure 3: A bar of music represented by (a) the original Sonic Pi syntax, (b) our new metre commands, and (c) traditional Western music notation. Note how the original Sonic Pi syntax loses information about the metre. The second and third arguments to `add_note` are the metrical level and duration (l and d in §3.4).

- Keeping track of playback position during the cycle.
- Converting a note length given as a metrical level and a duration into a quarter length.
- Checking if a note or rest fits in the remaining time in the cycle, and updating the bar's playback position accordingly.

A note's length is specified by a metrical level and a duration, where the duration is in the units of an event at the specified metrical level and acts as a multiplier. For example, if a note's length is defined as $(l, d) = (0, 3)$, its unit length is the duration of an event at level $l = 0$, and it lasts for $d = 3$ of these units.

The `add_note()` method checks if a note fits into the bar's remaining time; if it cannot, an exception is raised. This ensures the total ("actual") duration of the bar matches its metre's ("nominal") duration.

3.5 Playing Music

This framework for musical metre enabled the creation of new Sonic Pi commands. Figure 3 shows a comparison between the original Sonic Pi commands, our alternative commands, and traditional Western music notation.

There are two main commands for metre. The first is `use_metre(`m`)`, which changes the current thread's metre to m (using a thread-local variable). The second is `bar do ... end`, which creates a new Bar object, stores this to a thread-local variable, then executes a block of user code.

A user can use `add_note` to play a note on the current synthesiser. This works by first getting the current Bar object from the thread-local variables and calling the Bar's `add_note()` method to check if the note will fit in the bar. It then passes the note pitch to Sonic Pi's `play` function which creates the sound, and finally applies `sleep` for the remaining duration of the note.

4. MICRO-TIMING

4.1 Storing Micro-Timing Information

In order to add micro-timing functionality to our implementation, we first needed a way of representing and storing the micro-timing information for different musical styles. We implement this by storing each event in the metrical cycle, the theoretical (isochronous) position of that event in the cycle (e.g., 1), and the typical displacement of the from this position (the μ of the micro-timing, e.g., +0.004). Actual event occurrence is modelled by normal probability distribution around these μ values.

Samples can then be drawn from these distributions using the Box-Muller transform [26] on uniform random samples from Sonic Pi's random number generator. Sonic Pi's generator produces a deterministic, repeatable sequence of pseudorandom numbers, which means the output of a Sonic Pi program sounds the same each time it is run [27].

4.2 Applying Micro-Timing

When a user sets a metre with the `use_metre` command, they can optionally specify a style as well. This causes all music played with that metre to use the micro-timing of the chosen style.

At the start of each new bar, the `Metre` object samples new values from the `Style`'s probability distributions. When a note is played inside the bar, the `add_note` command requests the timing shift that should be applied to the note from the Metre. To calculate this, the Metre object calls its `metrical_level_indices()` method to determine which timing values from each level to use. The individual timing contributions of each metrical level are summed to produce an overall timing shift for the note. A positive value means the note should be played slightly late; a negative value means slightly early. This is returned to `add_note` which then uses Sonic Pi's `time_warp` function to adjust the timing of the call to `play`.

For example, if the sampled timings, T_l, for each level, l, are:

$$T_0 = [0, 0.1]$$
$$T_1 = [0.03, 0, 0, -0.02]$$

and the metrical level indices, L_l, for each level, l, are:

$$L_0 = 1$$
$$L_1 = 3$$

then the timing shift, t, would be calculated by:

$$\begin{aligned} t &= \sum_{i \in T.\text{keys}} T_i[L_i] \\ &= T_0[L_0] + T_1[L_1] \\ &= 0.1 + (-0.02) \\ &= 0.08 \end{aligned}$$

Therefore, the note will be played 0.08 quarter lengths after the reference value.

5. CASE STUDIES

Creating music with realistic micro-timing using the implementation we have described requires a set of probability distributions which accurately characterise a style of music. Clearly this is best implemented with data derived from real-life examples of the musical style in question. This project uses two different styles of music as contrasting case studies for evaluation: jembe (or "djembe") drum music from Mali and Viennese waltz music. These two styles both have robust, well-known micro-timing characteristics. The distributions derived here form the "preset" styles included in our Sonic Pi plugin.

5.1 Jembe Data Analysis

Jembe is a style of West African music involving a small ensemble of drummers (typically 3–4). It provides an ideal case study for our purposes because it has a highly consistent micro-timing strategy [8]. Malian drummers have been shown to exhibit some of the most consistent timing (lowest levels of variability) between performers in the world [28].

Moreover, jembe music is relatively constrained in terms of its pitch, timbre, and number of instruments. This also helps by enabling a clear focus on timing. Extensive research into the micro-timing of jembe music has included the release of high-quality datasets of processed live recordings [8–10].

The first dataset is from Jacoby et al. [10] and consists of 11 processed recordings of a piece called 'Suku', which is a very commonly played piece in this style. The second dataset is from the "Interpersonal Entrainment in Music Performance" (IEMP) Data Collection [29, 30]. This consists of 15 recordings across three different pieces: 'Manjanin', 'Maraka', and 'Woloso'. Both datasets here use recordings made by Rainer Polak in Mali. The datasets supply the following information:

- Onset of the drum stroke in seconds since the start;
- Phase: beats since the start of the *current cycle*;
- Cycle (bar) number: a natural number count;
- Categorical metrical position within the cycle associated with this event (integer, 0–11).

5.1.1 Micro-Timing Estimation

The pieces of jembe music in the dataset use a metre with four beats, [9] each of which divides into three, for a total of 12 metrical events at the first division level (similar to $\frac{12}{8}$ in Western classical notation). It is at this level, referred to as the "pulse", that the main micro-timing occurs.

Recall that the probability distributions described in §4.1 store the displacement of each event. We calculate this from the phase given by the datasets with the following equation:

$$\text{displacement} = \frac{(\text{phase} \times \text{beat division}) - \text{metric position}}{2}$$

[9] See Polak [8] for an ethnographically sensitive discussion of the extent to which metre applies in this context.

The phase is multiplied by the beat division (in this case, 3) to convert it into pulse units. The metric position at the pulse level is subtracted to get the displacement. The final division by 2 converts the displacement into quarter lengths (because each pulse unit is an eighth length).

For example, if an onset has metric position $= 6$ and phase $= 2.01$, the displacement would be calculated by:

$$\text{displacement} = \frac{(2.01 \times 3) - 6}{2} = 0.015 \text{ quarter lengths.}$$

Once the displacement has been calculated for each drum stroke, we were then able to estimate the distribution of displacements for each of the twelve metric locations using maximum likelihood estimation (MLE).

5.1.2 Tempo Estimation

Generating a synthetic piece of jembe music requires analysis of other musical features as well as the micro-timing to sound realistic. One of these is the *tempo*, which in jembe pieces of music typically increases substantially over the duration of the performance [10], with the last 15 seconds or so showing the tempo increasing at a much faster rate.

The *inter-beat interval* (IBI) is defined as the time between two consecutive beats in a piece of music, from which the instantaneous tempo can be calculated [31]. A moving average can be applied to the instantaneous tempo to obtain an estimate of the global tempo.

For the jembe data, we first filtered all the onsets to include just those played by Jembe 2 (because it plays on every beat as discussed in [10]), then filtered these to only consider onsets on the beats. We then calculated the inter-beat interval in bpm and applied a moving average with window size 10 to smooth the tempo estimate.

Inspection of the smoothed tempo graphs (§6.2) showed a logarithmic trend for the first ~95% of the piece. A sharper increase follows this which was modelled by a quadratic curve. To fit curves to the data, we used the `optimize.curve_fit` function from the SciPy Python library, which uses a non-linear least squares method [32]. The parameters estimated by the curve fitting are then used in Sonic Pi to control the tempo of a synthetic jembe piece during playback.

5.2 Waltz Data Analysis

The Viennese waltz is a style of fast waltz notated in $\frac{3}{4}$ (but often counted in 1), originally intended for ballroom dancing, and now often performed in concerts by Western classical orchestras.

The Viennese waltz provides a useful comparison to Malian jembe in evaluating this project's micro-timing implementation. The fast three beats ($\frac{3}{4}$) and typical hypermetrical grouping in 2s and 4s make that *metrical structure* somewhat similar to that of jembe music, but with a very different micro-timing profile. Distinctive micro-timing can be observed on (at least) the beat level, where it has a characteristic short-long-medium pattern [6, 33].

At the time this work was carried out, the micro-timing in Viennese waltz had not been studied in as much detail or

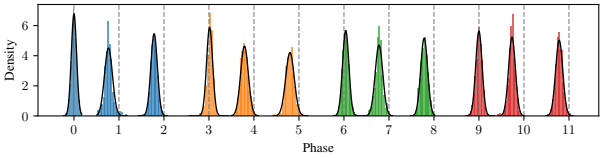


Figure 4: A histogram plot of the positions of each pulse within the cycle for Suku. Dashed lines show the isochronous division of the cycle for reference. Black curves show the PDF of the MLE-derived probability distributions and the colours distinguish to the four beat.

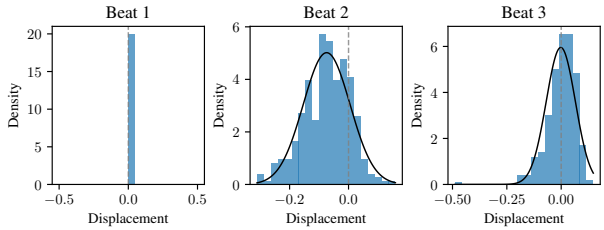


Figure 5: A histogram plot of the displacement of the second and third beats for the waltz dataset. Dashed lines show the metrical grid. Black curves show the PDF of the MLE-derived probability distributions.

as recently as jembe and there were no existing datasets of Viennese waltz performances with micro-timing. Therefore, we constructed a new dataset comprising of 30-second samples from seven waltz recordings performed by the Vienna Philharmonic Orchestra, all of which have noticeable and statistically significant micro-timing.

We then performed automatic beat tracking on this dataset using the libfmp Python library [34] (a dynamic programming approach introduced by Müller [35]), with some small manual corrections. Since the beat level is where the primary micro-timing in the Viennese waltz occurs, no additional onset detection was necessary.

Calculating the micro-timing displacement of each beat from onset times alone involves first identifying the start and end of each cycle, estimating the onset of each beat as if they were isochronous, then finding the difference between this and the actual onset to get the displacement. Once the displacements were derived, maximum likelihood estimation was again used to fit the probability distributions as discussed above for the jembe case.

6. RESULTS

6.1 Micro-Timing Estimation

Figure 4 shows the results of the micro-timing estimation for one of the jembe pieces in the datasets: "Suku". The histograms show the positions within the cycle where the 12 pulses occurred (phase). The dashed lines indicate where the event would occur if they were isochronous: from this the existence of the micro-timing can be seen clearly by the positions of the second and third pulses in each beat. By examining the positions of the histograms, we can see that the length of each pulse follows a short-medium-long pattern (SML), which is consistent across each beat. Also shown are the probability density functions (PDF) of the maximum-likelihood estimated normal distributions. The plots/data for each beats showed the same pattern which also corresponds to other jembe pieces and matches results previously reported by Polak [8].

Figure 5 shows the results for the waltz dataset. The calculations use the first beat as the definition for the start of the cycle, so every Beat 1 has a displacement of 0. The early onset of the second beat can be clearly seen in the plot ($\mu = -0.0743$, $\sigma = 0.0795$). A one-sample t-test confirms the micro-timing as significant ($t = -16.5$, $p = 0.000$). Beat 3 shows no significant deviation from a 3-part isochronous division of the cycle, so the overall pattern identified is the short-long-medium (SLM) discussed elsewhere [7].

6.2 Jembe Tempo Estimation

The results of the jembe tempo estimation showed the increase in tempo throughout the piece that is characteristic of Malian jembe music. The more dramatic speedup at the end is also reflected in this data – this is why we fit two different curves to the data. For example, in "Suku", the tempo starts at around 135 bpm at the beginning of the piece and ends at around 175 bpm. The tempo results match those found by Jacoby et al. [36], and each jembe piece showed the same trend.

7. CONCLUSION

In this project, we have investigated and implemented probabilistic style-specific micro-timing in a musical live coding language. To do this, we extended the Sonic Pi language with implementations of both musical metre and micro-timing, and we performed data analysis on recordings of music from two case study styles to generate music with realistic micro-timing.

In further work (not reported here but available on request), we conducted a user study to assess how "realistic" our synthesised micro-timing sounded for each of the case study styles. Significant results were obtained for the Viennese waltz, however participants struggled more with the jembe, likely due to their unfamiliarity with the style. Future work could conduct a new user study with expert participants, as in Neuhoff [37].

Naturally, other future work could focus on additional styles with well-documented micro-timing, such as jazz swing rhythms [38], candombe drum ensembles from Uruguay [11, 39], and Brazilian samba music [11, 40]. Likewise, larger datasets for the styles reported here would enable more accurate distributions – two notable datasets of Viennese waltz recordings have been released even since the work reported here: Weigl et al. [41] and Yang [7].

As for software functionality, we imagine extensions including new *variable gridline* positions in DAWs, and additional controls within Sonic Pi to dynamically adjust the *"strength" of the micro-timing*.

8. ACKNOWLEDGEMENTS

Thanks to Rainer Polak for his invaluable advice and feedback on this project, for his years of research on jembe music, and for releasing the data that made that case study possible. Thanks indeed to all who provided feedback on this project. This research was conducted in the context of author MG's time as an Affiliated Lecturer at the Department of Computer Science and Technology, University of Cambridge. We thank Alan Blackwell and others for their role in organising and supporting this.

9. REFERENCES

[1] J. London, "Rhythm. Grove Music Online," 2001. [Online]. Available: https://www.oxfordmusiconline.com/grovemusic/view/10.1093/gmo/9781561592630.001.0001/omo-9781561592630-e-0000045963

[2] ——, *Hearing in time: Psychological aspects of musical meter*, 2nd ed. Oxford University Press, 2012.

[3] R. Cohn, "Meter," in *The Oxford Handbook of Critical Concepts in Music Theory*. Oxford University Press, 01 2020. [Online]. Available: https://doi.org/10.1093/oxfordhb/9780190454746.013.9

[4] C. E. Seashore, "The psychology of music. ix. a performance score with phrasing score for the violin," *Music Educators Journal*, vol. 24, no. 1, pp. 28–29, 1937. [Online]. Available: http://www.jstor.org/stable/3385490

[5] E. F. Clarke, "Levels of structure in the organization of musical time," *Contemporary Music Review*, vol. 2, no. 1, pp. 211–238, 1987.

[6] I. Bengtsson and A. Gabrielsson, "Rhythm research in Uppsala," *Music, Room and Acoustics*, 1977.

[7] J. Yang, "Viennese style in viennese waltzes: An empirical study of timing in the recordings of the blue danube," *Musicologica Austriaca: Journal for Austrian Music Studies*, no. 2022, 2022.

[8] R. Polak, "Rhythmic feel as meter: Non-isochronous beat subdivision in jembe music from Mali," *Music Theory Online*, vol. 16, no. 4, 2010.

[9] J. London, R. Polak, and N. Jacoby, "Rhythm histograms and musical meter: A corpus study of Malian percussion music," *Psychonomic bulletin & review*, vol. 24, no. 2, pp. 474–480, 2017.

[10] N. Jacoby, R. Polak, and J. London, "Extreme precision in rhythmic interaction is enabled by role-optimized sensorimotor coupling: analysis and modelling of West African drum ensemble music," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 376, no. 1835, p. 20200331, 2021.

[11] M. Fuentes, L. S. Maia, M. Rocamora, L. W. Biscainho, H. C. Crayencour, S. Essid, and J. P. Bello, "Tracking beats and microtiming in Afro-Latin American music using conditional random fields and deep learning," in *Proceedings of the 20th Conference of the International Society for Music Information Retrieval*, 2019, pp. 251–258.

[12] M. E. P. Davies, M. Fuentes, J. Fonseca, L. Aly, M. Jerónimo, and F. B. Baraldi, "Moving in time: Computational analysis of microtiming in maracatu de baque solto," in *Proceedings of the 21st International Society for Music Information Retrieval Conference*, 2020, pp. 795–802.

[13] J. Bilmes, "Timing is of the essence: Perceptual and computational techniques for representing, learning, and reproducing expressive timing in percussive rhythm," Ph.D. dissertation, Massachusetts Institute of Technology, 1993.

[14] S. Flossmann, M. Grachten, and G. Widmer, "Expressive performance rendering: Introducing performance context," *Proceedings of the 6th Sound and Music Computing Conference (SMC)*, pp. 155–160, 2009.

[15] ——, *Expressive Performance Rendering with Probabilistic Models*. Springer London, 2013, pp. 75–98.

[16] A. F. Blackwell, E. Cocker, G. Cox, A. McLean, and T. Magnusson, *Live coding: a user's manual*. MIT Press, 2022.

[17] A. McLean and G. Wiggins, "Tidal–pattern language for the live coding of music," in *Proceedings of the 7th sound and music computing conference*, 2010, pp. 331–334.

[18] S. Aaron and A. F. Blackwell, "From Sonic Pi to Overtone: Creative musical experiences with domain-specific and functional languages," in *Proceedings of the first ACM SIGPLAN workshop on Functional art, music, modeling & design*, 2013, pp. 35–46.

[19] J. McCartney, "Rethinking the computer music language: Super collider," *Computer Music Journal*, vol. 26, no. 4, pp. 61–68, 2002. [Online]. Available: http://www.jstor.org/stable/3681770

[20] J. Forth, "Cognitively-motivated geometric methods of pattern discovery and models of similarity in music," Ph.D. dissertation, Goldsmiths, University of London, 2012.

[21] C. Ariza and M. S. Cuthbert, "Modeling beats, accents, beams, and time signatures hierarchically with music21 meter objects," in *Proceedings of the 2010 International Computer Music Conference, ICMC 2010*. Michigan Publishing, 2010. [Online]. Available: http://hdl.handle.net/2027/spo.bbp2372.2010.043

[22] H. C. Longuet-Higgins and C. S. Lee, "The rhythmic interpretation of monophonic music," *Music Perception: An Interdisciplinary Journal*, vol. 1, no. 4, pp. 424–441, 1984. [Online]. Available: http://www.jstor.org/stable/40285271

[23] G. Sioros, M. E. P. Davies, 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.

[24] M. Gotham, "The metre metrics: Characterising (dis)similarity among metrical structures," Ph.D. dissertation, University of Cambridge, 2015.

[25] J. Neal, "Songwriter's signature, artist's imprint: The metric structure of a country song," in *Country Music Annual 2000*, C. K. Wolfe and J. E. Akenson, Eds. Lexington, KY: University Press of Kentucky, 2000, pp. 112–140.

[26] G. E. P. Box and M. E. Muller, "A note on the generation of random normal deviates," *The Annals of Mathematical Statistics*, vol. 29, no. 2, pp. 610–611, 1958.

[27] S. Aaron, "Sonic Pi — reliable randomisation for performances," in *2016 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, 2016, pp. 242–243.

[28] M. Clayton, K. Jakubowski, T. Eerola, P. E. Keller, A. Camurri, G. Volpe, and P. Alborno, "Interpersonal entrainment in music performance: theory, method, and model," *Music Perception: An Interdisciplinary Journal*, vol. 38, no. 2, pp. 136–194, 2020.

[29] R. Polak, S. Tarsitani, and M. Clayton, "IEMP Malian jembe," 7 2020.

[30] M. Clayton, S. Tarsitani, R. Jankowsky, L. Jure, L. Leante, R. Polak, A. Poole, M. Rocamora, P. Alborno, A. Camurri *et al.*, "The interpersonal entrainment in music performance data collection," *Empirical Musicology Review*, vol. 16, no. 1, pp. 65–84, 2021.

[31] S. Dixon, "Automatic extraction of tempo and beat from expressive performances," *Journal of New Music Research*, vol. 30, no. 1, pp. 39–58, 2001.

[32] J. J. Moré, "The Levenberg-Marquardt algorithm: implementation and theory," in *Numerical analysis*. Springer, 1977, pp. 105–116. [Online]. Available: https://www.osti.gov/biblio/7256021

[33] I. Bengtsson, "Empirische rhythmusforschung in Uppsala," *Hamburger Jahrbuch für Musikwissenschaft*, vol. 1, pp. 195–219, 1974.

[34] M. Müller and F. Zalkow, "libfmp: A Python package for fundamentals of music processing," *Journal of Open Source Software*, vol. 6, no. 63, p. 3326, 2021.

[35] M. Müller, *Fundamentals of music processing: Using Python and Jupyter notebooks*, 2nd ed. Springer, 2021.

[36] N. Jacoby, R. Polak, and J. London, "Supplementary material from extreme precision in rhythmic interaction is enabled by role-optimized sensorimotor coupling: analysis and modelling of West African drum ensemble music," 7 2021.

[37] H. Neuhoff, R. Polak, and T. Fischinger, "Perception and evaluation of timing patterns in drum ensemble music from Mali," *Music Perception: An Interdisciplinary Journal*, vol. 34, no. 4, pp. 438–451, 2017.

[38] C. Dittmar, M. Pfleiderer, S. Balke, and M. Müller, "A swingogram representation for tracking micro-rhythmic variation in jazz performances," *Journal of New Music Research*, vol. 47, no. 2, pp. 97–113, 2018.

[39] L. Jure and M. Rocamora, "Microtiming in the rhythmic structure of Candombe drumming patterns," in *4th Int. Conf. on Analytical Approaches to World Music (AAWM)*, 6 2016.

[40] L. Naveda, F. Gouyon, C. Guedes, and M. Leman, "Microtiming patterns and interactions with musical properties in samba music," *Journal of New Music Research*, vol. 40, no. 3, pp. 225–238, 2011.

[41] D. M. Weigl, C. VanderHart, M. Pescoller, D. Rammler, M. Grassl, F. Trümpi, and W. Goebl, "The vienna philharmonic orchestra's new year's concerts: Building a fair data corpus for musicology," in *Proceedings of the 9th International Conference on Digital Libraries for Musicology*. Association for Computing Machinery, 2022, pp. 36–40.

A FEW-SHOT NEURAL APPROACH FOR LAYOUT ANALYSIS OF MUSIC SCORE IMAGES

Francisco J. Castellanos[1] **Antonio Javier Gallego**[1] **Ichiro Fujinaga**[2]
[1] University Institute for Computing Research, University of Alicante, Spain
[2] Schulich School of Music, McGill University, Montreal, Canada
{fcastellanos, jgallego}@dlsi.ua.es, ichiro.fujinaga@mcgill.ca

ABSTRACT

Optical Music Recognition (OMR) is a well-established research field focused on the task of reading musical notation from images of music scores. In the standard OMR workflow, layout analysis is a critical component for identifying relevant parts of the image, such as staff lines, text, or notes. State-of-the-art approaches to this task are based on machine learning, which entails having to label a training corpus, an error-prone, laborious, and expensive task that must be performed by experts. In this paper, we propose a novel few-shot strategy for building robust models by utilizing only partial annotations, therefore requiring minimal human effort. Specifically, we introduce a masking layer and an oversampling technique to train models using a small set of annotated patches from the training images. Our proposal enables achieving high performance even with scarce training data, as demonstrated by experiments on four benchmark datasets. The results indicate that this approach achieves performance values comparable to models trained with a fully annotated corpus, but, in this case, requiring the annotation of only between 20% and 39% of this data.

1. INTRODUCTION

Optical Music Recognition (OMR) is a research field dedicated to developing computational methods for transcribing musical notation from document images into digital formats [1]. While this task could be accomplished manually, the vast number and heterogeneity of music documents make this approach tedious, costly, and error-prone. The development of OMR systems has the potential to enhance music heritage accessibility and preservation, as well as enable the application of analysis algorithms to increase knowledge about this cultural legacy.

OMR typically follows a sequential workflow, which divides the transcription process into simpler tasks. The initial task is called Document Image Analysis (DIA), which is itself a research field that studies how to obtain a segmented version of the image by isolating the different layers of interest, such as staves, lyrics, instructions, ornaments, etc [2]. In the literature, multiple strategies can be found to perform this *layout analysis*, ranging from heuristic approaches that exploit specific features of the images to deep learning techniques. Although heuristic approaches achieve high performance in controlled scenarios, these solutions are poorly generalizable. To obtain better and generalizable results, the current trend is to rely on machine learning and, more specifically, on neural network architectures [3].

The application of deep learning in layout analysis has been extensively studied, as evidenced by several state-of-the-art works [4, 5]. However, a major drawback of these methods is the requirement for a large amount of annotated data for their training. This is particularly problematic for the layout analysis of music scores since their high variability in appearance and styles makes necessary the annotation of each new application domain in order to train robust models. Despite the importance of this issue, it has been overlooked in the OMR literature, with domain adaptation being the only explored solution [6]. Nevertheless, this technique also requires full annotations (even if it is from a different domain) and the performance obtained is not good or robust enough, which also makes it an impractical solution.

In this work, we propose a novel few-shot strategy for building robust models for layout analysis by utilizing only partial annotations, therefore requiring minimal human effort. Specifically, we introduce a masking layer and an oversampling technique to train models using a small set of annotated patches from the training images. Our approach aims to drastically reduce the manual workload without compromising performance, making it of particular interest to real-world applications. Experiments on four benchmark datasets indicate that this approach achieves performance comparable to models trained on a fully annotated corpus—but requiring the annotation of only between 20% and 39% of this data depending on the layer—thus making it a highly efficient and effective strategy.

2. RELATED WORK

Traditional OMR workflows relied on a combination of heuristic strategies to perform pixel-wise layout analysis and classify each pixel of the image according to a set

© F. J. Castellanos, A. J. Gallego, and I. Fujinaga. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** F. J. Castellanos, A. J. Gallego, and I. Fujinaga, "A Few-shot Neural Approach for Layout Analysis of Music Score Images", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.*, Milan, Italy, 2023.

of categories [2]. A binarization process was commonly applied to simplify the complexity of the image to detect the ink pixels, either using generic approaches [7, 8] or other particular ones proposed for the musical context [9, 10]. The recognition and isolation of the staff and the lyrics were then carried out using also heuristic techniques [11, 12]. From these detected staves, the musical symbols were finally processed, sometimes carrying out another step to remove the staff lines, as can be seen in the review by Dalitz et al. [13] or in more recent works [14–16].

More recently, all these steps were combined by means of machine learning techniques. Calvo-Zaragoza et al. [17] proposed a Convolutional Neural Network (CNN) to directly classify each pixel of the image—performing a pixel-wise layout analysis—which was later improved using a U-net-like architecture—referred to as Selectional Auto-Encoder (SAE)—to more efficiently classify the image by patches [18]. This later work, on which our proposal is based, trained a set of SAE specialized in the detection of each layer of information—staff lines, notes, text, or background.

The main challenge with layout analysis approaches that rely on supervised learning is the large amount of annotated data needed to train the models [19, 20]. This requires the annotation at the pixel level of a reference set of images, which has to be done by hand, so it is not a scalable solution given the high level of detail of these annotations and heterogeneity in music documents. In addition, when this constraint cannot be fulfilled, these learning-based architectures fail to converge to obtain a suitable model for the task at hand.

In the literature, we can find different proposals that seek to alleviate this issue [21], two of the most common being the use of regularization strategies [22] and data augmentation processes [23]. We can also find more specific proposals for cases of remarkable data scarcity, i.e., with a considerably fewer number of annotated training samples. These scenarios are known as *few-shot learning* [24] and typically employ specific neural architectures to estimate the similarity of the data [25]. Some of the most typical examples of these techniques are Siamese Neural Networks [26], Matching Networks [27], Prototypical Networks [28], and Relation Networks [29]. For a comprehensive review of these strategies, the reader is referred to the work by Jadon [30].

Our proposal follows a few-shot learning approach, but instead of using a specific few-shot architecture, a state-of-the-art layout analysis model—the previously described SAE network—is modified to integrate a masking layer that enables training with very little data. This layer is complemented by an oversampling proposal used during the training process to draw samples at random positions around the chunks with annotated data. A mask is applied to these pieces and used by the added layer to avoid processing the non-annotated parts, which will randomly appear in different positions in each iteration, thus forcing the architecture to generalize the learned weights.

In the related literature, masks have been used for different purposes. For example, Medhat et al. [31] proposed the use of binary masks for sound classification to filter out certain frequency bands. It has also been explored for image classification, specifically, Suresh et al. [32] studied the use of masks as a pre-processing task to filter the background of images with hand gestures, making the model focus only on the gestures to be classified. However, as far as we know, masks have not been used either in binarization tasks or for few-shot learning cases, so that the model does not use the unlabeled areas.

3. METHODOLOGY

Our approach aims to build a robust few-shot learning model for layout analysis of music score images that classifies each pixel of an input image into one of the following categories: `staff`, `notes`, `text`, and `background`. In our context, the few-shot scenario can be represented as a manual annotation of a limited number n of portions or patches from a set of images $\mathcal{I}$, with $n \ll N$, where N is the total number of possible patches that could be sequentially extracted without overlapping from $\mathcal{I}$. Therefore, when n is small, less human effort and cost are required to annotate the training set.

Note that labeling only part of the image makes the rest of it uninformative, even if there are ink pixels. In a typical training process, only the annotated patches would be used. However, when the amount of data is limited, this would lead to overfitting of the model. Although data augmentation may help mitigate this problem, in a few-shot learning scenario, it is not very useful due to the little information to be altered.

Our proposal introduces a novel approach to extract a larger—and more varied—number of samples from the scarce labeled information. Specifically, it is proposed to extract random patches around the annotated areas—keeping a minimum $\lambda\%$ of labeled information—to obtain more varied samples, thus generating variations in the position of the elements and their labeling. Since some parts of the extracted patches will fall outside the annotated area, it is proposed to mark those parts with a special label (-1) so that they are not used during training. This approach allows us to control the number of samples to be drawn from the images and get enough variability in the data to train the model, as we will demonstrate in the experiments.

Formally, let $\mathcal{X} \in \mathbb{R}^{w \times h}$ be a collection of patches of size $w \times h$ drawn from the input set of images $\mathcal{I}$, and $\mathcal{Y} \in \{0,1\}^{w \times h}$ be the corresponding pixel-level annotation matrices extracted from the annotation set $\mathcal{L}^l$ for the layer to be processed $l \in \{\texttt{staff}, \texttt{notes}, \texttt{text}, \texttt{background}\}$, where 1 is used to label the ink of that layer and 0 the rest, either background or information from another layer. Additionally, let $\mathcal{S} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i) : \boldsymbol{x}_i \in \mathcal{X}, \boldsymbol{y}_i \in \mathcal{Y}\}_{i=1}^{|\mathcal{S}|}$ represent an annotated collection of data where each datum $\boldsymbol{x}_i$ is related to label $\boldsymbol{y}_i$ by an underlying function $f^l : \mathcal{X} \rightarrow \mathcal{Y}$, that represents the objective function to be learned for each layer l, and for which the SAE state-of-the-art architecture will

be used. Note also that $\boldsymbol{x}^*$ will be used to refer to the input patches after applying the mask, which may contain values in the range $[0, 255]$, for the original pixels of the image, but also the value -1 as a mask to mark the parts without annotated information. This mask is therefore applied to the input data in $\mathcal{X}$ and will be used by the masking layer (described below) added to the networks f^l to ignore those parts during the training process.

Algorithm 1 describes the oversampling method proposed to obtain the set $\mathcal{S}$ previously described. This method receives as input the set of images $\mathcal{I}$, the set with the annotated data $\mathcal{L}$, the layer l to be processed, the $\lambda\%$ of minimum patch information, the total *size* of sampling to perform, and the set $\mathcal{M}$ that contains the list of patches annotated with their coordinates in the input images. The algorithm first iterates through the number of patches annotated in $\mathcal{M}$ (`line 3`) and for each one obtains the index j of the image it corresponds to (`line 4`). It then iterates for the number of samples that have to be extracted for that annotated patch (`line 5`) and, for each one, performs the following steps: 1) randomly selects the sample coordinates p using the mask of that patch and taking into account the minimum $\lambda\%$ of annotated pixels allowed (`line 6`); 2) extracts the patch $\boldsymbol{x}$ from $\mathcal{I}_j$ using the coordinates p (`line 7`); 3) applies the mask to set a constant value (-1) in those pixels that are not part of the annotated area (`line 8`); 4) retrieves the layout annotations $\boldsymbol{y}$ for that sample (`line 9`); and 5) both $\boldsymbol{x}^*$ and $\boldsymbol{y}$ are added to the set $\mathcal{S}$. The algorithm repeats this process until reaching the requested *size*, finally returning the set $\mathcal{S}$ obtained.

Algorithm 1 Random masking patches generator.

1: **function** SAMPLEGENERATION($\mathcal{I}, \mathcal{L}, \mathcal{M}, l, \lambda, size$)
2: $\quad\mathcal{S} \leftarrow \emptyset$
3: $\quad$**for** $i \leftarrow 1$ to $|\mathcal{M}^l|$ **do**
4: $\quad\quad j \leftarrow$ `getPatchIndex`($\mathcal{M}^l_i$)
5: $\quad\quad$**for** $k \leftarrow 1$ to $\frac{size}{|\mathcal{M}^l|}$ **do**
6: $\quad\quad\quad p \leftarrow$ `getRandomPosition`($\mathcal{M}^l_i, \lambda$)
7: $\quad\quad\quad \boldsymbol{x} \leftarrow$ `getWindow`($\mathcal{I}_j, p$)
8: $\quad\quad\quad \boldsymbol{x}^* \leftarrow$ `applyMask`($\boldsymbol{x}, \mathcal{M}^l_i, p$)
9: $\quad\quad\quad \boldsymbol{y} \leftarrow$ `getWindow`($\mathcal{L}^l_j, p$)
10: $\quad\quad\quad \mathcal{S} \leftarrow \mathcal{S} \cup (\boldsymbol{x}^*, \boldsymbol{y})$
11: $\quad\quad$**end for**
12: $\quad$**end for**
13: $\quad$**return** $\mathcal{S}$
14: **end function**

Note that the `getWindow`($\cdot$) function may apply additional data augmentation to the sample in order to further increase its variability.

This oversampling process is complemented by the proposal of a masking layer that is added to the network architecture f^l to ignore the pixels that are not annotated. This layer, as indicated in Section 2, has been previously used in other proposals to skip time steps in sequence processes and to mask the background in classification tasks. In this proposal, we adapt it to ignore the parts of the input with this mask and also propagate the mask to the following layers so that the non-annotated parts are not taken into account during the training process. Intuitively, the masking layer acts as a regularizer and data augmentation process. Given that the annotated and non-annotated parts will vary in position and size across iterations, the network is forced to generalize the weights learned during training by having to use different connections of the network and non-annotated pixels will not be used.

4. EXPERIMENTAL SETUP

This section describes the corpora and metrics considered for evaluation and the implementation details of the neural architecture. [1]

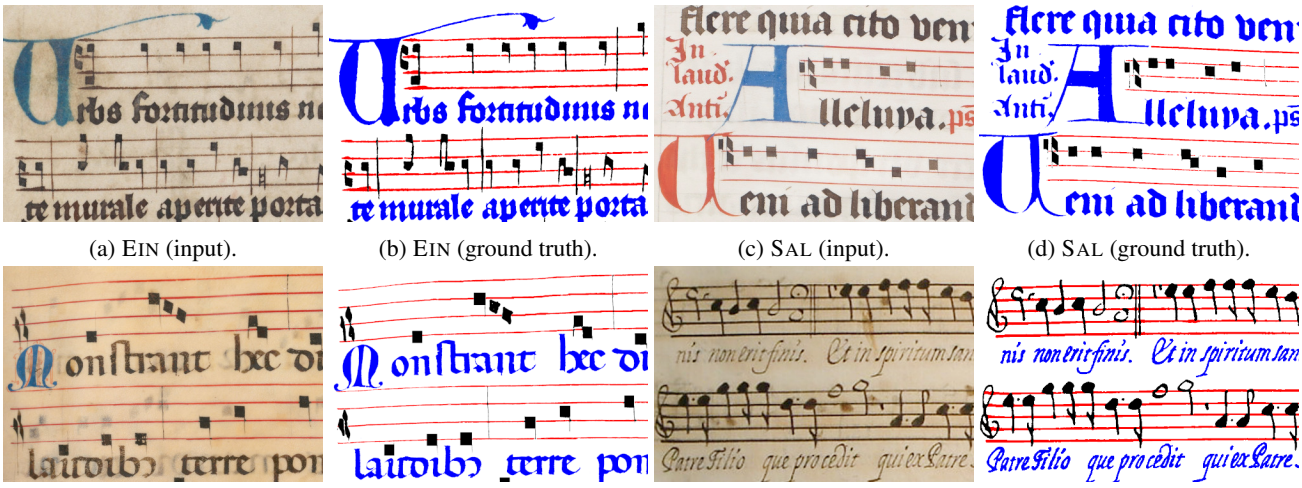
4.1 Corpora

For the experiments, we considered the following 4 datasets with manual pixel-wise annotations of 4 layers of information (`staff`, `notes`, `text`, and `background`). Figure 1 shows some examples for each manuscript and Table 1 includes a summary with their details.

- **EIN**: 9 high-resolution scanned pages of neumatic notation belonging to the Einsiedeln, Stiftsbibliothek, Codex 611(89), from 1314. [2]
- **SAL**: A set of 10 high-resolution images of pages from the Salzinnes Antiphonal manuscript (CDM-Hsmu M2149.14), in neumatic notation. It is available in the *Cantus Ultimus* platform. [3]
- **MS73**: Selection of 10 pages of square music notation from the miscellaneous choir book '*Dominican, CDN-Mlr MS Medieval 0073*' from Northern Italy, written between 13th and 15th centuries. This corpus is stored in the McGill Library collection, and it is online available through *Cantus Ultimus*. [4]
- **CAP**: A compilation of mensural notation manuscripts from the 17-18th centuries belonging to the 'Cathedral of Our Lady of the Pillar' in Zaragoza (Spain), introduced for OMR purposes by Calvo-Zaragoza et al. [33]. We use a subset of the corpus, with 10 manually pixel-wise annotated pages.

In all the cases, we used 4 images for training, 2 images for validation, and the remaining for testing. After preliminary experiments and also based on previous proposals, we selected a patch size of 256×256 pixels to extract from these images. To be fair and more realistic, we use the same number of samples for the validation set as for the training partition. This is because, in a real case, it would be necessary to annotate the validation partition as well, so it is not fair to use the entire pages to validate the models in a few-shot scenario. This does not apply to the test set, for which we use all available data.

[1] `https://github.com/fjcastellanos/FewShotLayoutAnalysisMusic.git`
[2] `http://www.e-codices.unifr.ch/en/sbe/0611/`
[3] `https://cantus.simssa.ca/manuscript/133/`
[4] `https://cantus.simssa.ca/manuscript/35/`



(a) EIN (input). (b) EIN (ground truth). (c) SAL (input). (d) SAL (ground truth).

(e) MS73 (input). (f) MS73 (ground truth). (g) CAP (input). (h) CAP (ground truth).

Figure 1: Examples of images extracted from the corpora described in Table 1. In the ground truth images: red pixels represent the staff lines annotation, black is used for music symbols, blue for text, and white for the background.

Corpus	# imgs	Resol.	Layers (%)			
			BG	St	No	Te
EIN	9	6 496 × 4 872	87.9	3.5	2.7	5.9
SAL	10	5 847 × 3 818	87.6	2.4	2.5	7.5
MS73	10	6 990 × 4 797	93.4	1.8	1.8	3.0
CAP	10	2 126 × 3 065	85.7	6.6	5.1	2.6

Table 1: Details of the corpora considered including the number of images (# imgs), the average resolution and the proportion of pixels for each layer of interest, with **BG** for background, **St** for staff lines, **No** for notes, and **Te** for text.

4.2 Metrics

To evaluate the performance of our few-shot approach, we resorted to the F-score (F_1) figure of merit to avoid possible biases toward any particular class given the inherent label imbalance in the datasets considered (see Table 1). Assuming a binary classification scenario, this metric is defined as

$$F_1 = \frac{2 \cdot TP}{2 \cdot TP + FP + FN}, \quad (1)$$

where TP, FP, and FN denote the True Positives, False Positives, and False Negatives, respectively.

Finally, given the non-binary nature of the task at hand, we considered the use of the macro-averaged F-score (F_1^m) as the average of the F_1 values computed for each layer. Mathematically, this metric is defined as

$$F_1^m = \frac{\sum_{l=1}^{|\mathcal{L}|} F_1^l}{|\mathcal{L}|}, \quad (2)$$

where F_1^l is the F_1 calculated for the layer l assuming a one-versus-all evaluation framework and $|\mathcal{L}|$ represents the total number of layers of information (in our case 4).

4.3 Implementation details

The architecture considered is based on a previous work [18], in which a framework consisting of a series of SAE models—one for each layer to be predicted—was proposed. SAE follows a U-net architecture, in which an image of size $w \times h$ (in our case a 256×256 pixels patch) is given as input, and the output is a matrix of the same size that contains the confidence value of pixels belonging to the layer of interest. In our case, we have four layers to be predicted, so we will have four SAE models, each one specialized in one particular layer.

For the experimentation, we resort to the same architecture proposed in the original work. An encoder with four blocks composed of a convolutional layer of 32 filters of 3×3, a sub-sampling of 2×2, a batch normalization, a Rectified Linear Unit (ReLU) activation, and a dropout of 0.4. On the decoder side, the blocks follow the same scheme except for the sub-sampling, which is replaced by an oversampling of the same rate. The last layer of the decoder is connected to a convolution with one 3×3 filter and a sigmoid activation to obtain the result of the prediction with values between 0 and 1. This architecture was only changed to add the masking layer after the input.

Note that each SAE was trained using the binary cross-entropy loss for up to 200 epochs with a batch size of 16, and an early stopping criterion of 20 epochs of no improvement on the validation set. Adam optimizer [34] was used with a learning rate of 0.001.

Furthermore, to favor the convergence of the model, the input images were normalized in the range $[0, 1]$. The mask was applied over this result, so the inputs can actually contain the values $\{-1\} \cup [0, 1]$. For the extraction of patches, a value of λ of 2.5% was used, since it allowed obtaining chunks with sufficient information. In addition, we also considered standard data augmentation to increase data variability by applying random rotations between -45° and 45°, zoom variations between 0.8x and 1.2x, and horizontal and vertical flips.

5. RESULTS

This section presents and discusses the results obtained with the proposed method.

First, a preliminary experiment was carried out to analyze the influence of the amount of oversampling. For this, starting from a single annotated patch, we studied the result obtained by increasing the number of randomly extracted samples around the annotated patch using the proposed technique. Fig. 2 shows the average results of this experiment in the validation set for all layers and considering both the application and non-application of data augmentation. For a small number of randomly extracted samples, the proposal achieves approximately 30% of F_1^m. The average result is improved as the number of samples extracted increases, reaching over 70% of F_1^m for 512 samples and barely improving for the case of 1 024 samples. Additional data augmentation does not help to improve the results, only for cases of sample size equal to or less than 128. This may be because the proposed oversampling method can be considered as a data augmentation process, so that, from a given amount of sampling, there is enough variability and other techniques of data augmentation may not be necessary.

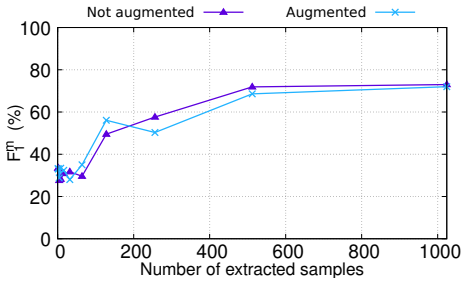


Figure 2: Preliminary experiment to study the influence of the number of samples drawn randomly from one annotated patch of 256 × 256 pixels. The result obtained in terms of F_1^m (%) in the validation partition is shown, considering both the application and the non-application of additional data augmentation.

Based on these results, the sampling size is set to 512 for the following experiments. Also, since standard data augmentation seems detrimental in combination with our proposal, we decided not to use it.

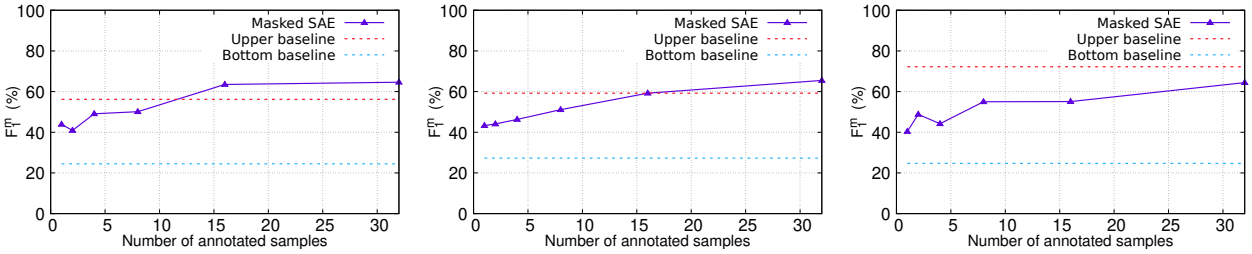
The selected configuration was evaluated using the test set, carrying out an analysis of the influence of the number of patches annotated (from 1 to 32) and the influence of these being extracted from the same page or from several (up to 4, which would generate more variability). Fig. 3 shows these results compared to two baselines: an upper one representing the state-of-the-art model [18] trained with all available information (if the entire training set was annotated) and a lower bound training this model with only one annotated patch (in both cases without applying the proposed masking layer). One initial observation is that the three case studies (with 1, 2, or 4 pages) demonstrate comparable trends. The results, as expected, show an increasing trend with the number of annotated samples, from an average F_1^m of 40% when training with one annotated sample to ∼62% when using 32 annotated patches, and stabilizing (or improving less) from 16 to 32 annotated patches.

If these results are compared with the baselines, it can be seen how the proposal exceeds the lower bound by 16% when training with one annotated sample and that it equals or even improves the upper baseline in the cases with 1 and 2 pages from 16 annotated samples. Also, it is only 7% worse than the state of the art for the 4-page case but with a much lower annotated data requirement (32 samples, which represents 39% of the total information).

Layer	Annotated samples						Baseline	
Corpus	**1 (1%)**	**2 (2%)**	**4 (5%)**	**8 (10%)**	**16 (20%)**	**32 (39%)**	**Bt (1%)**	**Up (100%)**
staff								
EIN	10.5	39.8	64.1	62.1	83.9	78.1	0.0	87.3
SAL	72.0	75.4	75.7	75.7	74.8	87.4	0.0	90.8
MS73	11.3	13.9	17.7	12.9	92.8	94.1	0.0	91.4
CAP	66.2	75.6	75.2	79.0	79.9	82.5	0.0	47.0
Avg.	40.0	51.2	58.3	57.4	82.9	85.5	0.0	79.1
note								
EIN	19.0	16.7	20.7	0.0	20.4	26.3	0.0	77.8
SAL	35.3	3.3	21.2	4.1	38.6	50.2	0.0	4.1
MS73	0.2	3.2	6.7	7.3	7.1	7.3	0.0	2.7
CAP	66.7	69.7	73.0	77.9	81.2	82.6	0.3	8.3
Avg.	30.3	23.2	30.4	22.3	36.8	41.6	0.1	23.2
text								
EIN	22.9	15.1	17.2	67.3	31.7	37.0	11.3	11.3
SAL	67.6	15.5	46.1	32.3	71.7	73.4	0.0	78.5
MS73	6.3	9.4	26.2	16.7	15.3	14.3	0.0	13.5
CAP	3.6	0.0	15.1	37.0	45.4	16.7	3.6	12.7
Avg.	25.1	10.0	26.2	38.3	41.0	35.4	3.7	29.0
background								
EIN	93.9	93.8	93.8	93.8	93.8	93.7	93.7	93.7
SAL	93.2	93.2	93.2	93.2	97.9	99.1	93.2	98.5
MS73	40.0	36.8	46.6	49.2	87.4	96.8	96.8	96.8
CAP	93.6	93.6	93.7	93.6	93.6	93.6	93.6	93.6
Avg.	80.2	79.4	81.8	82.5	93.2	95.8	94.2	95.7

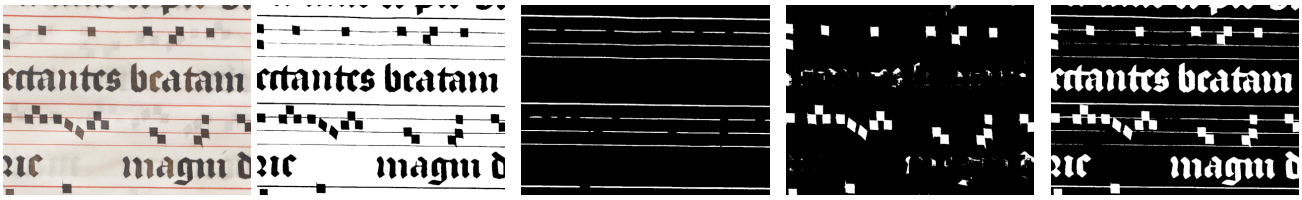
Table 2: Average results in terms of F_1 (%) for each layer considering 1 page in a few-shot evaluation. The percentage of annotated information is indicated between parentheses. **Bt** represents the bottom baseline, which is the state-of-the-art model trained with 1 annotated sample per page, and **Up** is the upper baseline, with full pages used for training. Both baselines do not apply any masking.

From these results, we now analyze in detail the case of a single page, since it represents the most extreme case as it has less variability available for the annotation. Table 2 shows a summary of the results obtained individually for each dataset and layer considered, including the baselines and the percentage of the image used in each case. As in the previous results, it is observed that the performance of our approach improves as more annotated samples are used. In this case, we can analyze how the results vary according to the layer and the corpus evaluated. In general, the proposal improves the bottom baseline, in some cases, such as `staff`, `notes`, and `text`, by a wide margin. However, note that this baseline fails to converge on most layers, except for the `background` one. In this case, on average, the proposal only improves the baseline by using 32 samples—39% of the image. This is due to the fact that for fewer annotated samples, poor overall results are obtained for the MS73 dataset. This may be because this dataset presents a greater variability of backgrounds. In



(a) 1 training page. (b) 2 training pages. (c) 4 training pages.

Figure 3: Average results in terms of F_1^m (%) with respect to the number of annotated samples (from 1 to 32) and the number of pages (from 1 to 4). Dashed lines represent baseline results for reference. The upper reference line indicates the results of the state-of-the-art model trained with fully annotated pages, while the lower reference line represents the results obtained when only one sample is annotated. Note that both baselines do not use the proposed masking method.



(a) SAL (input). (b) Background layer. (c) Staff layer. (d) Notes layer (e) Text layer.

Figure 4: Example of the results obtained in SAL for the four layers considered in this work. The method was trained with 32 samples drawn from one page. White represents the detected information for the particular layer.

fact, the rest of the layers of that corpus also obtain low-performance values when the annotated data is scarce.

Regarding the upper baseline, it can be seen how the proposal, on average, improves it in all layers, although it requires a different number of labeled samples depending on the layer. As stated before, on average, from 16 patches or 20% labeling, a better result is achieved. It is interesting that for the simplest and more homogeneous layers (such as `staff` and `background`), the upper baseline obtains a better result and it is more difficult for the proposal to overcome it, while for the more difficult ones that present greater variability (`notes` and `text`), the baseline obtains a worse result while the proposal achieves a greater margin of improvement. This may be due to the fact that the proposal performs some overfitting in the simplest cases with less variability and, therefore, requires a greater number of labeled samples to learn it.

To complement the quantitative results, Fig. 4 shows an example of prediction for SAL. As can be seen, the background and the staff layers are correctly retrieved, and some false positives can be found in the notes layer. The text layer seems the most challenging as it is not able to differentiate the ink of the text from other elements. However, the text is recovered, and the false positives could be removed by combining the predictions obtained for the other layers.

6. CONCLUSIONS

In this work, we presented a few-shot neural approach for pixel-wise layout analysis of music score images. The proposal includes a masking layer, which acts as a regularizer, that is combined with an oversampling technique to leverage the limited annotated information available. The oversampling technique extracts annotated parts of the images at different random positions at each training iteration, leaving annotated and non-annotated information in different positions of the input. This strategy forces the neural architecture to generalize the learned weights, similar to a data augmentation process but adapted to the case of few-shot and partial annotation in documents.

The proposal is evaluated on four benchmark datasets to study the influence of the amount of annotated data in the layout analysis task. We found that the number of annotated samples is key to optimizing performance, and annotating a relatively small number of them—between 16 and 32 samples, which represents using only between 20% and 39% of the total information—can achieve average results of 65.5% of F_1^m, which is very close to the result obtained by the state of the art (72%) using the entire training set annotated. It is also interesting to note that the proposal obtains similar results when labeling more pages, so it is enough to have a single page for training and perform a partial annotation of between 16 and 32 patches.

In general, the approach shows very competitive results in few-shot scenarios. Therefore, we hope this research can open doors to new avenues in this line. Reducing the amount of annotated data required for pixel-wise layout analysis is essential, and techniques such as domain adaptation and transfer learning may help to reduce human effort. We plan to investigate new ways to address this problem, including to combine domain adaptation techniques with our masking proposal and studying the feasibility of incremental and active learning.

7. ACKNOWLEDGMENT

This work was supported by the I+D+i project TED2021-132103A-I00 (DOREMI), funded by MCIN/AEI/10.13039/501100011033. This work also draws on research supported by the Social Sciences and Humanities Research Council (895-2013-1012) and the Fonds de recherche du Québec-Société et Culture (2022-SE3-303927).

8. REFERENCES

[1] D. Bainbridge and T. Bell, "The challenge of optical music recognition," *Computers and the Humanities*, vol. 35, no. 2, pp. 95–121, 2001.

[2] A. Rebelo, I. Fujinaga, F. Paszkiewicz, A. R. S. Marçal, 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, 2012.

[3] J. Calvo-Zaragoza, J. H. Jr., and A. Pacha, "Understanding optical music recognition," *ACM Comput. Surv.*, vol. 53, no. 4, Jul. 2020.

[4] J. Calvo-Zaragoza, F. J. Castellanos, G. Vigliensoni, and I. Fujinaga, "Deep neural networks for document processing of music score images," *Applied Sciences*, vol. 8, no. 5, p. 654, 2018.

[5] I. Fujinaga and G. Vigliensoni, "The art of teaching computers: The SIMSSA optical music recognition workflow system," in *27th European Signal Processing Conference, EUSIPCO, A Coruña, Spain, September 2-6*. IEEE, 2019, pp. 1–5.

[6] F. J. Castellanos, A. J. Gallego, and J. Calvo-Zaragoza, "Unsupervised domain adaptation for document analysis of music score images," in *Proceedings of the 22nd International Society for Music Information Retrieval Conference, ISMIR 2021, Online, November 7-12, 2021*, 2021, pp. 81–87.

[7] J. Sauvola and M. Pietikäinen, "Adaptive document image binarization," *Pattern Recognition*, vol. 33, no. 2, pp. 225–236, 2000.

[8] N. R. Howe, "Document binarization with automatic parameter tuning," *International Journal on Document Analysis and Recognition*, vol. 16, no. 3, pp. 247–258, 2013.

[9] T. Pinto, A. Rebelo, G. A. Giraldi, and J. S. Cardoso, "Music score binarization based on domain knowledge," in *5th Iberian Conference on Pattern Recognition and Image Analysis, Las Palmas de Gran Canaria, Spain*, 2011, pp. 700–708.

[10] Q. N. Vo, S. H. Kim, H. J. Yang, and G. Lee, "An MRF model for binarization of music scores with complex background," *Pattern Recognition Letters*, vol. 69, no. Supplement C, pp. 88–95, 2016.

[11] J. A. Burgoyne and I. Fujinaga, "Lyric extraction and recognition on digital images of early music sources," in *Proceedings of the 10th International Society for Music Information Retrieval Conference*, 2009, pp. 723–728.

[12] V. B. Campos, J. Calvo-Zaragoza, A. H. Toselli, and E. Vidal, "Sheet music statistical layout analysis," in *15th International Conference on Frontiers in Handwriting Recognition, Shenzhen, China*, 2016, pp. 313–318.

[13] C. Dalitz, M. Droettboom, B. Pranzas, and I. Fujinaga, "A comparative study of staff removal algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 5, pp. 753–766, 2008.

[14] J. Dos Santos Cardoso, A. Capela, A. Rebelo, C. Guedes, and J. Pinto da Costa, "Staff detection with stable paths," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 6, pp. 1134–1139, 2009.

[15] T. Géraud, "A morphological method for music score staff removal," in *International Conference on Image Processing*, 2014, pp. 2599–2603.

[16] A. Gallego and J. Calvo-Zaragoza, "Staff-line removal with selectional auto-encoders," *Expert Systems with Applications*, vol. 89, pp. 138–48, 2017.

[17] J. Calvo-Zaragoza, G. Vigliensoni, and I. Fujinaga, "One-step detection of background, staff lines, and symbols in medieval music manuscripts with convolutional neural networks," in *Proceedings of the 18th International Society for Music Information Retrieval Conference, Suzhou, China*, 2017, pp. 724–730.

[18] F. J. Castellanos, J. Calvo-Zaragoza, G. Vigliensoni, and I. Fujinaga, "Document analysis of music score images with selectional auto-encoders," 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. 256–263. [Online]. Available: http://ismir2018.ircam.fr/doc/pdfs/93_Paper.pdf

[19] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning (still) requires rethinking generalization," *Communications of the ACM*, vol. 64, no. 3, pp. 107–115, 2021.

[20] J.-M. Lee and D.-s. Kang, "Improved method for learning data imbalance in gender classification model using da-fsl," *Multimedia Tools and Applications*, pp. 1–19, 2021.

[21] X. Li, L. Yu, C.-W. Fu, M. Fang, and P.-A. Heng, "Revisiting metric learning for few-shot image classification," *Neurocomputing*, vol. 406, pp. 49–58, 2020.

[22] M. Cogswell, F. Ahmed, R. B. Girshick, L. Zitnick, and D. Batra, "Reducing overfitting in deep networks by decorrelating representations," in *4th International Conference on Learning Representations, ICLR*, 2016.

[23] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, no. 1, p. 60, 2019.

[24] Y. Wang, Q. Yao, J. Kwok, and L. M. Ni, "Generalizing from a few examples: A survey on few-shot learning," 2019.

[25] C. Simon, P. Koniusz, R. Nock, and M. Harandi, "Adaptive subspaces for few-shot learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4136–4145.

[26] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *International Conference on Machine Learning (ICML) - Deep Learning workshop*, vol. 2, 2015, pp. 1126–1135.

[27] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra *et al.*, "Matching networks for one shot learning," *Advances in neural information processing systems*, vol. 29, pp. 3630–3638, 2016.

[28] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Advances in Neural Information Processing Systems*, 2017, pp. 4077–4087.

[29] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1199–1208.

[30] S. Jadon, "An overview of deep learning architectures in few-shot learning domain," *arXiv preprint arXiv:2008.06365*, 2020.

[31] F. Medhat, D. Chesmore, and J. Robinson, "Masked conditional neural networks for environmental sound classification," in *Artificial Intelligence XXXIV: 37th SGAI International Conference on Artificial Intelligence, AI 2017, Cambridge, UK, December 12-14, 2017, Proceedings*. Springer, 2017, pp. 21–33.

[32] M. Suresh, A. Sinha, and R. Aneesh, "Real-time hand gesture recognition using deep learning," *International Journal of Innovations and Implementations in Engineering*, vol. 1, 2019.

[33] 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 InformationRetrieval Conference, ISMIR 2016, New York City, United States, August 7-11, 2016*, 2016, pp. 509–514.

[34] 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. [Online]. Available: http://arxiv.org/abs/1412.6980

TAPTAMDRUM: A DATASET FOR DUALIZED DRUM PATTERNS

Behzad Haki **Błażej Kotowski** **Cheuk Lun Isaac Lee** **Sergi Jordà**

Music Technology Group, Universitat Pompeu Fabra, Barcelona, Spain

behzad.haki@upf.edu, blazej.kotowski@upf.edu, clilee@connect.ust.hk, sergi.jorda@upf.edu

ABSTRACT

Drummers spend extensive time practicing rudiments to develop technique, speed, coordination, and phrasing. These rudiments are often practiced on "silent" practice pads using only the hands. Additionally, many percussive instruments across cultures are played exclusively with the hands. Building on these concepts and inspired by Einstein's probably apocryphal quote, "Make everything as simple as possible, but not simpler," we hypothesize that a dual-voice reduction could serve as a natural and meaningful compressed representation of multi-voiced drum patterns. This representation would retain more information than its corresponding monotonic representation while maintaining relative simplicity for tasks such as rhythm analysis and generation. To validate this potential representation, we investigate whether experienced drummers can consistently represent and reproduce the rhythmic essence of a given drum pattern using only their two hands. We present TapTamDrum: a novel dataset of repeated dualizations from four experienced drummers, along with preliminary analysis and tools for further exploration of the data.

1. INTRODUCTION

1.1 Motivation

Music is a fundamental aspect of human culture, and rhythm is a central element of musical expression. Many different cultures have developed complex and sophisticated rhythmic traditions that are deeply rooted in their history, language, and social structures [1]. Representing music, and more specifically rhythm, in a symbolic form is crucial for a wide range of purposes, including communication, and preservation of musical traditions. Moreover, symbolic representations of music are essential for enabling the efficient processing and manipulation of musical data by computers, enabling new possibilities for music analysis and creation. However, representing complex rhythmic patterns in a notation system that accurately captures their essence and feeling can be a challenging task, particularly when the rhythms are highly complex or involve multiple layers of interlocking patterns.

Some attempts on representing percussion in a domain specific-way have been made, initially focusing on a single stream of onsets [2]. In [3], Toussaint offers a holistic analysis of rhythmic patterns, with an approach largely centred around monotonic representations. Representing a rhythmic pattern in a monotonic stream of offsets has indeed proven to be successful for some tasks like transforming a sequence of taps to a fully-orchestrated percussive pattern in GrooVAE [4]. However, while a rhythmic pattern reduced to a monotonic stream of its onsets retains part of its horizontal structure related to temporality, it also loses its vertical quality, which is related to the interplay between different voices [3]. As a result, a monotonic pattern transformation fails to capture the complete essence of a multi-voice rhythm, as noticed by Lartillot and Bruford [5]. Inspired by Einstein's probably apocryphal quote, "Make everything as simple as possible, but not simpler," in this work we hypothesise that a dual-voice reduction could serve as a more natural and meaningful compressed representation of multi-voiced drum patterns than its monotonic equivalent, and that this representation could preserve some quality related to the interaction or tension between instruments, while maintaining relative simplicity for tasks such as rhythm analysis and generation.

The rationale behind this approach can also be related to the role of the human motor system in rhythm perception. The importance of the motor system in rhythm creation and perception is rooted in the fact that many musical rhythms are based on movements that involve two limbs. It was previously documented that the synchronisation of movement to a musical pulse happens automatically, whether of the hand movements [6, 7], walking [8], or dancing [9]. A number of studies prove that motor areas of the brain play a significant role in beat perception and synchronisation. As Patel and Iversen argue in [10], the ability to coordinate two limbs in a synchronised and precise manner is essential for playing musical instruments and dancing to music. This is exemplified by the tradition of drumming, which typically involves striking a drum with two hands. The renowned drummer Jaki Liebezeit, even simplified his drum kit to only require two hands to play, and developed a rhythmic notation system called E-T that employs only two symbols to represent any rhythm, accommodating any musical situation [11]. Thus, the use of two limbs in music and rhythm is not only a fundamental aspect of motor

© 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, "TapTamDrum: A Dataset for Dualized Drum Patterns", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

coordination, but also a practical consideration in the representation and communication of musical rhythms.

A bell pattern is a two-voiced pattern frequently used in West African music, which is usually used as a key pattern to suggest a temporal organisation for different instruments and musicians [12]. This pattern is used from the Sub-Sahara, to West Africa, to the central lands of Congo and the Nyusa lands in Southeast Africa [13]. Agawu [14] claims that the notoriously complex rhythms in West African music can be represented with a bell pattern, which could usually be played with two-voiced percussion instruments. This African tradition migrated to the new world during the colonial period. Rooted in the Caribbeans, the Clave cross-rhythm also splits the beat into two voices, one regular beat and another irregular [15].

Additionally, subjective rhythmization is a widely recognized phenomenon in auditory perception [16], which occurs when individuals are exposed to monotonous auditory stimuli such as the ticking of a clock. Rather than hearing a simple "tick-tick-tick" pattern, our brains transform the sound into a more complex rhythmic sequence, such as "tick-tock-tick-tock," comprising two distinct parts. This naturally occurring cognitive process enhances the subjective perception of rhythm and makes the stimulus more engaging and dynamic to the listener. The concept of rhythm streams, which is linked to the theory of auditory streaming, is examined by Witek et al. [17]. Their research reveals that the addition of a single instrumental component to a monotonic rhythmic pattern can greatly influence its perceived rhythm, whereas the addition of another component to a two-instrument pattern has a discernible impact only in certain instrumentation contexts. Finally, Lartillot and Bruford [5] argue that any rhythm can be reduced to an oscillation between two states: high and low. Their rule-based system transforms multi-voice patterns into a monotonic stream of timed events representing the toggles between the states and their accentuations.

All the aforementioned evidences provide support for the initial hypothesis that simplified, dual-voice representations of multi-voice rhythmic patterns may be adequate in communicating both the vertical and horizontal characteristics of rhythm.

1.2 Rhythm Pattern Dualization

To formalize this idea, we introduce the novel concept of dualization of rhythm patterns. The task of rhythm dualization could be defined as the transformation of any multi-voice rhythmic pattern to another pattern composed of a maximum of two voices, while preserving the coherence and the perceptual essence of the original rhythm as much as possible. Dualization involves simplifying and highlighting the most essential features of complex rhythmic patterns. This dualized representation can be viewed as a form of abstraction, in which the most essential features of a rhythm are distilled and represented in a way that is easier to process, grasp, and perform.

Dualization could also enhance the creative and expressive potential of contemporary musicians, by providing them with a tool for exploring and adapting traditional rhythmic patterns in new and innovative ways. Moreover, the study of dualization can shed light on the cognitive and neural mechanisms involved in rhythm perception and performance, as well as the cultural and historical factors that shape rhythmic traditions. By introducing this novel concept, we hope to stimulate further research and innovation in the field of rhythm notation, and to advance our understanding of the cognitive, cultural, and creative processes involved in rhythmic expression.

In the next section, we present the dualization experiments we have conducted with the participation of four highly skilled professional drummers to validate our hypothesis. We introduce the dataset that emerged from these experiments, which serves as a valuable resource for our analysis. In Section 3, we delve into a detailed analysis of this dataset, shedding light on key findings and insights. Subsequently, in Section 4, we highlight potential applications and promising avenues for future research.

2. METHODOLOGY

While monotonic representations of multi-voiced rhythms can be obtained by simply flattening any rhythmic pattern, the process of dualizing a multi-voiced rhythm appears to be less straightforward. To further support our hypothesis that dualized patterns can serve as a natural and more meaningful compressed representation of multi-voiced drum patterns compared to their monotonic counterparts, we aimed to investigate whether there is a level of consensus or consistency in how multi-voiced patterns can be dualized. To this end, we recruited 4 professional drummers and conducted various dualization exercises, exploring different approaches and gathering insights into the process of dualization.

2.1 Preparation of Rhythmic Material

To ensure a diverse range of rhythms for dualization, we utilized Magenta's Groove MIDI Dataset [4]. This dataset comprises of 13.6 hours of live recordings performed on a Roland TD-11 electronic drum kit, each labeled with beat type (i.e. "beat" or "fill"), time signature, and style. The recordings are not quantized (neither in time nor velocity) and vary in duration and styles. For our work, we focused on the "beat" subset of the dataset resulting in 503 original recordings. In this subset, only twelve performances were not in 4/4 meter, and subsequently, we dismissed them. Table 1 summarizes the distributions of styles within the final selected samples.

	Rock	Funk	Latin	Jazz	Hip-hop	Other
Count	124	42	41	34	25	79
Percentage	36%	12%	12%	10%	7%	23%

Table 1. Style distributions within the 345 2-bar samples selected from GMD

The original GMD recordings vary from several seconds to a few minutes. As a result, for each session we

selected a single 2-bar segment with the highest total cosine similarity with every other 2-bar segments within the session (similarity was calculated between fully quantized patterns). This approach ensured that we had a diverse and representative set of 2-bar rhythms from different sessions and styles for our dualization experiments. These representative segments were further processed to exclude patterns that contained only 2 or less voices (which would have made the dualization task trivial). This left us with a final set of 345 individual meaningful and non-trivial 2-bar loops, suitable for our dualization experiments.

2.2 Data Collection Sessions

We hired 4 drummers with expertise in Western drumming tradition, two of them with conservatory experience, as shown in Table 2. Each drummer participated in several individual sessions of approximately one hour each. The experiments consisted of playing 2-bar multi-voice drum loops to the participants and asking them to concurrently perform dualized versions using only drum sticks on a Roland HandSonic HPD-15 MIDI drum pad. Participants were instructed to only hit the left region of the drum pad with the left hand and vice-versa.

	P1	P2	P3	P4
Age	41	20	24	22
Experience (Years)	25	11	10	9
Dominant Hand	Left	Left	Right	Right

Table 2. Demographics of the participants

For each of the data collection sessions, an Ableton Live [1] project was prepared in advance, containing the drum patterns to be dualized at their original tempo, without any velocity or timing quantization. As shown in Figure 1, the sessions were set up such that the participant would listen to a looping 2-bar pattern and after 1 or 2 repetitions would start to concurrently play their dualized interpretations of the pattern. The participants were allowed to continue their dualizations for as many repetitions as desired until they were confident in the accuracy of the dualization. All the drum patterns in all of the sessions were synthesized using a single sound source. [2] No auditory feedback of dualizations were provided to the participants except for the acoustic sound of the drum pad.

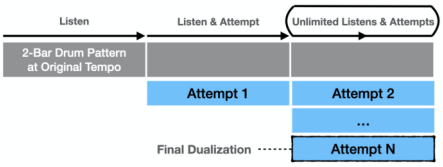


Figure 1. Set up of the dataset collection sessions

2.3 Post Session Questionnaire and Interviews

After the first recording session, each participant was given a questionnaire and an open interview was also conducted.

[1] www.ableton.com
[2] *Neutral* preset of Addictive Drums 2's Fairfax sound pack

The questionnaire comprised of three sections: (1) general information about the participant, (2) assessment of the intuitiveness of the task and confidence in their performance, and (3) exploration of how various rhythmic factors and metrics (e.g., number of instruments, tempo, genre, density, syncopation, mostly extracted from [18, 19]) could potentially influence the dualization process. Parts 2 and 3 of the questionnaire utilized a 7-point Likert scale ranging from 0 to 6. Results from these two parts are summarized in Table 3.

	P1	P2	P3	P4	Avg.
General Impressions					
Intuitiveness	4	5	6	5	5
Confidence	6	3	4	4	4.25
Influence of Rhythmic Features*					
No. of Instruments	0	6	5	5	4
Beat Division	0	4	3	5	3
Tempo	0	4	3	2	2.25
Style	6	3	5	6	5
Familiarity with Style	6	6	6	5	5.75
Syncopation-ness	6	5	5	1	4.25
Dynamics	6	4	3	4	4.25
Note Density	6	4	5	4	4.75
LMH Distribution	6	3	2	5	4
LMH Syncopation-ness	6	2	4	1	3.25
LMH Dynamics	6	4	3	4	4.25
LMH Density	3	4	4	6	4.25

Table 3. Summary of the questionnaires.
* refers to the perceived importance that different rhythmic features have on the dualizations. LMH refers to the Low, Mid and High frequency regions. Descriptors taken from [18, 19].

Additionally, after the questionnaire, open interviews were conducted with each participant. In the following sections, we provide a summary with some of the more relevant an recurrent topics discussed.

2.3.1 Meaningfulness, Replicability and Universality

All four participants unanimously agreed that the concept of dualization is valid and meaningful. Despite one of them admitting to not having thought much about it before, they all found the concept intuitive and useful after the session. Despite that the patterns in the introductory sessions were randomly repeated without notifying the participants, they were able to notice the duplication of the same 2-bar tracks in the session, and they strongly believed that even though their corresponding dualized patterns might not be identical, they would likely share a high similarity. While the results were not entirely conclusive (as discussed in the next section), all participants also expressed confidence in having similar mindsets as their peers while dualizing the same rhythm. Participant 1 justified this idea by referencing the development of the Western percussive tradition from the snare drum, stating that fundamentally, drummers listen to key elements represented by the snare.

2.3.2 The Effects of Style and Repetition

One participant stated that genre greatly influences which voices to focus on; for example, in the case of rock, the snare and the bass drum would more often be followed,

whereas in jazz-influenced styles, the hi-hat and the ride cymbal could probably be included in the reduction, as they emphasise lay back and swing. However, not all participants agreed on interpreting dualization solely as the replication of the two most prominent voices. Some believed that they needed to first digest and understand each rhythm, before extracting its essential form for dualization. Two participants also mentioned that listening to the same pattern repeatedly in the experiment promoted a "dualization refinement", listening more deeply to each iteration and thus refining the extraction of the pattern's rhythmic essence.

2.3.3 Problems with the Recording Sessions

Participants reported some issues during the recording sessions, such as using always the same VSTI for all patterns, independently of their musical style. Also, the use of Roland HPD-15, which has a single surface with multiple pads, seems to encourage some drummer habits such as paradiddles, and some participants would have preferred a separated two-pad device. Some tracks were inadequately selected for a 4/4 based recording as Participant 1 interpreted them as a 6/8 time signature, even though GrooveMIDI labelled them as in 4/4. For some swing and shuffled rhythms sometimes the last semiquaver was cut. Participants were also aware of the style bias and of the abundance of Western influenced styles specifically rooted from a snare drum tradition.

3. DATASET AND ANALYSIS

In this section, we will start with providing an overview of the dataset. Subsequently, we will provide a preliminary analysis of the collected dualizations. The objective of this analysis is to establish whether there is any validity to the hypothesis that professional drummers are able to dualize drum patterns with some level of consistency, so as to establish whether further research on this topic is warranted.

During the dataset collection sessions, we had a limited time span during which all four participants were available. For the initial session, we had access to all four participants. This session was set up in a manner that would allow us to investigate two main questions: (1) whether a single drummer dualizes a given pattern consistently at different times (Intra Participant consistency) and (2) how consistently different drummers dualize the same drum pattern (Inter Participant consistency). As such, we selected 24 drum patterns to be presented randomly three times to each of the four drummers without notifying them about the repetitions (Subset A1). During the first stage of the data collection sessions, Participant 1 was able to participate longer, as a result, 48 more drum patterns were dualized by him in the same manner (Subset A2).

During the A2 sessions, Participant 1 notified us that he was aware that we were testing a single drum pattern multiple times. Following this comment, he told us that, if needed, he can dualize the drum patterns in two different ways, in his own terms, in a "simple" and a "complex" manner. Inspired by this comment, we confirmed with the other participants about this view of dualization. Subsequently, we modified the remaining sessions so as to explore whether there is any consistencies among the "simple" and "complex" dualizations. We were able to partially conduct these sessions with Participants 1 and 2 (Subset B1), while the remaining sessions were only conducted with Participant 1. Table 4 summarizes the collected data.

The final dataset consists of 1116 dualizations obtained from the participants. These dualizations were obtained from the set of 345 unique drum patterns. Moreover, each of the drum patterns were used in a single dualization test, that is, for example, a drum pattern used in Subset A1 was not reused in the A2, B1 and B2 subsets. The decision to not reuse the drum patterns in different subsets was made to maximize the number of drum patterns for which at least one set of dualizations were available in the final dataset.

In Section 3.1, we will use the collected dataset to provide a preliminary analysis on whether there are any intra/inter participant consistencies between the dualizations obtained at random from all four participants (Subset A1) - see Figure 2. Lastly, in Section 3.2, we will compare simple dualizations with their complex counterparts from both intra- and inter-participant perspectives. The analysis presented in this section has been done on a binary representation of the dualizations, fully quantized to a 16th note grid. Moreover, it should be noted that, as confirmed with the participants, a given dualization can be reproduced using an inverse hand combination. As such, to compare the rhythmic similarity between the dualizations, a hand-agnostic measure should be employed (e.g. $L0R0LL$ and *R0L0RR* patterns should be treated as identical - *L* and *R* refer to *Left* and *Right* hand hits at a 16th note time step and 0 refers to silence).

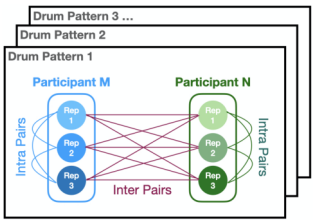


Figure 2. Inter-/Intra- pairs. For each drum pattern, intra-pairs are selected from each of the participants. Inter-pairs are selected by pairing a participant's repetition with all other repetitions corresponding to the same test.

To this end, we focused our analysis on the flattened versions of the dualizations, that is, the left and right hand patterns were superimposed onto a single sequence (e.g. *101011*). While this approach does not explore the function of each of the dualized streams, we believe that it is valid as a preliminary investigation, because if the flattened patterns show no rhythmic consistency, further analysis on the function of each hand may not be warranted. In other words, the validity of our hypothesis is contingent upon the presence of rhythmic consistency in the dualized patterns, and further analysis may be needed to determine the functional aspects of each hand in the dualizations.

Subset		Tested Drum Patterns	Repetitions Per Test				Total Dualizations
			P1	P2	P3	P4	
Three Repetitions (A)	Multi-Participant (A1)	24	3	3	3	3	288 (24×3 × 4)
	Single Participant (A2)	48	3	-	-	-	144 (48×3 × 1)
Simple vs Complex (B)	Multi-Participant (B1)	69	2	2	-	-	276 (69×2 × 2)
	Single Participant (B2)	204	2	-	-	-	408 (204×2 × 1)
Total		345	762	210	72	72	1116

Table 4. Summary of the collected dataset

3.1 Three Repetitions (A1 Session, Participants 1-4)

In Subset A1, we presented 24 drum patterns[3] to each of the four participants three times in a random order, resulting in a total of 288 obtained dualizations. We first examine the consistency of each participant's dualizations over the three repetitions (intra-participant analysis), and then we investigate the consistency of each of dualizations with their counterparts[4] from other participants (inter-participant analysis). To establish the similarity between two given patterns, we used the Jaccard similarity measure, defined as the ratio of the overlap of two sequences divided by the union. Moreover, to establish the perceptual similarity of the dualizations, we used Edit Distance [20, 21], defined as the minimum number of operations (insertions, deletions, or substitutions) required to transform one sequence into another. Figures 3 and 4 summarize the results of inter/intra-participant analysis using a pair-wise comparison. In both cases, in order to establish a baseline comparison, we also calculate the Edit and Jaccard values for the same number of pairs randomly selected (each random pair comes from dualizations obtained from two randomly selected participants and are ensured not to be associated with the same drum pattern).

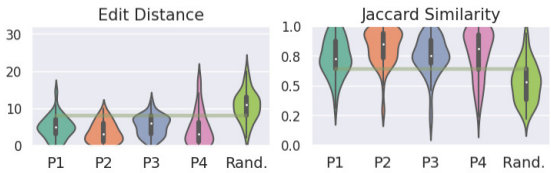


Figure 3. Intra-Participant Analysis of Subset A1

The results of the intra-participant distributions (Figure 3) show that the Edit distances are smaller for any of the four participants' repetitions compared to the distance between randomly paired dualizations. Similarly, the Jaccard similarities are also higher than the random pairs.

Unlike the intra-participant distributions, the edit distances for inter-participant dualizations have some overlap with the random pairs. This overlap is also observed in the Jaccard similarity values. However, despite this overlap, the inter-participant distributions still show a trend towards higher similarity values compared to the random pairs. The lower consistency between inter-participant dualizations, compared to the intra-participant dualizations may be an indicator that experienced drummers have a consistent dualized interpretation of rhythms, however, these interpretations vary to some extent compared to other drummers. While this is a possibility, we believe more comprehensive analysis is required prior to making this conclusion as there are a number of limitations to the approach taken in this paper.

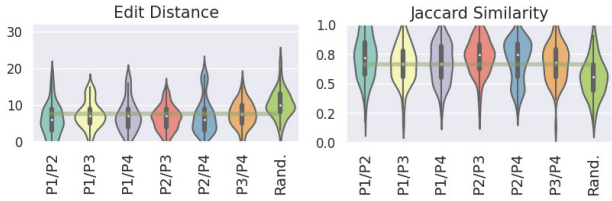


Figure 4. Inter-Participant Analysis of Subset A1

The current state of our analysis imposes several restrictions and simplifications that limit the depth of our study of the dataset. Firstly, it is constrained to the flattened versions of the dualizations. Secondly, it dismisses (quantizes) the velocity and micro-timing information, which are important factors in drumming that can affect nuances and groove. Experienced drummers often utilize velocity and micro-timing to add expressiveness to their playing, potentially leading to differences in the perceived rhythm. Therefore, a comprehensive investigation of the dualizations should incorporate these dimensions for a more nuanced understanding.

3.2 Simple/Complex (B Sessions, Participant 1)

As mentioned previously, Participant 1 pointed in the open interview that a dualization can be done in different manners: simple and complex. For further exploring this idea, in a second phase of the dataset collection sessions, two participants were asked to first dualize a drum pattern in a simple manner, and also immediately after, re-dualize the same pattern in a more complex manner. These sessions were partially conducted with Participants 1 and 2 (session B1), and more tests were done using Participant 1 (session B2). For the sake of brevity, we focus the analysis of this subset on Participant 1 (a total of 273 paired simple and complex dualizations).

Figure 5 shows that a major distinction between the simple and complex dualizations of Participant 1 is that the simple versions are considerably less active than their complex counterparts. One interesting observation is that the activity level of the dualizations obtained from Participant 1 during the first session (A1) (in which this simple/complex distinction had not been introduced) are more on par with the complex dualizations. This observation

[3] latin: 4, hiphop: 3 , jazz: 3, rock: 3, funk: 2, soul: 2, afrobeat: 1, afrocuban: 1, dance: 1, neworleans: 1, pop: 1, punk: 1, reggae: 1

[4] dualizations obtained from the same drum pattern tested

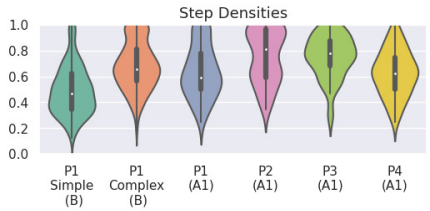


Figure 5. Step Densities. Left two distributions correspond to the 273 simple and complex dualizations obtained from P1 in sessions B1 and B2; remaining four correspond to 72 dualizations per each participant in A1

along with the step density distributions for Participants 2-4 raises a question that perhaps, unless restricted, the drummers default to more active dualizations. This tendency may be explainable from two perspectives: (1) in most styles, the hands are generally highly active as they are responsible for playing the majority of the drum kit, and (2) the dualizations may be by default more biased towards the rudiments that many drummers strenuously practice using their hands.

To further analyze the simple and complex dualizations, we calculated the edit distance and Jaccard distance between each pair of dualizations (see Figure 6). Similar to section 3.1, to establish a baseline comparison, we used an equal number of uncorrelated simple and complex pairs from Participant 1 dualizations. The results here show that, while lower in general, the distribution of the edit distances between the simple and complex pairs are partially overlapping. Moreover, this distribution is higher than the intra repetition distribution in Subset A1. These trends are also evident in the Jaccard similarity distributions. These differences relative to the intra distances in subset A1 are fully expected knowing that the dualizations are intentionally varied in this experiment.

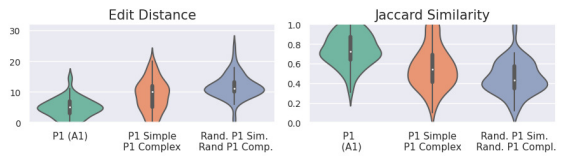


Figure 6. Distributions of distances/similarities between all of the Participants simple dualizations and their corresponding complex dualizations

4. APPLICATIONS AND FUTURE WORK

The majority of our work has been focused on data collection and curation, as well as preparing tools to allow the community to easily explore and study the data. All the collected data have been carefully processed and organized for both A and B subsets. Moreover, we have prepared an accompanying website that enables researchers to listen to all paired repetitions while also visualizing the piano rolls. Lastly, we have also developed an open-source API that allows for easy access to the data, visualization, synthesis, and analysis using both pre-implemented and third-party tools of interest. All these resources are publicly available at `https://taptamdrum.github.io`.[5]

We envision that the dataset can be used in a variety of studies. Rhythm reduction studies could use it to examine the simplification of multi-voiced patterns into dual-voiced ones. The dataset can also be used to develop computational models for drum pattern reduction. These models would be highly valuable as they allow for easier study of polyphonic drum patterns. Moreover, knowing that the dualizations of a single participant for a given drum pattern are highly correlated (i.e. they are perceptual interpretations of the same pattern), researchers can validate whether the features extracted from the dualizations are also highly correlated. This would provide a way to evaluate the effectiveness of rhythm feature extraction algorithms and potentially improve them. Similarly, to establish the perceptual relevance of rhythmic distance/similarity measures, researchers can use the simple/complex subset of the dataset to ensure that the proposed measures result in reasonable distances that correlate with the perceptual re-interpretations of a given pattern.

Lastly, the dataset can also be used for drum generation tasks. For instance, a generative model could be developed so as convert a dual sequence into a full drum pattern (similar to single voice rhythm into multi-voice drum generators [4, 22]). Such generative models can be used in many creative ways as during the inference stage, each of the left/right streams fed into the drum generator can be extracted from separate instruments/sources. Moreover, the random repetitions and the simple/complex repetitions can be used in developing deep metric learning models that rely on paired training samples.

5. CONCLUSIONS

In this work, we presented TapTamDrum, a novel dataset consisting of 1116 dualizations of drum patterns performed by four experienced drummers, covering 345 unique drum patterns selected from Magenta's GrooveMIDI dataset. The analysis conducted in section 3.1 provides valuable insights into the dataset. Firstly, it shows that there are intra-participant consistencies in the dualizations. That said, the inter-participant analysis are less definitive and require further detailed investigation. Moreover, the simple/complex comparisons (section 3.2) show that the complex dualizations are significantly more active than the simple ones while adhering to some level of rhythmic consistency with their simple counterparts. The analysis conducted in this work was preliminary and limited and has not explored the full potential of the dataset. The main focus of this work was to collect, curate, organize the dataset and also provide resources for prompt exploration of the data. To this end, we have prepared an accompanying website and an open-source API. Finally, this dataset can be used in a variety of rhythm related studies ranging from perception to generation.

[5] `https://github.com/taptamdrum/dataset`

6. ACKNOWLEDGMENTS

This research was partly funded by the Ministry of Science and Innovation of the Spanish Government. Agencia Estatal de Investigación (AEI). (Reference: PID2019-111403GB-I00).

7. REFERENCES

[1] J. Powell, "What is temporal art? a persistent question revisited," *Contemporary Aesthetics*, vol. 13, 2015.

[2] W. T. Berry, *Structural functions in music.* Courier Corporation, 1987.

[3] G. T. Toussaint, *The geometry of musical rhythm: What makes a "Good" rhythm good?* CRC Press, 2016.

[4] J. Gillick, A. Roberts, J. Engel, D. Eck, and D. Bamman, "Learning to groove with inverse sequence transformations," in *Proc. of the 36th International Conference on Machine Learning*, California, USA, May 2019, pp. 2269–2279.

[5] O. Lartillot and F. Bruford, "Bistate reduction and comparison of drum patterns," in *Proc. of the 21st International Society for Music Information Retrieval Conference*. Montreal, Canada: ISMIR, Oct. 2020, pp. 318–324.

[6] B. H. Repp and A. Penel, "Auditory dominance in temporal processing: New evidence from synchronization with simultaneous visual and auditory sequences." *Journal of Experimental Psychology: Human Perception and Performance*, vol. 28, no. 5, pp. 1085–1099, 2002.

[7] P. J. Treffner and M. Turvey, "Handedness and the asymmetric dynamics of bimanual rhythmic coordination." *Journal of Experimental Psychology: Human Perception and Performance*, vol. 21, no. 2, p. 318, 1995.

[8] L.-A. Leow, C. Rinchon, and J. Grahn, "Familiarity with music increases walking speed in rhythmic auditory cuing," *Annals of the New York Academy of Sciences*, vol. 1337, no. 1, pp. 53–61, 2015.

[9] M. Leman, D. Moelants, M. Varewyck, F. Styns, L. van Noorden, and J.-P. Martens, "Activating and relaxing music entrains the speed of beat synchronized walking," *PLoS ONE*, vol. 8, no. 7, p. e67932, Jul. 2013.

[10] A. D. Patel and J. R. Iversen, "The evolutionary neuroscience of musical beat perception: the action simulation for auditory prediction (ASAP) hypothesis," *Frontiers in Systems Neuroscience*, vol. 8, May 2014.

[11] J. Podmore, *Jaki Liebezeit: The Life, Theory And Practice Of A Master Drummer*. Unbound, 2020.

[12] E. D. Novotney, "The 3:2 relationship as the foundation of timelines in West African musics," Ph.D. dissertation, University of Illinois at Urbana-Champaign, 1998.

[13] G. Kubik, *Africa and the Blues*. University Press of Mississippi, 1999.

[14] K. Agawu, *The African Imagination in Music*. Oxford University Press, 04 2016.

[15] D. Peñalosa and P. Greenwood, *The Clave Matrix: Afro-Cuban Rhythm : Its Principles and African Origins*, ser. Unlocking clave. Bembe Books, 2012.

[16] R. Bååth, "Subjective Rhythmization: A Replication and an Assessment of Two Theoretical Explanations," *Music Perception*, vol. 33, no. 2, pp. 244–254, 12 2015.

[17] M. A. G. Witek, E. F. Clarke, M. L. Kringelbach, and P. Vuust, "Effects of Polyphonic Context, Instrumentation, and Metrical Location on Syncopation in Music," *Music Perception*, vol. 32, no. 2, pp. 201–217, 12 2014.

[18] D. Gómez-Marín, "Similarity and style in electronic dance music drum rhythms," Ph.D. dissertation, Pompeu Fabra University, Spain, 2018.

[19] D. Gómez-Marín, S. Jordà, and P. Herrera, "Drum rhythm spaces: From polyphonic similarity to generative maps," *Journal of New Music Research*, vol. 49, no. 5, pp. 438–456, 2020.

[20] O. Post and G. Toussaint, "The edit distance as a measure of perceived rhythmic similarity," *Empirical Musicology Review*, vol. 6, no. 3, pp. 164–179, 2011.

[21] M. Moritz, M. Heard, H.-W. Kim, and Y. S. Lee, "Invariance of edit-distance to tempo in rhythm similarity," *Psychology of Music*, vol. 49, no. 6, pp. 1671–1685, 2021.

[22] 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, Sep. 2022. [Online]. Available: 10.5281/zenodo.7088343

REAL-TIME PERCUSSIVE TECHNIQUE RECOGNITION AND EMBEDDING LEARNING FOR THE ACOUSTIC GUITAR

Andrea Martelloni
Queen Mary University of London
a.martelloni@qmul.ac.uk

Andrew P McPherson
Imperial College
andrew.mcpherson@imperial.ac.uk

Mathieu Barthet
Queen Mary University of London
m.barthet@qmul.ac.uk

ABSTRACT

Real-time music information retrieval (RT-MIR) has much potential to augment the capabilities of traditional acoustic instruments. We develop RT-MIR techniques aimed at augmenting percussive fingerstyle, which blends acoustic guitar playing with guitar body percussion. We formulate several design objectives for RT-MIR systems for augmented instrument performance: (i) causal constraint, (ii) perceptually negligible action-to-sound latency, (iii) control intimacy support, (iv) synthesis control support. We present and evaluate real-time guitar body percussion recognition and embedding learning techniques based on convolutional neural networks (CNNs) and CNNs jointly trained with variational autoencoders (VAEs). We introduce a taxonomy of guitar body percussion based on hand part and location. We follow a cross-dataset evaluation approach by collecting three datasets labelled according to the taxonomy. The embedding quality of the models is assessed using KL-Divergence across distributions corresponding to different taxonomic classes. Results indicate that the networks are strong classifiers especially in a simplified 2-class recognition task, and the VAEs yield improved class separation compared to CNNs as evidenced by increased KL-Divergence across distributions. We argue that the VAE embedding quality could support control intimacy and rich interaction when the latent space's parameters are used to control an external synthesis engine. Further design challenges around generalisation to different datasets have been identified.

1. INTRODUCTION

There is increasing interest in deep neural networks for processing audio in real time with sufficiently low latency to be used in musical performance. There is also a drive to provide small self-contained platforms that could perform inference *at the edge*, that is, on a device that can be fitted in a musical interface or a musical instrument [1–3]. Many of the tasks in Music Information Retrieval, such as onset detection [4], playing technique classification [5], timbre transfer [6], re-synthesis of musical information [7] and generative composition [8], find an application in the design of Digital Musical Instruments (DMI) and augmented instruments, as long as the solutions conform to real-time requirements. For Real-Time MIR (RT-MIR), two physical constraints that limit the application of Deep Neural Network (DNN) models are *causality*, implying the inability to look into the future, and *low action-to-sound latency* [9]. Acceptable action-to-sound latency in music performance was found to be 10 ms [10] for percussion instruments, and the latency's *jitter* (the variation) was also found to be a factor in the quality of the interaction [11]. Although there are ways to work around higher latencies, for example by synthesising generic attacks before a specific sound is generated [12], the ideal approach would be to develop a system fulfilling the latency constraints in the first place.

In this work, we investigate RT-MIR for the processing and mapping of guitar body hit sounds to augment the timbral palette of the instrument in percussive fingerstyle. Percussive fingerstyle is an extended guitar technique that uses layered arrangements, alternate tunings and hits on the guitar's body to create the impression of a "one-man band" [13]. Our method relies on deep learning to develop recognition and embedding learning of guitar body percussion. Our model addresses the task of generating representations of body hits according to performers' percussive gestures, separating them by hand part and location. One possible application is to map such a description as parameters for a synthesis engine, such as real-time physics-based synthesis. We adapted an Automatic Drum Transcription (ADT) model based on a Convolutional Neural Network (CNN) to fit the practical constraints of an augmented instrument for percussion. Our longer-term aim is to design a network that not only works as a classifier, but also describes guitar body hits with a set of features unique for each sample, to support control intimacy [14] and try to achieve the same level of nuance afforded by acoustic instruments. To this end, we propose a variation of our model that jointly trains a classifier and a Variational Autoencoder (VAE) [15].

2. BACKGROUND

Percussion DMIs. In opposition to the direct control offered by acoustic percussion, digital percussion instru-

© A. Martelloni, A. P. McPherson, M. Barthet. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** A. Martelloni, A. P. McPherson, M. Barthet, "Real-Time Percussive Technique Recognition and Embedding Learning for the Acoustic Guitar", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

ments have historically afforded indirect control of discrete events [16], with hit dynamics often being the only expressive parameter over individual hits. Jathal [17] provides a detailed description of commercial digital percussion instruments, emphasising the fact that they force the player to adapt their technique to the tool, usually a set of buttons or a zone-based sample trigger. The author also advocates for the design of interfaces that interpret the technique that performers of a particular acoustic instrument have already mastered: traditional techniques will be the first sensorimotor reference that expert players will use for the exploration of DMIs [18], and they have been used in the past as the basis for controllers to navigate synthesiser spaces [19].

Hit classification. One approach is to take data from audio transducers or other sensors for on-the-fly event detection and classification, and the use of that data to trigger the generation of a sound associated to that category. Examples are Turchet *et al.*'s Smart Cajón [20], Jathal's HandSolo [17] and Zamborlin *et al.*'s Mogees [21]. This approach is well supported by music tools and software for machine learning in music such as `bonk~` for Max/PD [22], timbreID's `barkSpec~` and the Wekinator [23]. This has also been applied to the acoustic guitar through the work of Lähdeoja [24] and Stefani *et al.* [3, 25], the latter applying fully-connected DNN layers for multi-class classification of guitar techniques, including percussive ones. No direct attempts have been made to use machine learning to achieve a description beyond classification, especially one that would support Moore's *control intimacy* [14].

Automatic Drum Transcription in MIR. A task related to guitar percussion classification in MIR is Automatic Drum Transcription, the audio-based detection and inference of score notations for percussive parts. Current literature does not only address the Western drum kit, but also other percussion instruments such as the tabla [26]. A recent review [27] reports that, in the current state of the art in ADT, solutions either use non-negative matrix factorisation (NMF) or look into the relationships between hits and tackle the problem with a language model or a recurrent neural network. The most relevant system for our application is based on a CNN that jointly performs event detection and classification for ADT using a sliding buffer of 150 ms [28] and was trained on the MIREX17 drums dataset [29]. Mattur Ananthanarayana (MA) *et al.* fine-tuned the model on a dataset of tabla strokes, noting a resemblance between Kick Drum, Snare Drum and Hi-Hat sounds and the tabla strokes themselves [30].

Real-time DNNs for music performance. Our purpose is not the direct re-synthesis of guitar body hits, but rather the control of the parameters of a synthesis engine. However, we are inspired by the introduction of neural networks as tools for music performance, through solutions such as Neural Audio Synthesis and Neural or Differentiable Digital Signal Processing (DDSP). Bottlenecks and latent spaces have been used with VAEs [31, 32] and autoencoder-like structures [33] for re-synthesis of sounds and timbre transfer, with successful real-time implementations such as RAVE [34] and DDX7 [7]. NN-based solutions have also been applied to model linear [35] and non-linear audio systems, such as guitar amplifiers [36] and stomp-box overdrives [37, 38]. Solutions exist to load an arbitrary neural DSP network into a plugin to be run in a DAW, such as the Neutone VST host by Qosmo [1] and IRCAM's `nn~` [2] Max/PD external.

Tool	Latency
`bonk~` (Puckette [22])	6 ms
Stefani *et al.* [3]	20 ms
RAVE (Caillon *et al.* [34])	DAW-defined
Mogees (Zamborlin [21])	23 ms
HandSolo (Jathal [17])	17 ms
Tabla stroke classifier (MA *et al.* [30])	150 ms

Table 1: Reported buffer values for detection and inference for some of the works and studies cited in this section.

Latency. The impact of latency and jitter in music performance systems was investigated, for example, by McPherson *et al.* [11]. Table 1 reports the measurements published by the authors cited so far on the latency of their tools, specifically the duration of analysis and inference rather than audio input-to-output latency, which is system-dependent more than algorithm-dependent. Most tools that are meant for real-time use achieve latencies in the region of 20 ms, which exceeds Wessel and Wright's 10 ms ceiling for musical instruments [10].

Challenges in rich representation. Gesture classification toolkits like `bonk~` have been deployed in many music-making interfaces, including guitars [24], but they were shown to make percussive guitar performers uneasy owing to the chance of misclassification for ambiguous or unexpected inputs [39]. Standard classifiers also do not represent subtle variability within gestural categories, leading to a small gestural *bottleneck* [18]. Related studies in Human-Computer Interaction (HCI) also promote the design of DMIs sensitive to the *micro-scale* of musical actions, the scale of differences across gestures of the same category [40]; authors have suggested that rich and controllable behaviour could be as important as high classification accuracy for creative applications [41]. Dimensionality reduction of input representations through VAEs, as performed for example by RAVE, could help investigate these rich dimensions.

3. METHODOLOGY

3.1 From taxonomy to datasets

This work builds upon our two prior studies on the investigation of the technique of percussive fingerstyle [13] and the design of a prototype augmented guitar to optimally capture those techniques [39]. Those observations firstly

[1] `https://neutone.space/plugin/`
[2] `https://github.com/acids-ircam/nn_tilde`

	Input Features	CNN Type	Bottleneck
TablaCNN	80-band Mel	2-layer 2D. Kernel size: 1x7, 1x3	128 dimensions, reduced through PCA
PercCNN	512-bin FFT decimated to 64 bins	3-layer 1D. Kernel size: 6, 5, 5	2 dimensions
PercVAE	512-bin FFT decimated to 64 bins	3-layer 1D. Kernel size: 6, 5, 5	2 dimensions (μ + σ)

Table 2: Differences across network architectures used.

	Hand Part	Location	In networks
2-class	Kick - K (heel), Non-Kick - NK (all others)	None	*TablaCNN, PercCNN, PercVAE*
4-class	Heel - H, Thumb - T, Fingers - F, Nails - N	None	*TablaCNN, PercCNN, PercVAE*
4-class + 5-class (Hierarchical)	Heel - H, Thumb - T, Fingers - F, Nails - N	Soundhole, Upper Bout, Lower Bout, Upper Side, Lower Side	*TablaCNN, PercCNN*

Table 3: Output layers mapped to guitar body percussion taxonomy.

led to the creation of a *taxonomy of guitar body percussion*, inspired by the work by Goddard on the taxonomy of bass playing techniques [42]:

$$\begin{cases}\text{hit}\\ \text{scrape}\end{cases} \text{ the guitar with } \begin{cases}\text{heel}\\ \text{thumb}\\ \text{fingers}\\ \text{nails}\end{cases} \text{ at the } \begin{cases}\text{soundhole}\\ \text{upper bout}\\ \text{lower bout}\\ \text{upper side}\\ \text{lower side}\end{cases}$$

This was used to create the labelled dataset *GPercRep* by producing 50 examples (one hit per second) of each combination of taxonomy attributes, excluding those that are ergonomically impossible, e.g. reaching the lower sides of the body with the heel of the hand. This leads to an imbalanced but ecologically valid dataset [43]. Each combination was repeated at four dynamics levels (*p*, *mp*, *mf*, *f*). All recordings were made by the first author on the guitar built for [39], which has a six-channel output made out of one magnetic pickup and five piezo sensors on each of the locations (soundhole, etc..., see taxonomy above). The guitar had 12-53 gauge strings in standard tuning, muted with the left hand, and the hits were played with the bare right hand. After excluding scrapes from the analysis, as they require a time-based gesture follower, the dataset has 3,157 examples extracted from 52 minutes and 37 seconds of audio at 44.1 kHz. The dataset is currently not public.

3.2 Network architectures

The baseline model for our experiments is an adaptation of the tabla transcription model proposed in MA *et al.* [30]. This network processes three stacked spectrograms with different time/frequency resolutions on a window of 150 ms. Each frame of the spectrogram has an 80-bin Mel representation of a window.

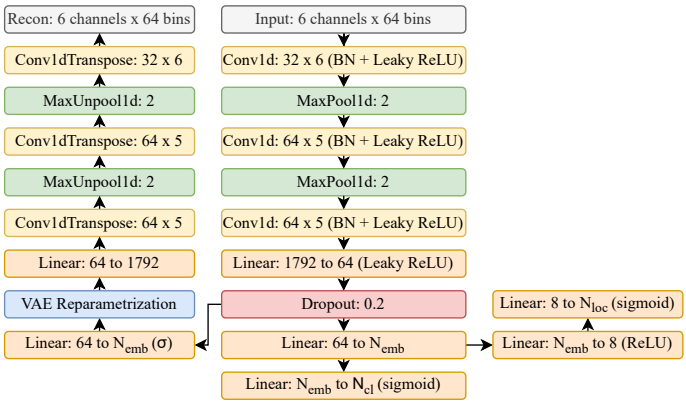


Figure 1: Architecture of *PercCNN*. The extra layers for location classification are on the right-hand side, the decoder of *PercVAE* on the left. $N_{emb} = 2$, $N_{loc} = 5$, $N_{cl} = 2$ or 4.

To adapt this network to real-time requirements we constrained the input window to be 512 samples, or 11.6 ms. Our adaptation (*TablaCNN*) receives one window of six single Mel-frequency spectra, one for each pickup of the prototype. A further modification (*PercCNN*) processes down-sampled FFT features through three one-dimensional convolutional layers and a bottleneck layer of two dimensions before the output (Figure 1). To perform dimensionality reduction jointly with classification, we implemented reparametrisation from the bottleneck layer and a decoder mirroring the encoding CNN (*PercVAE*).

3.3 Output classes

The labels according to the guitar body taxonomy were simplified to: (i) a 2-class scenario with "kicks" (heel hits, in reference to kick drum sounds that heel hits are supposed to imitate) and "non-kicks"; a 4-class output implementing all hand parts; (ii) 4-class hand part plus another 5-class output trained on hit location on the body (hierarchical output). The hierarchical output was not implemented on the VAE. This gave us a total of eight network configurations. Tables 2 and 3 illustrate differences between network architectures, and the mappings between the taxonomy in Section 3.1 and the output layers.

Loss functions used were Binary Cross-Entropy for 2-way classification, Cross-Entropy for 4-way classification, and a sum of two equally weighted Cross-Entropies for hierarchical classification (hand part and location). The VAE used the following loss function, where $\gamma = 0.001$ and $\beta = 3$ after hyperparameter search, and BCE replaced by Cross-Entropy in the four-class model:
$L_{VAE} = BCE + \gamma(MSE_{Recon} + \beta KLD)$

3.4 Training, data augmentation, cross-validation

All networks were trained on the *GPercRep* dataset with hold-out cross-validation: a stratified 20% of the shuffled dataset was reserved for testing, whereas the remainder of

	GPercRep					*GPercHeel*	*GPercPat*		
	K	**NK**	**F**	**N**	**W/Avg**	**Recall**	**K**	**NK**	**W/Avg**
TablaCNN - 2-class	97.46	99.42			99.05	85.69	44.44	85.07	74.56
PercCNN - 2-class	**98.33**	**99.61**			**99.37**	**91.68**	0.00	85.14	63.10
PercVAE- 2-class	97.87	99.51			99.20	85.02	0.00	85.14	63.10
	H	**T**	**F**	**N**	**W/Avg**				
TablaCNN - 4-class	**97.48**	**91.06**	**89.81**	**94.46**	**92.92**	91.35	35.71	87.32	73.97
PercCNN - 4-class	94.61	78.95	80.86	93.26	86.92	69.05	**74.29**	**93.33**	**88.40**
PercVAE - 4-class	97.44	90.30	87.44	93.16	91.63	81.86	0.00	85.14	63.10
TablaCNN - Hierarchical	96.61	92.24	89.59	93.99	92.77	89.18	23.08	86.11	69.80
PercCNN - Hierarchical	95.73	82.05	87.60	94.18	90.12	69.55	0.00	85.14	63.10

Table 4: F-Measure (as percent) for each network on the three test datasets (hold-out of *GPercRep*, *GPercHeel* and *GPercPat*). **H** = heel, **K** = kick, **T** = thumb, **NK** = non-kick, **F** = fingers, **N** = nails, **W/Avg** = weighted average.

the examples was used for training (80%) and validation (20%). All networks were trained with an Adam optimiser for 100 epochs with a batch size of 128, saving the model with the highest accuracy on the test set. We trained with the following data augmentation functions: high-pass at 80 Hz, high-pass at 160 Hz, $tanh()$ waveshaping distortion with gain of 5, phase inversion and random changes in gain between the six channels of each example. Those functions are meant to represent the different input impedances and different gains of other audio preamplifiers.

Further to the *GPercRep* dataset, generalisation was checked by performing cross-dataset evaluation [44]. We recorded a snippet of real-world guitar percussion patterns (*GPercPat*); musically coherent patterns, still with no tonal sounds, were played, rather than hits repeated every second. Acquisition was done with a different audio interface, which led to a different combination of gains and frequency responses across channels in the input audio. The dataset was annotated only with a "kick" and "non-kick" label, leading to 85 hits in one minute of audio.

Testing on the *GPercPat* dataset highlighted a bias in our networks against heel hits or "kicks". An explanation was thought to be the lack of balance across classes in the dataset, however the issue was not mitigated by balancing the dataset, ensuring the same number of examples for each category. To gather further information about this phenomenon, we created a third dataset consisting exclusively of 601 heel hits, acquired and labelled with the same taxonomy as *GPercRep*: this will be called *GPercHeel*.

3.5 Evaluation metrics

The classification performance of the networks was evaluated with Precision, Recall and F-measure for each category, as a 2-class or 4-class problem (see Table 3).

We also wanted to quantitatively investigate the quality of the network's embeddings. Thus, we made subsets of the data in *GPercRep* according to each label the network was trained on (kick VS non-kick or hand part), and for each category, we drew distributions for each of the other parts in *GPercRep*'s taxonomy: for example, we divided non-kicks according to their location or their dynamics. Then we calculated the KL-divergences between the probability distributions of each sub-category. The hypothesis behind this method is that, if the embeddings do not carry any meaningful information beyond the classes that the network was trained on, the distributions will overlap and their KL-divergences will be small and noisy. If, on the other hand, different hit properties lead to different positions in the embeddings, KL-divergences will be different across sub-categories and the sub-categories will be arranged following a certain order of similarity.

Reconstruction metrics for the VAE were not evaluated beyond their inclusion in the loss function. Future work could focus on the correlation between better reconstruction and better separation of each feature.

4. EVALUATION

4.1 Classification

Table 4 contains the F-Measure for the predictions of each network, with the three test datasets. In the case of *GPercHeel*, only the Recall is reported; the Precision is always 1, as all hits are heel hits and there cannot be false positives (non-heel hits classified as heel hits).

2-Class discrimination. All networks are able to precisely discriminate between kicks and non-kicks with an F-measure above 99%. The *GPercHeel* dataset shows much reduced but still effective classification, especially with *PercCNN*. However, the test on *GPercPat* exposes a generalisation problem: despite performing data augmentation during training, all networks show a bias toward non-kicks. *PercCNN* and *PercVAE* return only non-kicks in the dataset, despite the two classes being visually separable when data points are extracted and plotted from their embeddings (not pictured). This result may suggest that the networks still overfit to the extent that they are very sensitive to the way that the data is acquired.

4-class discrimination. Uniformly across the tests, the networks yield an F-measure around 90% for *GPercRep*. The introduction of the classification by location (in the two *Hierarchical* networks) does not affect the score of the hand-part classifier. *GPercHeel* yields a similar Recall score, although higher in the case of *TablaCNN* networks. Interestingly, the weakest model in *GPercRep*, the 4-class

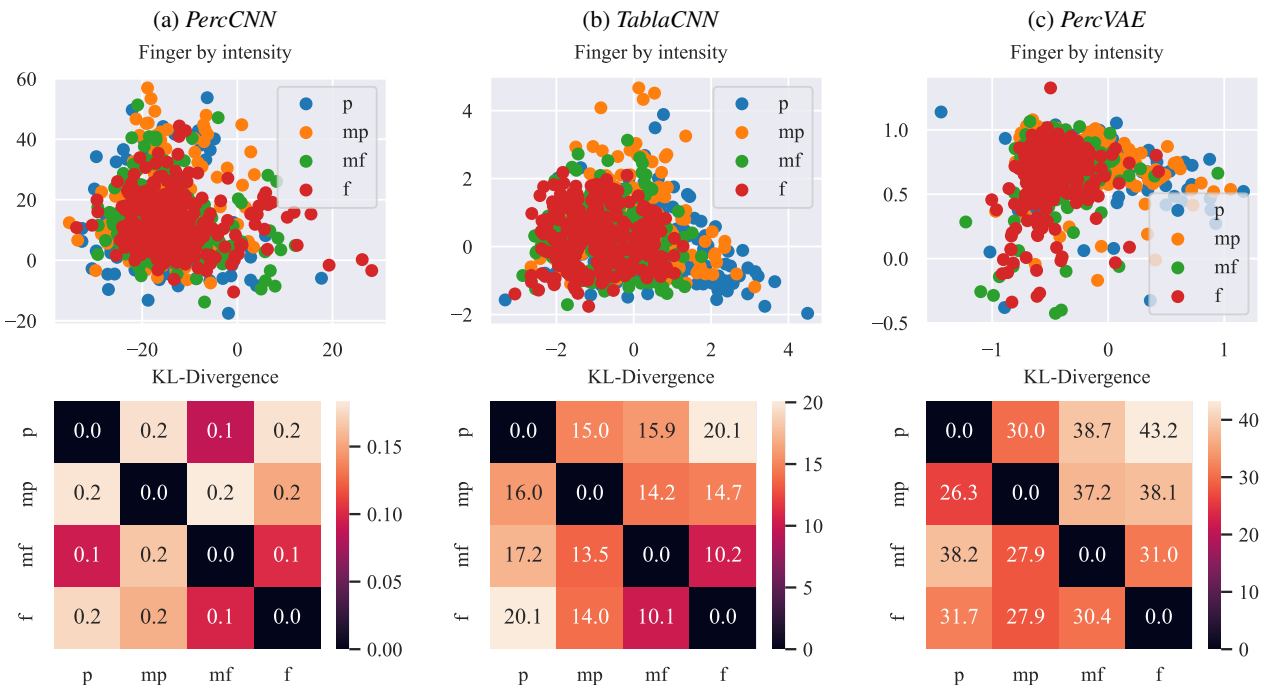


Figure 2: Embeddings from *GPercRep*: example with finger hits labelled by dynamics, with matrix of KL divergence across the distributions of each dynamic level.

PercCNN, ends up being the best model in *GPercPat*, although an F-measure of 74% for kick hits may still not be satisfactory in musical performance. *PercVAE* shows better performance than *PercCNN*, although it still fails to generalise to *GPercPat* and defaults to flagging all events as non-kicks.

The F-Measures in our results are higher on average than the ones found in MA *et al.*'s work on tabla hit transcription [30]. At the same time, our results fall below the 95% accuracy achieved by Stefani [45] with an 8-class discriminator on guitar techniques, and the 97% by Jathal [17] on the three-way discriminator for tabletop drumming. These results, however, are not directly comparable as the figures refer to different datasets.

4.2 Computation times and latency

	Avg	Std Dev
PercCNN	0.496	0.332
TablaCNN	0.422	0.496
PercCNN in Max	12.675	1.132
System (**PercCNN**)	22.310	0.670
System (no NN)	9.922	0.020

Table 5: Computation times in μs of both networks measured through TorchScript in a C++ wrapper, then end-to-end within Max and with an external analogue excitation.

Our models all require a fixed 11.6 ms input buffer to populate the input window after an event is detected, for example through a time-based attack detector [46]. TorchScript was used to wrap the two-class (*PercCNN* and *TablaCNN*) into a C++ test routine and a Max/MSP external for a synthetic soak test and real-world latency tests on a laptop with an Intel i7-8665U CPU running Windows (Table 5).

PercCNN and *TablaCNN* have comparable latencies in the synthetic test. They both execute in less than half a millisecond on average when called 10,000 times. The real-world latency measured manually within Max/MSP (over 30 examples) reports a value that is consistent with the 11.6 ms window plus the synthetic timing reported above, with the attack detection not introducing much further latency or jitter. The low-power laptop used requires an audio buffer size of 256 samples to comfortably run **PercCNN** in real time alongside a suitable synthesis engine: the total system latency jumps to 22 ms when probing with a Bela [3] board attached to the laptop's sound card (averaged over 500 examples). Input and output buffers can be greatly reduced on ad-hoc hardware or software.

4.3 Embeddings

As introduced in Section 3.5, the distribution of subclasses of the taxonomy within each class was explored in the embeddings of each network. In addition to a visual and qualitative inspection of the distribution through scatter plots, KL-Divergence is used here as a similarity metric to measure the distance between distributions. In the following analysis, we will take finger hits divided according to dynamics as an example, but our observations are valid for all other hand parts, and versus hit location (e.g. heel hits

[3] `https://bela.io`

divided by body location).

Classifier embedding. When *PercCNN* is only trained as a classifier, the four dynamic points overlap in the embeddings (Figure 2a). KL-Divergences range between $2 \cdot 10^{-5}$ and 0.2, so this range of numbers will be used as a baseline for the interpretation of further results. *PercCNNHierarch*, trained to discriminate according to hand part and location, shows very precise segmentation of hit locations but no meaningful segmentation by dynamics (not pictured [4]).

Embedding with PCA. *TablaCNN*'s embeddings are not a bottleneck within the network itself, but they are calculated through PCA on the 128-dimensional dense layer. Principal Component Analysis is shown to disentangle some of the other features in the dataset, as the dynamics subclasses are distributed along a right-to-left gradient (Figure 2b). The KL-Divergence across those distributions reaches a maximum of 20.1.

Embedding/VAE latent space. *PercVAE* shows a similar but more pronounced subdivision in the 2-dimensional latent space. The right-to-left gradient is visible but the KL-Divergence is much greater at a maximum of 43.2. The KL-Divergence values steadily increase from *p* to *f*, more evidently than in the embeddings extracted via PCA. This was noticeable also when hits were segmented by location (not pictured): for example, Lower Side had a KL-Divergence of 30.8 versus Upper Side, 31.7 vs Lower Bout, 38.7 vs Upper Bout, and 40.7 vs Soundhole.

5. DISCUSSION

The evaluation shows that all our models act as very accurate 2-class classifiers. Even though classification accuracy is not as high as in other situations (with different datasets), the simplicity of our models and the 11.6 ms input buffer makes them faster than those systems, and well suited for implementation on an edge device.

Challenges. The main issue arising from our evaluation is the poor generalisation to our *GPercPat* dataset. Still, we have anecdotal evidence that these networks do not behave like poor classifiers in the real-world context of musical performance with our augmented guitar prototype, the HITar [5] . The 2-class *PercCNN* was coupled with a time-domain hit detector and made to run in real time; its continuous output probability was mapped to a linear interpolation of parameters on the modal synthesis engine MetaSynth by CNRS-AMU PRISM [47]; the signal chain was connected to a different guitar (same make and model) to the one the network was trained with; the network is able to reliably adapt synthesis parameters even when used by players other than the main author. There is scope to expand the training and the evaluation by involving more guitars, more players and different data augmentation techniques. However, the augmented guitar that we built allows us to pursue a further type of *behavioural* evaluation with guitar players. In particular, musicians performing in real time may adapt their gestures until they reliably produce a desired set of outcomes, something not possible with prerecorded data. A study on the performance of guitar players with different network configurations running on the augmented guitar prototype will help investigate the degree to which the musicians can adapt to the expectation of the network; such a study would continue our work in [39].

Support for rich interaction. We observed that *PercVAE* is able to encode differences in hit dynamics and location within the embeddings without being trained to discriminate between them; rather than separating them with decision boundaries like *PercCNNHierarch*, each subcategory overlaps neighbouring subcategories, providing a smooth transition that could map well to continuous quantities such as dynamics or location on a surface. The use of a bottleneck layer is also a more efficient solution than PCA, as performing PCA would require extra matrix computation that was not captured in the timings measured at Section 4.2. The parameters of a synthesis engine such as MetaSynth could be controlled not just by the categorical output of the discriminator, but also by the latent representation of the VAE, either directly or through a transform. A mapping function could be designed between the embeddings and synthesis parameters, or the embedding vectors could be exposed directly to synthesisers as MIDI Polyphonic Expression (MPE) [48] controls.

6. CONCLUSIONS

We presented three adaptations of Automatic Drum Transcription for guitar body percussion classification and embedding learning, to support real-time music performance and the augmentation of an acoustic guitar through Deep Neural Networks. We chose and simplified a model for ADT that was shown to be effective in the detection of tabla strokes; a variant was also proposed which supports high-level continuous feature representation through the use of embeddings jointly trained as a Variational Autoencoder's latent space. All network configurations were trained on a dataset of percussive fingerstyle hits acquired *ad hoc*, and they were tested on a hold-out portion of that dataset plus two other datasets of similar material. The networks performed very well on a simplified 2-class discrimination, and comparably to the state of the art on the full 4-class stroke classification with smaller latency. However, they generalise poorly on a dataset that was recorded with different computer equipment. The embeddings were analysed both qualitatively and quantitatively through KL-Divergence between subclasses in the taxonomy; they show that the network encodes some information beyond the categories with which it was trained. We argue that this information can be used to support richness in musical interaction with digital and augmented instruments based on DNN analysis.

[4] All pictures of embeddings available at `https://github.com/iamtheband/martelloni_et_al_ismir2023`

[5] Performance of the HITar at the Guthman Musical Instrument Competition 2023: `https://www.youtube.com/live/NPtHGYH0JV0?t=1150`

HITar's Linktree: `https://linktr.ee/hit4r`

7. ACKNOWLEDGEMENTS

This work was supported by UK Research and Innovation's CDT in AI & Music [grant number EP/S022694/1] and by PRISM Laboratory (CNRS, Aix-Marseille University).

8. REFERENCES

[1] L. Turchet, C. Fischione, G. Essl, D. Keller, and M. Barthet, "Internet of musical things: Vision and challenges," *IEEE access : practical innovations, open solutions*, vol. 6, pp. 61 994–62 017, 2018.

[2] T. Pelinski, V. Shepardson, S. Symons, F. S. Caspe, A. L. Benito Temprano, J. Armitage, C. Kiefer, R. Fiebrink, T. Magnusson, and A. McPherson, "Embedded AI for NIME: Challenges and Opportunities," in *NIME*, Jun. 2022.

[3] D. Stefani, S. Peroni, and L. Turchet, "A comparison of deep learning inference engines for embedded real-time audio classification," in *Int. Conf. on Digital Audio Effects*, vol. 3, 2022, pp. 256–283.

[4] S. Böck, A. Arzt, F. Krebs, and M. Schedl, "Online real-time onset detection with recurrent neural networks," in *Proceedings of the 15th International Conference on Digital Audio Effects (DAFx-12), York, UK*. sn, 2012, pp. 17–21.

[5] C.-Y. Wang, P.-C. Chang, J.-J. Ding, T.-C. Tai, A. Santoso, Y.-T. Liu, and J.-C. Wang, "Spectral–temporal receptive field-based descriptors and hierarchical cascade deep belief network for guitar playing technique classification," *IEEE Transactions on Cybernetics*, vol. 52, no. 5, pp. 3684–3695, 2020.

[6] D. K. Jain, A. Kumar, L. Cai, S. Singhal, and V. Kumar, "ATT: Attention-based timbre transfer," in *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2020, pp. 1–6.

[7] F. Caspe, A. McPherson, and M. Sandler, "DDX7: Differentiable FM Synthesis of Musical Instrument Sounds," in *ISMIR*, 2022.

[8] 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.

[9] D. Stefani and L. Turchet, "On the challenges of embedded real-time music information retrieval," in *DAFx*, 2022.

[10] D. Wessel and M. Wright, "Problems and Prospects for Intimate Musical Control of Computers," *Computer Music Journal*, vol. 26, no. 3, p. 13, 2002.

[11] A. McPherson, R. Jack, and G. Moro, "Action-sound latency: Are our tools fast enough?" in *NIME*, 2016, pp. 20–25.

[12] D. Stowell and M. D. Plumbley, "Delayed Decision-making in Real-time Beatbox Percussion Classification," *Journal of New Music Research*, vol. 39, no. 3, pp. 203–213, Sep. 2010.

[13] A. Martelloni, A. McPherson, and M. Barthet, "Percussive fingerstyle guitar through the lens of NIME: An interview study," in *NIME*, Jul. 2020, pp. 440–445.

[14] F. R. Moore, "The Dysfunctions of MIDI," *Computer Music Journal*, vol. 12, no. 1, p. 19, 1988.

[15] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," *arXiv:1312.6114 [cs, stat]*, May 2014.

[16] M. M. Wanderley, "Gestural control of music," in *International Workshop Human Supervision and Control in Engineering and Music*, 2001, pp. 632–644.

[17] K. Jathal, "Real-Time Timbre Classification for Tabletop Hand Drumming," *Computer Music Journal*, vol. 41, no. 2, pp. 38–51, Jun. 2017.

[18] R. H. Jack, T. Stockman, and A. McPherson, "Rich gesture, reduced control: The influence of constrained mappings on performance technique," in *Proceedings of the 4th International Conference on Movement Computing - MOCO '17*, 2017, pp. 1–8.

[19] P. A. Tremblay and D. Schwarz, "Surfing the Waves: Live Audio Mosaicing of an Electric Bass Performance as a Corpus Browsing Interface," in *NIME*, 2010.

[20] L. Turchet, A. McPherson, and M. Barthet, "Real-Time Hit Classification in a Smart Cajón," *Frontiers in ICT*, vol. 5, Jul. 2018.

[21] B. Zamborlin, "Studies on customisation-driven digital music instruments," Ph.D. dissertation, Goldsmiths, University of London.

[22] M. S. Puckette, T. Apel, and D. D. Zicarelli, "Real-time audio analysis tools for Pd and MSP," in *ICMC*, 1998, p. 5.

[23] R. Fiebrink and P. R. Cook, "The Wekinator: A system for real-time, interactive machine learning in music," in *ISMIR*, 2010.

[24] O. Lahdeoja, "Augmenting Chordophones with Hybrid Percussive Sound Possibilities," in *NIME*, 2009, p. 4.

[25] D. Stefani and L. Turchet, "Demo of the TimbreID-VST Plugin for Embedded Real-Time Classification of Individual Musical Instruments Timbres," in *FRUCT*, 2020, p. 3.

[26] K. Narang and P. Rao, "Acoustic Features for Determining Goodness of Tabla Strokes," in *ISMIR*, 2017.

[27] 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.

[28] C. Jacques and A. Roebel, "Automatic drum transcription with convolutional neural networks," in *DAFx*, 2018, p. 8.

[29] R. Vogl and P. Knees, "Mirex submission for drum transcription 2018," in *MIREX Extended Abstracts, 19th International Society for Music Information Retrieval Conference (ISMIR)*, 2018.

[30] R. MA, A. Bhattacharjee, and P. Rao, "Four-way classification of tabla strokes with models adapted from Automatic Drum Transcription," in *ISMIR*, 2021.

[31] P. Esling, A. Chemla–Romeu-Santos, and A. Bitton, "Generative timbre spaces: Regularizing variational auto-encoders with perceptual metrics," *arXiv:1805.08501 [cs, eess]*, Oct. 2018.

[32] A. Bitton, P. Esling, and A. Chemla-Romeu-Santos, "Modulated Variational auto-Encoders for many-to-many musical timbre transfer," *arXiv:1810.00222 [cs, eess]*, Sep. 2018.

[33] J. Engel, L. Hantrakul, C. Gu, and A. Roberts, "DDSP: Differentiable Digital Signal Processing," *arXiv:2001.04643 [cs, eess, stat]*, Jan. 2020.

[34] A. Caillon and P. Esling, "RAVE: A variational autoencoder for fast and high-quality neural audio synthesis," *arXiv:2111.05011 [cs, eess]*, Nov. 2021.

[35] J. T. Colonel, C. J. Steinmetz, M. Michelen, and J. D. Reiss, "Direct design of biquad filter cascades with deep learning by sampling random polynomials," *arXiv:2110.03691 [cs, eess]*, Oct. 2021.

[36] F. Eichas, S. Möller, and U. Zölzer, "Block-oriented Gray Box Modeling of Guitar Amplifiers," in *DAFx*, 2017, p. 8.

[37] J. T. Colonel, M. Comunità, and J. Reiss, "Reverse Engineering Memoryless Distortion Effects with Differentiable Waveshapers," in *AES Convention*, Oct. 2022, p. 10.

[38] E.-P. Damskagg, L. Juvela, and V. Valimaki, "Real-Time Modeling of Audio Distortion Circuits with Deep Learning," in *SMC*, 2019.

[39] A. Martelloni, A. McPherson, and M. Barthet, "Guitar augmentation for Percussive Fingerstyle: Combining self-reflexive practice and user-centred design," in *NIME*, Jun. 2021.

[40] J. Armitage, T. Magnusson, and A. McPherson, "Studying subtle and detailed Digital Liutherie: Motivational contexts and technical needs," in *NIME (Accepted)*, 2023.

[41] G. Vigliensoni, P. Perry, and R. Fiebrink, "A Small-Data Mindset for Generative AI Creative Work," in *CHI*, 2022.

[42] C. Goddard, "Virtuosity in computationally creative musical performance for bass guitar," Ph.D. dissertation, Queen Mary University of London, 2021.

[43] D. D. Ramyachitra and P. Manikandan, "Imbalanced Dataset Classification and Solutions: A Review," *International Journal of Computing and Business Research*, vol. 5, no. 4, 2014.

[44] A. Livshin and X. Rodet, "The importance of cross database evaluation in musical instrument sound classification: A critical approach." in *ISMIR*, Jan. 2003.

[45] D. Stefani, S. Peroni, and L. Turchet, "A Comparison of Deep Learning Inference Engines for Embedded Real-time Audio classification," p. 9, 2022.

[46] Luca Turchet, "Hard Real-Time Onset Detection Of Percussive Sounds," in *Proceedings of the 21st International Conference on Digital Audio Effects (DAFx-18)*, 2018, p. 9.

[47] S. Conan, E. Thoret, M. Aramaki, O. Derrien, C. Gondre, R. Kronland-Martinet, and S. Ystad, "Navigating in a space of synthesized interaction-sounds: Rubbing, scratching and rolling sounds," p. 9, 2013.

[48] T. Romo, "MIDI: A Standard for Music in the Ever Changing Digital Age," Capstone Project, California State University, 2018.

IteraTTA: AN INTERFACE FOR EXPLORING BOTH TEXT PROMPTS AND AUDIO PRIORS IN GENERATING MUSIC WITH TEXT-TO-AUDIO MODELS

Hiromu Yakura
University of Tsukuba
Tsukuba, Japan
hiromu.yakura@aist.go.jp

Masataka Goto
National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, Japan
m.goto@aist.go.jp

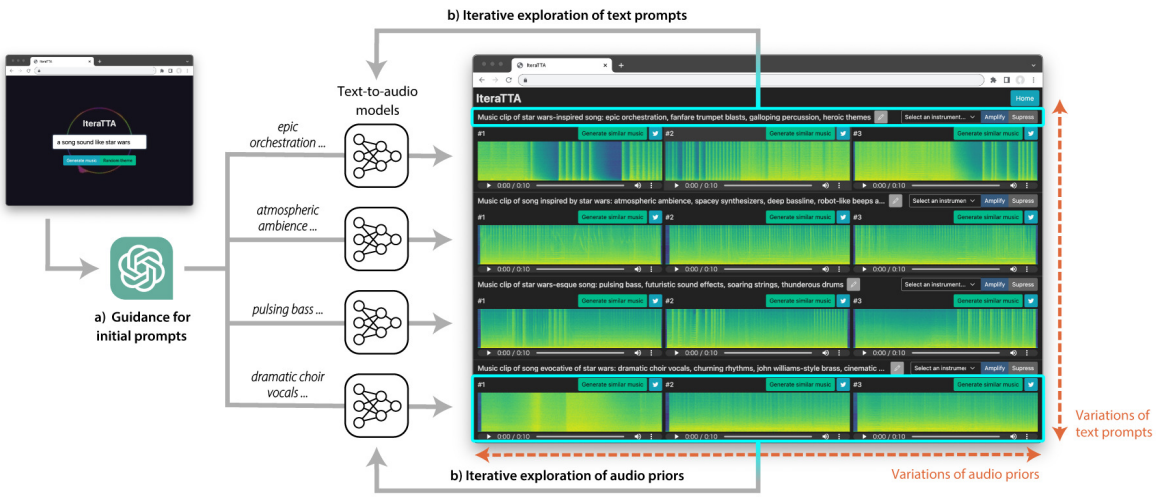


Figure 1. IteraTTA is an interface dedicated for allowing novice users to show their creativity in text-to-audio music generation processes. It provides a) computational guidance for constructing initial prompts and b) dual-sided iterative exploration of text prompts and audio priors.

ABSTRACT

Recent text-to-audio generation techniques have the potential to allow novice users to freely generate music audio. Even if they do not have musical knowledge, such as about chord progressions and instruments, users can try various text prompts to generate audio. However, compared to the image domain, gaining a clear understanding of the space of possible music audios is difficult because users cannot listen to the variations of the generated audios simultaneously. We therefore facilitate users in exploring not only text prompts but also audio priors that constrain the text-to-audio music generation process. This dual-sided exploration enables users to discern the impact of different text prompts and audio priors on the generation results through iterative comparison of them. Our developed interface, IteraTTA, is specifically designed to aid users in refining text prompts and selecting favorable audio priors from the generated audios. With this, users can progressively reach their loosely-specified goals while understanding and exploring the space of possible results. Our implementation and discussions highlight design considerations that are specifically required for text-to-audio models and how interaction techniques can contribute to their effectiveness.

© H. Yakura and M. Goto. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** H. Yakura and M. Goto, "IteraTTA: An interface for exploring both text prompts and audio priors in generating music with text-to-audio models", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.*, Milan, Italy, 2023.

1. INTRODUCTION

Recent advances in generative machine learning techniques open up novel ways for a diverse group of individuals to engage in creative processes [1, 2]. Specifically, music generation models can foster creative expression among novice users, who may not necessarily possess formal musical knowledge [3, 4]. Consequently, several approaches have been proposed to enable users to control various musical attributes of generated audios, such as specifying the note or rhythm density [5, 6] and chord progression [7–9]. Text-to-audio models [10, 11] are promising in terms of allowing users who are not familiar with the concepts of such musical attributes to generate their own sounds.

Nevertheless, there are still several gaps toward deploying such models to support the creativity of novice users. For example, the models rely on annotated labels of music clips presented in their training datasets [12–14], which primarily consist of musical descriptions such as genres, instruments, and moods. Therefore, providing such infor-

mation as a text prompt is crucial for enabling fine-grained control over generated music audios. However, this may prove challenging for novice users due to disparities in artistic vocabulary among individuals with varying levels of musical knowledge [15]. Experimentally, it has been suggested that non-musicians tend to rely more on abstract concepts, such as the pleasantness or complexity of music, when appreciating musical pieces [16], which may pose difficulties in fully exploring various text prompts.

Moreover, understanding the space of possible results is also challenging, particularly when compared to the use of text-to-image models. In text-to-image generation, users can look over various generation results at a glance, which fosters their understanding of the space and helps them decide on directions to explore [17]. From the perspective of explainable AI (XAI), we can say that such results serve as *explanations by example* [18] because the results implicitly invite the users to infer the behavior of the models. However, in text-to-audio generation, users cannot simultaneously listen to multiple generation results, thus impeding their comprehension and ability to efficiently explore the space. These points imply that specific design considerations are necessary to fully leverage the potential of text-to-audio models and exploring them would also provide a new perspective in terms of XAI.

In this paper, we introduce IteraTTA, an interface dedicated to the text-to-audio (TTA) music generation processes of novice users. This interface enables iterative exploration of both text prompts and audio priors, allowing users to gain a comprehensive understanding of the space of possible results by sufficiently constraining the generation processes. We constructed this interface based on our observations and related literature on creativity support, which emphasize the importance of 1) computational guidance for constructing initial prompts and 2) dual-sided iterative exploration of text prompts and audio priors. Moreover, we deployed the interface as a publicly-available Web service and analyzed the diverse ways in which users utilized it in their creative processes. Our results and discussions shed light on ways to utilize models developed in the MIR community to unleash the creativity not only of expert users [19] but also of individuals with varying degrees of musical knowledge.

2. RELATED WORK

2.1 Music Generation Techniques

Music generation has been one of the central topics with the MIR community [20–24], and recently, generative machine learning techniques have been widely employed for this purpose [24, 25]. While methods for symbolic music generation that output MIDI files have been popular [26–32], some methods use generative models to directly output audio, leveraging their expressiveness [33–36]. For example, Jukebox [33] and RAVE [34] use variational autoencoders and autoregressive models trained on large-scale music datasets to generate diverse music audios.

Controllability in music generation has been also emphasized [5–9, 37–39] because it is vital to open up its applications for supporting users' creative processes [40, 41]. For instance, Music FaderNets [5] allows users to modify the rhythm and note densities of generation results, while Music SketchNet [6] enables them to specify pitch contours and rhythm patterns. Wang *et al.* [7] and Dai *et al.* [8] have proposed methods to further constrain the chord progression of generation results. However, as mentioned in Section 1, users are not always familiar with such concepts, and then, they would have difficulties in using these methods to output music audios they want to generate. We acknowledge that some methods [38, 39] provide perceptual control that does not require extensive musical knowledge: emotion-based musical generation. Nevertheless, they are based on Russell's valence-arousal model [42] consisting of four classes, which limits the range of controls and may hamper users' agency [43] when the methods are used to support their creative processes.

In this context, recent text-to-audio models [10, 11] can be an effective tool for such novice users. These models learn the relationship between music audios and their text descriptions (more specifically, latent representations encoded from the descriptions by RoBERTa [44]) and use it to guide results in generating new audios from an inputted text (*i.e.*, text prompt). As RoBERTa can encode text prompts with variable length and content, the models can provide flexible control without requiring specific musical knowledge of rhythm patterns or chord progressions. Moreover, they allow users to constrain generation results not only by text prompts but also by audio priors, ensuring that the results have similar characteristics to the priors. For example, the diffusion model [45] employed by AudioLDM [11] usually uses Gaussian noise for the seed of its generation process, but by using a noise-infused audio prior, we can obtain generation results preserving the characteristics of the provided audio.

Here, text-to-image models that use similar schemes have been shown to unleash the creativity of novice users, allowing them to iteratively explore open-ended variations of text prompts [17] and customize their intermediate results by specifying image prior constraints [46]. Similarly, text-to-audio models can be leveraged to provide users with such iterative exploration or customization. However, we also expect that text-to-audio music generation processes may pose several specific difficulties, as explained in Section 1. Therefore, we explored how interaction techniques can address these challenges by developing an interface dedicated to text-to-audio models.

2.2 Interfaces for Music Generation

There is a series of research on building interfaces to let users interact with music generation techniques effectively [47–53]. MySong [47], for instance, involves a music accompaniment generation model, with which users can control the happiness or jazziness of generation results. Louie *et al.* [49] proposed an interactive interface for novice users so that they can use a symbolic music generation technique with control of happiness or randomness.

The interface also allows users to constrain generation results by providing music priors, which was experimentally confirmed to be effective in iteratively refining the results. Zhou *et al.* [52,53] utilized a user-in-the-loop Bayesian optimization technique to enable novice users to iteratively explore melodies composed by a generative model.

These interfaces underscore the significance of providing controls and supporting iterative exploration in facilitating the creativity of novice users using music generation techniques. Consequently, the provision of recent text-to-audio models to novice users would be highly suitable for this purpose, as they offer more flexible control, compared to using several parameters such as happiness, while also allowing the use of audio priors. Our paper contributes to this series of research by examining design considerations of interfaces for text-to-audio music generation processes, aiming to expand the scope of applications of recent techniques developed in the MIR community.

3. DESIGN REQUIREMENTS

As stated in Section 1, our goal is to leverage text-to-audio models to facilitate the creative expression of novice users regardless of their musical knowledge. To this aim, we embarked upon an examination of potential challenges that these users may encounter during text-to-audio music generation processes and subsequently derived a set of design requirements to address these issues. Guided by the principles of human-computer interaction, we utilized the think-aloud protocol [54, 55] by involving three volunteers who self-reported that they possessed no formal musical training beyond compulsory education. Specifically, we provided the volunteers with access to one of the latest text-to-audio models [11] on Google Colab using its official implementation [1] , which enabled them to provide any text prompts and subsequently listen to three music audios generated from the text prompts. Here, since the remotely-participated volunteers were Japanese speakers recruited via word-of-mouth communication, we told them that they can use DeepL Translator to translate text prompts into English to obtain better results with the model that is mainly trained on the dataset with English text labels [12–14]. They freely used the model for approximately 30 minutes while sharing their screens on a video call and verbalizing their thoughts and feelings. This allowed us to identify the challenges that they encountered and the factors that contributed to these challenges. We then conducted semi-structured interviews to validate the challenges identified and to gain further insight into the reasons behind them. Their responses were analyzed based on open coding [56], which yielded the following design requirements in line with the existing literature on creativity support.

3.1 Computational guidance for constructing initial prompts

We observed that the volunteers frequently encountered difficulty in formulating appropriate text prompts to initiate their use of the model. For example, one volunteer entered the phrase "a song sounds like star wars," resulting in audio containing a battle cry with a space-like sound effect. This can be attributed to the characteristics of the text labels in the dataset used to train the model [12–14]. Specifically, the labels of music clips consist primarily of musical descriptions such as genres, instruments, and moods, like: "An orchestra plays a happy melody while the strings and wind instruments are being played [14]." Therefore, providing such a description would be essential to ensure that the model trained on the dataset generates music audio as intended. The volunteer was unable to generate music-like audio until he attempted several prompts and finally entered "solemn music starting with a trumpet fanfare."

In the context of creativity support, two underlying factors could explain the aforementioned observation. First, an inherent gap in artistic vocabulary exists between expert and novice users [15]. Without deep musical knowledge, it can be challenging to conceive a precise description of music audios. Additionally, novice users often have loosely-specified goals when starting a creative endeavor [57–59]. They refine their objectives gradually by exploring the space of possible results through iterative exploration [60, 61]. However, the dependency of text-to-audio models on precise descriptions of clearly-defined goals makes it difficult for novice users to initiate such exploration. This suggests that supporting them computationally in constructing initial prompts could potentially facilitate the creativity of novice users.

3.2 Dual-sided iterative exploration of text prompts and audio priors

We also observed that the volunteers encountered challenges in efficiently exploring the generated results. One volunteer who had prior experience with text-to-image models mentioned the point, as:

> *"Unlike text-to-image models, comparing various results at a glance was difficult with the text-to-audio model. So, finding a text prompt reflecting my intention most faithfully became much tough."*

In other words, iteratively trying different text prompts would not necessarily assist users in comprehending the space of potential results, although it is necessary for novice users to refine their loosely-specified goals [60, 61]. Therefore, users cannot determine which direction would be closest to their goals and what text prompt to try next. Another volunteer mentioned an issue he faced, as:

> *"I once found a generation result with a good melody, but I wanted to change its tone. So, I added 'with a flute' to its text prompt and regenerated. However, the melody was then completely changed, which was frustrating."*

This implies that we need to let users utilize not only text prompts but also audio priors to constrain the tune of generation results. In sum, supporting the creativity of novice users in text-to-audio music generation processes requires enabling them to efficiently explore variations of both text

[1] https://github.com/haoheliu/AudioLDM

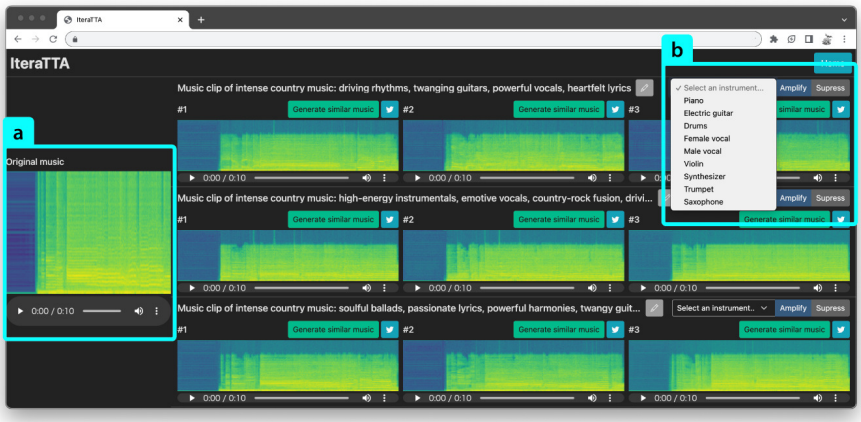


Figure 2. To facilitate the exploration of text prompts and audio priors, IteraTTA allows a) comparison of generation results with an audio prior and b) instant edit of a text prompt.

prompts and audio priors, allowing them to iteratively refine their goals by understanding the space of possible results. This demands us to develop an interface specifically tailored for text-to-audio models to provide such dual-sided exploration of text prompts and audio priors.

4. IteraTTA

Based on the above design requirements, we present IteraTTA, a dedicated interface for text-to-audio music generation processes. It was implemented as a Web-based system, allowing novice users to instantly benefit from the latest text-to-audio models in their creative processes.

4.1 Design

As illustrated in Figure 1, our interface requires users to first input a theme phrase for music audios to generate. The inputted phrase need not include precise musical descriptions since IteraTTA leverages a large language model to derive such descriptions suitable for text-to-audio models using knowledge embedded in the models [62]. Specifically, the interface queries a large language model that "Please give me four variational lists of comma-separated phrases describing what does a music clip of "*[theme phrase]*" sound." It then uses the four responded phrase lists as a variety of the first text prompts to start the music generation processes in parallel. This feature allows novice users to translate loosely-specified goals in their minds into musical descriptions, which can also help them to envisage variations of text prompts to explore.

IteraTTA then generates three music audios for each of the four prompts. The generated audios are arranged in two dimensions (see Figure 1), which enables novice users to understand how different music audios are generated by different text prompts, and also, how different music audios are generated by the same text prompts. This is intended to assist users in identifying which text prompts and audio priors are closely aligned with their goals and which direction is worth exploring. If a user identifies a suitable candidate text prompt, they can customize the prompt and generate new music audios with it. Alternatively, if the user discovers a suitable music audio, they can use it as an audio prior to generate new music audios. In essence, the user can explore the subspace of possible results that are proximate to their goals by constraining either text prompts or audio priors, while gradually refining their goals by themselves.

We have incorporated several features to facilitate the exploration of text prompts and audio priors, as shown in Figure 2. For instance, when a user specifies an audio prior, IteraTTA enables the user to compare generated results with it. It also offers an instant editing feature of text prompts, allowing users to amplify or suppress the sound of a selected instrument. This is achieved by simply adding a phrase of "with strong *[instrument]*" or "with no *[instrument]*" into a text prompt, but it provides an example of how they can modify generation results through prompts.

4.2 Implementation

As mentioned, we developed IteraTTA as a Web-based system to invite novice users for trying music generation with it. For the implementation of its back-end server, we utilized Python with FastAPI and incorporated an API of GPT-3.5[2] to construct initial prompts, while AudioLDM [11] was employed to generate the music audios. The length of music audios to generate was predetermined at 10 seconds so that our GPU server harnessing an NVIDIA RTX 2080 Ti can afford the generation of 12 audios (3 audios $\times$ 4 prompts) simultaneously. On average, the generation process takes approximately 15 seconds. In addition, we used DeepL API to translate text prompts into English when they were provided in non-English languages because we observed that it led to better results in Section 3. For the front-end interface of IteraTTA, we utilized Vue.js, which enables users to download the generated music audios or share them on Twitter.

[2] We used `gpt-3.5-turbo` of `https://platform.openai.com/docs/models/gpt-3-5`.

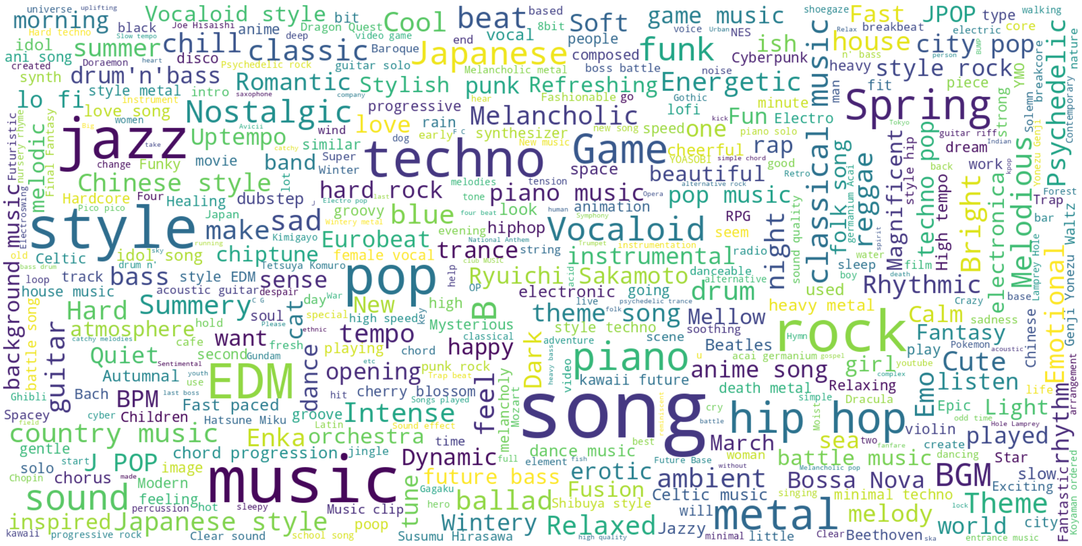


Figure 3. Word cloud of theme phrases the 8,831 users inputted on our Web service.

5. ANALYSIS

To investigate the effectiveness of IteraTTA in supporting diverse users in the wild, we deployed it as a publicly-available Web service in Japanese [3]. Within two days of release, 8,831 users generated 246,423 music audios. In this section, we discuss the insights we extracted from their usage logs and their responses to a form that we put a link to it on the Web service so that they can share their opinion and feedback voluntarily.

5.1 Diversity of theme phrases

We first examined the theme phrases that users inputted to initiate text-to-audio music generation processes and found that they were highly varied. Some users provided music-related phrases, such as “nice city pop” and “cute future bass,” while others were more specific, like: “80’s hip hop that break dancers would dance to.” There were also phrases expressing more abstract ideas, such as “Arabian caves” and “silent dream of a priestess.” Figure 3 visualizes the words often used in the translated phrases in the form of a word cloud, showing their diversity.

To explore the role of IteraTTA, we compared the theme phrases inputted by the users and the text prompts derived from them by the large language model to the text labels in the dataset used for training the text-to-audio model. Specifically, we randomly sampled 1,000 cases for each of the theme phrases, text prompts, and text labels [4] and calculated their representation vectors using the same pre-trained model of RoBERTa [44] as the text-to-audio model. We then visualized the distribution of the vectors using t-SNE [63], as presented in Figure 4. This indicates that IteraTTA guided the large language model to derive text prompts that bridged the gap between the diverse users’ theme phrases and the text labels in the training dataset. In fact, we found that the large language model successfully derived text prompts containing musical descriptions even from abstract phrases, such as “otherworldly harmonies, delicate strings, minimalistic percussion, dreamlike vocals” for “silent dream of a priestess.” These results suggest the effectiveness of guiding the construction of initial prompts to support the creative processes of novice users, as discussed in Section 3.1.

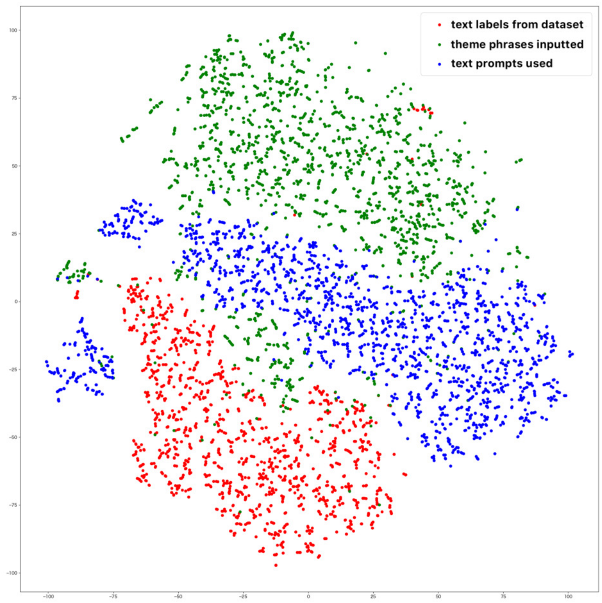


Figure 4. Visualization of the representation vectors of the theme phrases inputted by the users, the text prompts computationally derived from them, and the text labels in the training dataset.

5.2 Journey of iterative exploration

We also investigated how the users interacted with generated results produced by IteraTTA. We analyzed the in-

[3] Its English version is currently available at `http://iteratta.duckdns.org/`, and readers can try it on their Web browsers (Google Chrome is recommended).

[4] For the text labels, we extracted labels containing “music” from AudioCaps [13].

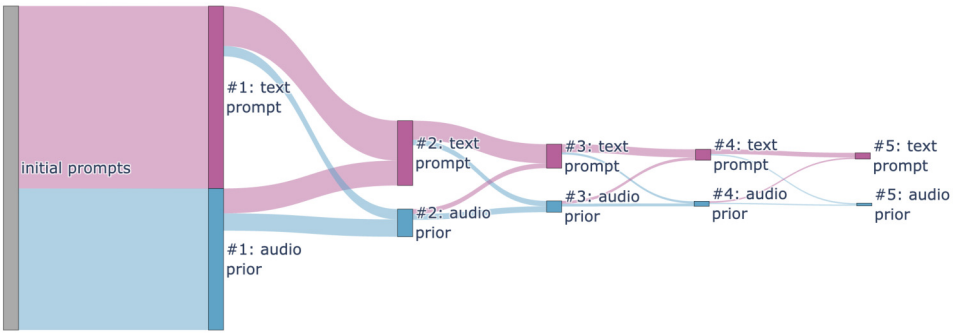


Figure 5. Visualization of how the users utilized the dual-sided exploration of IteraTTA.

teraction log of the service and obtained Figure 5. While some users just tried the exploration feature once, we found that others made iterative use of the feature, alternating between providing text prompts and audio priors. Interestingly, one user repeated this refinement process 32 times, specifying text prompts 14 times and audio priors 18 times before sharing their final result on Twitter. These points imply that our design, which enables dual-sided iterative exploration, helped the users effectively utilize the text-to-audio model.

5.3 Unleashing the creativity of novice users

We lastly analyzed the users' responses to the feedback form, which received 33 responses in total. Overall, most of them expressed their affirmative experiences with the text-to-audio music creation processes, like:

> *"It was a very interesting trial. I can interact with it throughout the day."*
>
> *"In my personal opinion, it can be used as a source of sampling materials and an idea generator. As a person who usually composes music, I never had any negative feelings about composing from text using this. It is wonderful."*

The latter comment suggests that the features of IteraTTA prepared for novice users can also benefit experienced users in different ways.

It is also notable that the users left comments implying the importance of the design requirements discussed in Section 3, such as how they enjoyed the open-ended exploration starting from loosely-specified theme phrases.

> *"It was fun to encounter songs that fit the theme I provided but I had never heard before."*
>
> *"I really enjoyed the points that I could take advantage of ChatGPT's ability to associate and verbalize even seemingly unconnected ideas, which allowed me to provide crazy theme phrases that would not be understood by a human. I also learned a lot about how to describe songs by looking at the derived text prompts."*

Interestingly, in the form, some users left a successful prompt that they reached after exploration:

> *"I would like to report that including a phrase of 'simple progression' or limiting the number of tracks yielded stabilized music audios, like: 'Ideal harmonious song: balanced instrumentation, band sound, simple chord progressions, rhythmic drum patterns, catchy pop melody, up to 12 tracks.' "*
>
> *"Adding 'clear sound quality' produces less noisy audios."*

It is surprising that, even though we provided no explicit description of the behavior of text-to-audio models, the users were able to gain such knowledge by themselves through the iterative exploration with IteraTTA. While such *prompt modifiers* (also known as *quality boosters*) [64] that influence results in a specific way have been discovered for text-to-image models in a community-driven manner [17, 64], the above comments would be the first examples for text-to-audio models, to the best of our knowledge. We assume that this is a manifestation of users' creativity in text-to-audio music generation processes and would be hard to derive without IteraTTA.

6. CONCLUSION

This paper introduces IteraTTA, an interface specifically designed for supporting novice users in their text-to-audio music generation processes. Its design is guided by two main principles, providing a) computational guidance for constructing initial prompts and b) dual-sided iterative exploration of text prompts and audio priors. The former can help novice users translate their loosely-specified goals into text prompts, which serve as starting points for exploration, even if they do not have rich artistic vocabularies. The latter is important for enabling them to comprehend the space of possible results and gradually refine their goals. To examine how diverse users utilize IteraTTA in their creative processes, we deployed it as a publicly-available Web service and analyzed users' behaviors, which highlight the importance of these design considerations in supporting the users' creativity. Importantly, these principles are applicable not only to the specific text-to-audio model but to other models, including those to be proposed in the near future. We believe that this paper can serve as a foundation for enabling novice users to benefit from state-of-the-art models in the MIR community.

7. ACKNOWLEDGEMENT

This work was supported in part by JSPS KAKENHI Grant Number JP21J20353, JST ACT-X Grant Number JPMJAX200R, and JST CREST Grant Number JPMJCR20D4, Japan.

8. REFERENCES

[1] G. Franceschelli and M. Musolesi, "Creativity and machine learning: A survey," *arXiv*, vol. abs/2104.02726, 2021.

[2] M. J. Muller, L. B. Chilton, A. Kantosalo, C. P. Martin, and G. Walsh, "Proceedings of the GenAICHI workshop: Generative AI and HCI," in *Extended Abstracts of the 2022 ACM SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2022, pp. 110:1–110:7.

[3] 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, p. 14, 2020.

[4] M. Rohrmeier, "On creativity, music's AI completeness, and four challenges for artificial musical creativity," *Transactions of the International Society for Music Information Retrieval*, vol. 5, no. 1, pp. 50–66, 2022.

[5] H. H. Tan and D. Herremans, "Music FaderNets: Controllable music generation based on high-level features via low-level feature modelling," in *Proceedings of the 21st International Society for Music Information Retrieval Conference*. ISMIR, 2020, pp. 109–116.

[6] K. Chen, C. Wang, T. Berg-Kirkpatrick, and S. Dubnov, "Music SketchNet: Controllable music generation via factorized representations of pitch and rhythm," in *Proceedings of the 21st International Society for Music Information Retrieval Conference*. ISMIR, 2020, pp. 77–84.

[7] Z. Wang, D. Wang, Y. Zhang, and G. Xia, "Learning interpretable representation for controllable polyphonic music generation," in *Proceedings of the 21st International Society for Music Information Retrieval Conference*. ISMIR, 2020, pp. 662–669.

[8] S. Dai, Z. Jin, C. Gomes, and R. B. Dannenberg, "Controllable deep melody generation via hierarchical music structure representation," in *Proceedings of the 22nd International Society for Music Information Retrieval Conference*. ISMIR, 2021.

[9] 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, pp. 722–729.

[10] 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*, vol. abs/2301.11325, 2023.

[11] H. Liu, Z. Chen, Y. Yuan, X. Mei, X. Liu, D. P. Mandic, W. Wang, and M. D. Plumbley, "AudioLDM: Text-to-audio generation with latent diffusion models," *arXiv*, vol. abs/2301.12503, 2023.

[12] 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 *Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2017, pp. 776–780.

[13] 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*. ACL, 2019, pp. 119–132.

[14] K. Drossos, S. Lipping, and T. Virtanen, "Clotho: an audio captioning dataset," in *Proceedings of the 2020 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2020, pp. 736–740.

[15] K. Swanwick, *Musical Knowledge: Intuition, analysis and music education*. London, UK: Routledge, 2002.

[16] J. E. Gromko, "Perceptual differences between expert and novice music listeners: A multidimensional scaling analysis," *Psychology of Music*, vol. 21, no. 1, pp. 34–47, 1993.

[17] J. Oppenlaender, R. Linder, and J. Silvennoinen, "Prompting AI art: An investigation into the creative skill of prompt engineering," *arXiv*, vol. abs/2303.13534, 2023.

[18] A. B. Arrieta, N. D. Rodríguez, J. D. Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera, "Explainable artificial intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI," *Information Fusion*, vol. 58, pp. 82–115, 2020.

[19] K. Andersen and P. Knees, "Conversations with expert users in music retrieval and research challenges for creative MIR," in *Proceedings of the 17th International Society for Music Information Retrieval Conference*. ISMIR, 2016, pp. 122–128.

[20] C. Roads, "Research in music and artificial intelligence," *ACM Computing Surveys*, vol. 17, no. 2, pp. 163–190, 1985.

[21] J. D. Fernández and F. J. Vico, "AI methods in algorithmic composition: A comprehensive survey," *Journal of Artificial Intelligence Research*, vol. 48, pp. 513–582, 2013.

[22] C. Liu and C. Ting, "Computational intelligence in music composition: A survey," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 1, no. 1, pp. 2–15, 2017.

[23] J. Briot and F. Pachet, "Deep learning for music generation: challenges and directions," *Neural Computing and Applications*, vol. 32, no. 4, pp. 981–993, 2020.

[24] E. Deruty, M. Grachten, S. Lattner, J. Nistal, and C. Aouameur, "On the development and practice of AI technology for contemporary popular music production," *Transactions of the International Society for Music Information Retrieval*, vol. 5, no. 1, p. 35, 2022.

[25] S. Ji, J. Luo, and X. Yang, "A comprehensive survey on deep music generation: Multi-level representations, algorithms, evaluations, and future directions," *arXiv*, vol. abs/2011.06801, 2020.

[26] L. Yang, S. Chou, and Y. Yang, "MidiNet: A convolutional generative adversarial network for symbolic-domain music generation," in *Proceedings of the 18th International Society for Music Information Retrieval Conference*. ISMIR, 2017, pp. 324–331.

[27] H. Dong and Y. Yang, "Convolutional generative adversarial networks with binary neurons for polyphonic music generation," in *Proceedings of the 19th International Society for Music Information Retrieval Conference*. ISMIR, 2018, pp. 190–196.

[28] 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*. OpenReview.net, 2019.

[29] Y. Huang and Y. Yang, "Pop Music Transformer: Beat-based modeling and generation of expressive pop piano compositions," in *Proceedings of the 28th ACM International Conference on Multimedia*. ACM, 2020, pp. 1180–1188.

[30] W. Hsiao, J. Liu, Y. Yeh, and Y. Yang, "Compound Word Transformer: Learning to compose full-song music over dynamic directed hypergraphs," in *Proceedings of the 35th AAAI Conference on Artificial Intelligence*. AAAI Press, 2021, pp. 178–186.

[31] G. Mittal, J. H. Engel, C. Hawthorne, and I. Simon, "Symbolic music generation with diffusion models," in *Proceedings of the 22nd International Society for Music Information Retrieval Conference*. ISMIR, 2021, pp. 468–475.

[32] B. Yu, P. Lu, R. Wang, W. Hu, X. Tan, W. Ye, S. Zhang, T. Qin, and T. Liu, "Museformer: Transformer with fine- and coarse-grained attention for music generation," in *Proceedings of the 36th Conference on Neural Information Processing Systems*. Curran Associates, Inc., 2022, pp. 1376–1388.

[33] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, and I. Sutskever, "Jukebox: A generative model for music," *arXiv*, vol. abs/2005.00341, 2020.

[34] A. Caillon and P. Esling, "RAVE: A variational autoencoder for fast and high-quality neural audio synthesis," *arXiv*, vol. abs/2111.05011, 2021.

[35] T. Hung, B. Chen, Y. Yeh, and Y. Yang, "A benchmarking initiative for audio-domain music generation using the FreeSound loop dataset," in *Proceedings of the 22nd International Society for Music Information Retrieval Conference*. ISMIR, 2021, pp. 310–317.

[36] 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, pp. 543–550.

[37] T. Akama, "Connective fusion: Learning transformational joining of sequences with application to melody creation," in *Proceedings of the 21th International Society for Music Information Retrieval Conference*. ISMIR, 2020, pp. 46–53.

[38] H. Hung, J. Ching, S. Doh, N. Kim, J. Nam, and Y. Yang, "EMOPIA: A multi-modal pop piano dataset for emotion recognition and emotion-based music generation," in *Proceedings of the 22nd International Society for Music Information Retrieval Conference*. ISMIR, 2021, pp. 318–325.

[39] P. Neves, J. Fornari, and J. B. Florindo, "Generating music with sentiment using Transformer-GANs," in *Proceedings of the 23rd International Society for Music Information Retrieval Conference*. ISMIR, 2022, pp. 717–725.

[40] K. Wang and J. V. Nickerson, "A literature review on individual creativity support systems," *Computers in Human Behavior*, vol. 74, pp. 139–151, 2017.

[41] C. A. Huang, H. V. Koops, E. Newton-Rex, M. Dinculescu, and C. Cai, "Human-AI co-creation in songwriting," in *Proceedings of the 21th International Society for Music Information Retrieval Conference*. ISMIR, 2020, pp. 708–716.

[42] J. A. Russell, "A circumplex model of affect," *Journal of Personality and Social Psychology*, vol. 39, no. 6, pp. 1161–1178, 1980.

[43] J. Heer, "Agency plus automation: Designing artificial intelligence into interactive systems," *Proceedings of the National Academy of Sciences*, vol. 116, no. 6, pp. 1844–1850, 2019.

[44] 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*, vol. abs/1907.11692, 2019.

[45] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, 2022, pp. 10 674–10 685.

[46] O. Gafni, A. Polyak, O. Ashual, S. Sheynin, D. Parikh, and Y. Taigman, "Make-a-scene: Scene-based text-to-image generation with human priors," in *Proceedings of the 17th European Conference on Computer Vision*. Springer, 2022, pp. 89–106.

[47] I. Simon, D. Morris, and S. Basu, "MySong: automatic accompaniment generation for vocal melodies," in *Proceedings of the 2008 ACM SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2008, pp. 725–734.

[48] C. A. Huang, C. Hawthorne, A. Roberts, M. Dinculescu, J. Wexler, L. Hong, and J. Howcroft, "The Bach doodle: Approachable music composition with machine learning at scale," in *Proceedings of the 20th International Society for Music Information Retrieval Conference*. ISMIR, 2019, pp. 793–800.

[49] 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 ACM SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2020, pp. 610:1–610:13.

[50] S. Rau, F. Heyen, S. Wagner, and M. Sedlmair, "Visualization for AI-assisted composing," in *Proceedings of the 23rd International Society for Music Information Retrieval Conference*. ISMIR, 2022, pp. 151–159.

[51] Y. Zhang, G. Xia, M. Levy, and S. Dixon, "COSMIC: A conversational interface for human-AI music co-creation," in *Proceedings of the 21th International Conference on New Interfaces for Musical Expression*. nime.org, 2021.

[52] Y. Zhou, Y. Koyama, M. Goto, and T. Igarashi, "Generative melody composition with human-in-the-loop bayesian optimization," in *Proceedings of the 2020 Joint Conference on AI Music Creativity*. DiVA.org, 2020.

[53] ——, "Interactive exploration-exploitation balancing for generative melody composition," in *Proceedings of the 26th International Conference on Intelligent User Interfaces*. ACM, 2021, pp. 43–47.

[54] P. C. Wright and A. F. Monk, "The use of think-aloud evaluation methods in design," *ACM SIGCHI Bulletin*, vol. 23, no. 1, pp. 55–57, 1991.

[55] O. Alhadreti and P. J. Mayhew, "Rethinking thinking aloud: A comparison of three think-aloud protocols," in *Proceedings of the 2018 ACM SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2018, p. 44.

[56] A. Strauss and J. Corbin, *Basics of Qualitative Research: Grounded Theory Procedures and Techniques*. Newbury Park, CA: Sage Publications, 1990.

[57] J. O. Talton, D. Gibson, L. Yang, P. Hanrahan, and V. Koltun, "Exploratory modeling with collaborative design spaces," *ACM Transactions on Graphics*, vol. 28, no. 5, pp. 1–10, 2009.

[58] C. Lynch, K. D. Ashley, N. Pinkwart, and V. Aleven, "Concepts, structures, and goals: Redefining ill-definedness," *International Journal of Artificial Intelligence in Education*, vol. 19, no. 3, pp. 253—266, 2009.

[59] H. Yakura, Y. Koyama, and M. Goto, "Tool- and domain-agnostic parameterization of style transfer effects leveraging pretrained perceptual metrics," in *Proceedings of the 30th International Joint Conference on Artificial Intelligence*. IJCAI, 2021, pp. 1208–1216.

[60] L. Tweedie, "Interactive visualisation artifacts: How can abstractions inform design?" in *Proceedings of the 10th BCS Conference on Human-Computer Interaction*. Cambridge University Press, 1995, pp. 247–265.

[61] M. A. Terry and E. D. Mynatt, "Side views: Persistent, on-demand previews for open-ended tasks," in *Proceedings of the 15th Annual ACM Symposium on User Interface Software and Technology*. ACM, 2002, pp. 71–80.

[62] J. Wei, Y. Tay, R. Bommasani, C. Raffel, B. Zoph, S. Borgeaud, D. Yogatama, M. Bosma, D. Zhou, D. Metzler, E. H. Chi, T. Hashimoto, O. Vinyals, P. Liang, J. Dean, and W. Fedus, "Emergent abilities of large language models," *Transactions on Machine Learning Research*, vol. *in press*, 2022.

[63] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, no. 86, pp. 2579–2605, 2008.

[64] J. Oppenlaender, "A taxonomy of prompt modifiers for text-to-image generation," *arXiv*, vol. abs/2204.13988, 2022.

SIMILARITY EVALUATION OF VIOLIN DIRECTIVITY PATTERNS FOR MUSICAL INSTRUMENT RETRIEVAL

Mirco Pezzoli **Raffaele Malvermi** **Fabio Antonacci** **Augusto Sarti**
Dipartimento di Elettronica, Informazione e Bioingegneria - Politecnico di Milano, Milan, Italy
mirco.pezzoli@polimi.it, raffaele.malvermi@polimi.it

ABSTRACT

The directivity of a musical instrument is a function that describes the spatial characteristics of its sound radiation. The majority of the available literature focuses on measuring directivity patterns, with analysis mainly limited to visual inspections. Recently, some similarity metrics for directivity patterns have been introduced, yet their application has not being fully addressed. In this work, we introduce the problem of musical instrument retrieval based on the directivity pattern features. We aim to exploit the available similarity metrics for directivity patterns in order to determine distances between instruments. We apply the methodology to a data set of violin directivities, including historical and modern high-quality instruments. Results show that the methodology facilitates the comparison of musical instruments and the navigation of databases of directivity patterns.

1. INTRODUCTION

The analysis of the directional sound radiation characteristics of musical instruments is a rather old topic in the literature with first works by Olson [1] and Meyer [2–4] dating back to the seventies. In the past few decades, numerous studies were proposed mainly focusing on accurate measurements of the directivity patterns [5–8] or on qualitative comparisons of the instrument characteristics [9–11].

Recently, the interest in spatial audio technologies [12] for virtual and augmented reality increased the attention towards the modeling and analysis of directivity patterns. In particular, the modeling of directional sound sources showed to provide improved sound field reconstruction for the navigation of sound scenes [13, 14]. Therefore, different solutions have been proposed to include the directivity of acoustic sources in simulation frameworks such as boundary and finite element methods [15], numerical simulation [16] and geometrical acoustics [17]. As a matter of fact, the directivity of sound sources impacts on the accuracy of room acoustics simulation [17] and it was shown to be relevant for auralization [18].

© M. Pezzoli, R. Malvermi, F. Antonacci and A. Sarti. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** M. Pezzoli, R. Malvermi, F. Antonacci and A. Sarti, "Similarity evaluation of violin directivity patterns for musical instrument retrieval ", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

In [19], the authors demonstrated that users are able to perceive differences between omnidirectional and directional sound sources, however the evaluation is limited to a single-tone dependent directivity pattern. In the work [20], it was shown that fluctuations occurring in the directivity patterns due to the movements of the musician influence the perception of listeners both in anechoic and reverberant conditions. More recently, in [21], the difference between frequency-dependent directivities and an *average* directivity pattern has been investigated proving the importance of modelling specific frequency-dependent directivities by means of listening tests.

Several studies [22–26] focus on the analysis of voice directivity patterns. In particular, [23, 25] analyze the patterns associated to held or isolated vowels and consonants from speech and singing voice [24]. Interestingly, the results on mouth and vocal tract configurations [26] showed their impact on the directivity pattern shape.

As far as the musical instruments are concerned, most of the works put the emphasis on accurate measurement procedures. Typically, the directional sound pressure is acquired in anechoic environments and under controlled conditions [6, 7]. Alternatively, near-field acoustic holography [27, 28] has been employed for the evaluation of the directional sound radiation using scanning microphone arrays [29]. More recently, a flexible procedure for measuring the directivity pattern of sound sources that works in low-reverberant environments was introduced in [8].

In [9], the directivity patterns of forty one orchestral instruments have been acquired and analyzed. The instruments were played by musicians, rather than mechanically excited, showing that the presence of the player body has the effect of smoothing the patterns. Nevertheless, although [9] draws an interesting analysis of the patterns, the evaluation is mainly limited to graphical inspection without a systematic comparison of the directivity patterns.

As a matter of fact, the quantitative and objective comparison of directivity patterns is still an open challenge. In the literature, some simple metrics have been proposed [30–33]. Although effective, the interpretation of the results and the quantification of the differences might not be easily interpreted. In general, most of the proposed metrics rely on the correlation between the directivity patterns, either in the spherical harmonics domain [30] or in the spatial domain [32]. In [30], the authors employed the normalized cross correlation (NCC) over the spherical harmonics coefficients of the directivity patterns. The anal-

ysis assessed similarities of partials at a given frequency generated by different played pitches. In [34], a rotation-invariant version of the NCC has been proposed to compare the directivity patterns of the data sets in [9,35], which have been made available through [36]. The devised metric has been used to find similarities across the partials of one instrument or between different instruments and a visualization of the corpus through MultiDimensional Scaling (MDS) [37] has been provided.

More recently, in [38], a novel set of metrics has been introduced, which includes the Jaccard similarity index (JSI) and the centers of mass distance (CMD) in addition to NCC. Both JSI and CMD are derived from the analysis of the so-called *principal radiation regions*, namely angular regions of the directivity pattern which exhibit the highest sound energy radiation. In [38], the metrics are used for the characterization of directivity patterns of prestigious historical violins enabling the quantitative comparison of the instruments. Nonetheless, the analysis is limited to a small set of 10 instruments and the conclusions drawn by the analysis of each metric, although relevant, are not readily combined.

In this work, we aim to exploit available similarity metrics for directivity patterns in a comprehensive and systematic fashion. Considering the problem of musical instrument retrieval based on the directivity pattern features [34, 39], we introduce novel distances, namely the Jaccard similarity distance (JSD) based on JSI, and the directivity index distance (DID) derived from the so-called *directivity index* (DI), which are combined with the CMD in a cumulative *Directivity Pattern Distance* (DPD). The proposed distances are blind with respect to the source type, because they work directly on the directivity values. Therefore, ideally they can be applied on any kind of sound sources, including musical instruments of different families. The joint adoption of multiple distances allows us to take into account different aspects of the directivity patterns without limiting the comparison to a single metric. Moreover, the introduced DPD provides a single-valued solution that represents the distance between the directivity patterns combining the information provided by each considered metric.

Although the proposed distances can be applied on different musical instruments, we tested them on a data set of violin directivities. As a matter of fact, violins represent an interesting case study due to the highly variability of directivity patterns among the instruments [40, 41]. The corpus contains a total of 18 instruments equally divided between 10 historical and 8 modern high-quality violins. To the best of our knowledge, this is the largest data set of violin directivity patterns evaluated in the current literature. The analysis allowed us to observe interesting similarities among the instruments, identifying relevant information in the data set. In particular, modern instruments are relatively distant from the historical ones. Moreover, thanks to the adoption of DPD, we could identify clusters of historical instruments made by one violin maker and two modern "twin" violin. Similarly to [34], we exploit the MDS technique for the visualization of the data set, which allows us to graphically assess the distances between the instruments observing the clusters of similar violins within. The obtained results pave the way to the retrieval of musical instruments according to their directional sound radiation and open novel perspectives for the exploitation of directivity pattern databases.

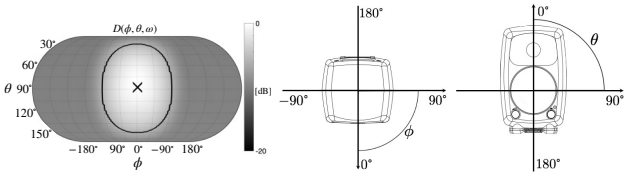


Figure 1. Example of directivity pattern $D(\phi, \theta, \omega)$ of a Genelec 8030A at $1.4\,\mathrm{kHz}$, taken from [43]. The principal radiation region $\mathcal{P}$ is delimited by a solid black line, while the center of mass $\mathbf{r}$ is marked by a black cross. The reference system is reported from top and frontal views.

2. SIMILARITY METRICS FOR DIRECTIVITY PATTERNS

Let us define the directivity pattern of an acoustic source as the square-integrable function $D(\cdot) \in \mathbb{L}^2(\mathbb{S}^2)$ describing the energy of the directional sound radiation. The directivity pattern is thus defined over a unit sphere comprising all the possible directions of emission. It follows that the directivity pattern can be conveniently expressed through the widely adopted spherical harmonics expansion [5, 8, 13] as

$$D(\phi, \theta, \omega) = \sum_{n=0}^{\mathrm{N}} \sum_{m=-n}^{n} C_n^m(\omega) Y_n^m(\phi, \theta), \qquad (1)$$

where $\phi \in [0, 2\pi]$ and $\theta \in [0, \pi]$ are the azimuth and inclination angles, respectively, $C_n^m(\omega)$ are the spherical harmonics coefficients associated with the source directivity pattern and $Y_n^m(\phi, \theta)$ is the spherical harmonic of degree n and order m [42]. It is worth noting that the directivity pattern (1) depends on the temporal frequency ω. Moreover, in (1), we assumed the directivity pattern to be band-limited being N the maximum expansion order. In Fig. 1, an example of a loudspeaker directivity pattern is reported.

2.1 Data model

2.1.1 Binary directivity pattern

In [38], the principal radiation region of a directivity pattern is defined as the set of adjacent directions $\mathcal{P}$ that correspond to the maximum acoustic energy emission. In particular, given a threshold value τ, the principal radiation region is defined as

$$\mathcal{P}(\omega) = \left\{ (\bar{\phi}_\mathrm{p}, \bar{\theta}_\mathrm{p}) : D_\mathrm{dB}(\bar{\phi}_\mathrm{p}, \bar{\theta}_\mathrm{p}, \omega) \geq \tau \right\}, \qquad (2)$$

where $\tau = -3\,\mathrm{dB}$ and

$$D_\mathrm{dB}(\phi, \theta, \omega) = 10 \log_{10} \left(\frac{D(\phi, \theta, \omega)}{\max(D(\phi, \theta, \omega))} \right) \qquad (3)$$

represents the normalized directivity pattern in decibel scale with $\max$ the function extracting its maximum value.

In Fig. 1, the principal radiation region $\mathcal{P}$ is delimited by a solid black line.

The thresholding procedure in (2) allows one to define the binary directivity pattern indicating the principal radiation region as

$$\bar{D}(\phi,\theta,\omega) = \begin{cases} 1 & (\phi,\theta) \in \mathcal{P}(\omega) \\ 0 & \text{otherwise} \end{cases} . \quad (4)$$

The adoption of the binary patterns is preferable rather than considering only the direction of the maximum, i.e. a single point, in the directivity pattern. As a matter of fact, the binary pattern indicates the regions of high energy emission, i.e. principal radiation, which can have arbitrary shape and extension accordingly to the overall directional characteristics of the directivity pattern.

2.1.2 *Centers of mass*

Although the binary pattern (4) provides a comprehensive representation of the principal radiation regions, it is convenient to further identify a "preferred" direction of emission for each region. Therefore, we define the center of mass for a principal radiation region $\mathcal{P}$ as [38]

$$\mathbf{r}(\omega) = \frac{1}{M} \sum_{p \in \mathcal{P}(\omega)} m_{\mathrm{p}} \boldsymbol{r}_{\mathrm{p}}, \quad (5)$$

where $\boldsymbol{r}_{\mathrm{p}} = \left[\sin\bar{\theta}_{\mathrm{p}}\cos\bar{\phi}_{\mathrm{p}}, \sin\bar{\theta}_{\mathrm{p}}\sin\bar{\phi}_{\mathrm{p}}, \cos\bar{\theta}_{\mathrm{p}}\right]^T$ are the points belonging to the set defined in (2). In practice, the directions of $\mathcal{P}$ are weighted using the corresponding energy value in the normalized pattern, namely

$$m_{\mathrm{p}} = \frac{D(\bar{\phi}_{\mathrm{p}}, \bar{\theta}_{\mathrm{p}}, \omega)}{\max(D(\cdot,\omega))}, \quad \text{with} \quad M = \sum_{\mathrm{p} \in \mathcal{P}(\omega)} m_{\mathrm{p}}. \quad (6)$$

The center of mass of the directivity pattern is marked in Fig. 1 by a black cross.

2.2 Distance Metrics

In order to compare the directivity patterns of acoustic sources within a data set, we rely on a set of metrics recently proposed in [38]. Differently from customarily directivity pattern comparisons, where a single metric is considered, the employment of multiple metrics allows us to take into account different characteristics that are captured by each metric.

2.2.1 *Jaccard Similarity Distance (*JSD*)*

According to [38], we define the Jaccard similarity index (JSI) between two binary directivity patterns as

$$\mathrm{JSI}_{k,j}(\omega) = \frac{|\bar{D}_k(\omega) \cap \bar{D}_j(\omega)|}{|\bar{D}_k(\omega) \cup \bar{D}_j(\omega)|}, \quad (7)$$

where $\cap$ is the intersection operator and $\cup$ is the union between the binary patterns of the kth and jth sources. From the definition in (7), it follows that $\mathrm{JSI}_{k,j}(\omega) = 1$ when two binary patterns match exactly, while $\mathrm{JSI}_{k,j}(\omega) = 0$ when the corresponding principal radiation regions do not overlap. In order to interpret the JSI in terms of a distance, we introduce the JSD metric as

$$\mathrm{JSD}_{k,j}(\omega) = 1 - \mathrm{JSI}_{k,j}(\omega), \quad (8)$$

so that the JSD decreases up to 0 when two principal radiation regions are matched and the maximum value of JSD is 1, indicating two completely disjoint regions.

2.2.2 *Center of Mass Distance (*CMD*)*

The CMD is defined in order to compute the distance between two centers of mass as [38]

$$\mathrm{CMD}_{k,j}(\omega) = \arctan\left(\frac{|\mathbf{r}_k(\omega) \times \mathbf{r}_j(\omega)|}{\mathbf{r}_k(\omega) \cdot \mathbf{r}_j(\omega)}\right), \quad (9)$$

where $\times$ and $\cdot$ denote the vectorial cross and dot products, respectively. As in [33], when multiple centers of mass are present inside the directivity patterns, the vectors $\mathbf{r}$ (5) are selected in order to retain the lowest $\mathrm{CMD}_{k,j}(\omega)$ values.

2.2.3 *Directivity Index Distance (*DID*)*

The directivity index (DI) is a well-known feature that describes the directionality of a sound source [33]. In particular, the DI measures how much energy is concentrated around the principal directions of a directivity pattern. In this work, we consider the DI of the normalized directivity patterns defined as

$$\mathrm{DI}_k(\omega) = \frac{1}{\int_0^{2\pi}\int_0^{\pi} \widehat{D}_k(\phi,\theta,\omega) d\phi d\theta}, \quad (10)$$

where $\widehat{D}_k$ is the normalized directivity pattern of the kth source in linear scale. The DI in (10) is computed with respect to the maximum value of the directivity pattern, which in case of normalized patterns is equal to 1. It follows that high DI values occur for directivity patterns with large principal radiation regions, and vice versa.

In order to compare two directivity patterns in terms of their DI values, we define the DID as

$$\mathrm{DID}_{k,j}(\omega) = \sqrt{(\mathrm{DI}_j(\omega) - \mathrm{DI}_k(\omega))^2}, \quad (11)$$

where DI_k and DI_j are the DI (10) of the kth and jth sources, respectively.

2.2.4 *Directivity Pattern Distance (*DPD*)*

In order to conveniently compare two sound sources in terms of their directivity features, we introduce an overall metric that combines the previously defined JSD, CMD and DID into a scalar value. Hence, we define the so-called directivity pattern distance DPD metric as

$$\mathrm{DPD}_{k,j} = \overline{\mathrm{JSD}}_{k,j} + \frac{\overline{\mathrm{CMD}}_{k,j}}{\max(\overline{\mathrm{CMD}}_{k,j})} + \frac{\overline{\mathrm{DID}}_{k,j}}{\max(\overline{\mathrm{DID}}_{k,j})}, \quad (12)$$

where $\overline{\mathrm{JSD}}_{k,j}, \overline{\mathrm{CMD}}_{k,j}, \overline{\mathrm{DID}}_{k,j}$ denote the mean of the three distance metrics over the frequency axis. It must be noted that the values of $\overline{\mathrm{CMD}}_{k,j}$ and $\overline{\mathrm{DID}}_{k,j}$ in (12) are normalized with respect to the maximum value encountered in the data set under analysis, such that all the components of the sum vary within the same dynamic range, i.e. between 0 and 1, and thus have the same relative importance in the definition of DPD.

3. EVALUATION

3.1 Data set of violin directivity patterns

The proposed methodology is applied to a data set of violin directivities. The data set includes the frequency-dependent directivity patterns of eighteen violins, including ten historical violins made between the 16th and 17th centuries and eight modern violins made during the last two centuries. For all the instruments, the owners provided consent for the usage of the results in an anonymous fashion. For this reason, and for the ease of reading, we will denote all the historical violins with labels H1–H10, while we will refer to the modern ones as M1–M8.

Concerning the collection of modern violins, it is noteworthy that the instruments labeled with M1–M6 are fine violins selected among the candidates of the "Antonio Stradivari International Triennial Competitions of Stringed instrument making". The competition, held in the city of Cremona since 1976, embraces both Cremonese and international competitors. Moreover, violins M7 and M8 were made by a Cremonese luthier and are known as "twin violins". The twin violins were built by employing the very same block of tonewood and following the same geometrical model. As a matter of fact, previous research already showed the high similarity in all the spatial characteristics of their sound.

The patterns were collected experimentally through the measurement procedure described in [8] and were evaluated at varying frequency within the range $[200, 5000]$ Hz using a 4th-order spherical harmonics expansion in (1). The instruments are played by one professional violinist who is free to move and play comfortably, while the source position and orientation are estimated by the system enabling the measurement of the directivity as described in [8]. The data processing pipeline and the computation of the metrics is developed using the MATLAB software.

3.2 Analysis of the metrics

To assess the significance of the proposed distance metrics, we first compare the frequency-averaged values of $\overline{\mathrm{JSD}}$, $\overline{\mathrm{CMD}}$ and $\overline{\mathrm{DID}}$ computed over the set of violin directivity patterns to those obtained for the same data with a commonly used similarity metric, namely NCC. The NCC metric provides a measure of the element-wise similarity between two patterns. In order to properly compare the previously defined distances with the baseline, the NCC between the patterns of the k-th and l-th violins is formalized in terms of a distance as

$$\overline{\mathrm{NCC}}_{k,l} = 1 - \frac{1}{S}\sum_{s=1}^{S} \frac{\widehat{D}_k(\phi,\theta,\omega_s)\widehat{D}_l(\phi,\theta,\omega_s)}{\|\widehat{D}_k(\phi,\theta,\omega_s)\|\|\widehat{D}_l(\phi,\theta,\omega_s)\|}, \quad (13)$$

where ω_s is the s-th frequency at which the directivity patterns are evaluated, with $s = 1, \ldots, S$ and S the total number of frequency bins in the data set. In this way, $\overline{\mathrm{NCC}}_{k,l}$ is close to zero when two patterns are similar and reaches a value equal to 2 when they are inversely correlated. Fig. 2 shows a comparison between all the metrics under study. For any possible pair of metrics, a 2D scatter plot is reported. The coordinates of the markers in the plot correspond to the two distances for all the possible pairs of violins in the data set. Z-score normalization is applied to the resulting values to ensure the same dynamic range along the axes [44]. The scatter plots in the first row show the comparison between $\overline{\mathrm{NCC}}$ (13) and each of the proposed metrics, while the scatter plots in the second row present the comparison between $\overline{\mathrm{JSD}}$, $\overline{\mathrm{CMD}}$ and $\overline{\mathrm{DID}}$ only. By inspecting the resulting distributions of points, it is possible to highlight correlations between the metrics.

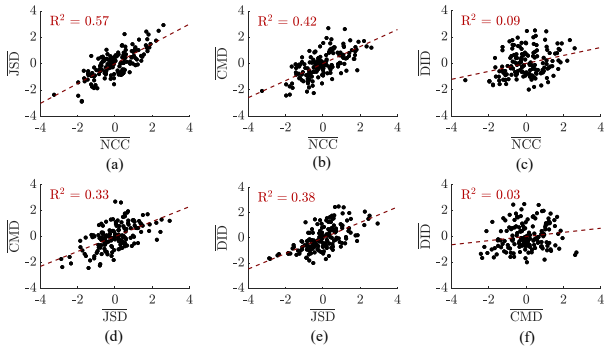


Figure 2. Comparison between proposed distance metrics ($\overline{\mathrm{JSD}}$, $\overline{\mathrm{CMD}}$, $\overline{\mathrm{DID}}$) and Normalized Cross Correlation ($\overline{\mathrm{NCC}}$). For each combination of metrics, a 2D scatter plot of the corresponding frequency-averaged values is shown. Z-score normalization is applied to ensure the same dynamic range along the axes [44]. Linear regression is performed to analyze the correlation between the metrics. The regressed line and the R^2 value, measuring the degree of correlation, are highlighted in red.

On the one hand, it can be noticed that some pairs of metrics exhibit a point distribution that concentrates along a line. A linear trend, in fact, can be observed in Fig. 2a and 2b, showing the ($\overline{\mathrm{NCC}}$, $\overline{\mathrm{JSD}}$) and ($\overline{\mathrm{NCC}}$, $\overline{\mathrm{CMD}}$) point distributions, respectively. Although less emphasized, a similar trend can be noticed in Fig. 2d and 2e, reporting the distribution of ($\overline{\mathrm{JSD}}$, $\overline{\mathrm{CMD}}$) and ($\overline{\mathrm{JSD}}$, $\overline{\mathrm{DID}}$), respectively.

The presence of linearity in these point distributions can be interpreted as due to correlation, i.e. shared information, between the metrics under analysis. This can be particularly true for $\overline{\mathrm{NCC}}$ and $\overline{\mathrm{JSD}}$, which both measure the degree of pattern matching by definition. More interestingly, however, correlation can be observed between $\overline{\mathrm{CMD}}$ and $\overline{\mathrm{NCC}}$. We can thus conclude that two violins with similar principal directions of radiation tend to exhibit highly matching directivity patterns. Furthermore, $\overline{\mathrm{JSD}}$ and $\overline{\mathrm{CMD}}$ can be used instead of $\overline{\mathrm{NCC}}$ to provide two similarity measures by looking at the pattern shape and at the principal direction of radiation separately without losing information. Indeed, Fig. 2d shows that $\overline{\mathrm{JSD}}$ and $\overline{\mathrm{CMD}}$ are less correlated than when considering $\overline{\mathrm{NCC}}$.

On the other hand, Fig. 2c and Fig. 2e do not exhibit a linear distribution. We can interpret this evidence as the absence of correlation between $\overline{\mathrm{DID}}$, $\overline{\mathrm{NCC}}$ and $\overline{\mathrm{CMD}}$. As a matter of fact, $\overline{\mathrm{DID}}$ measures the difference in the directivity index of two patterns, which is related to the energy distribution, and thus extracts an energy-related informa-

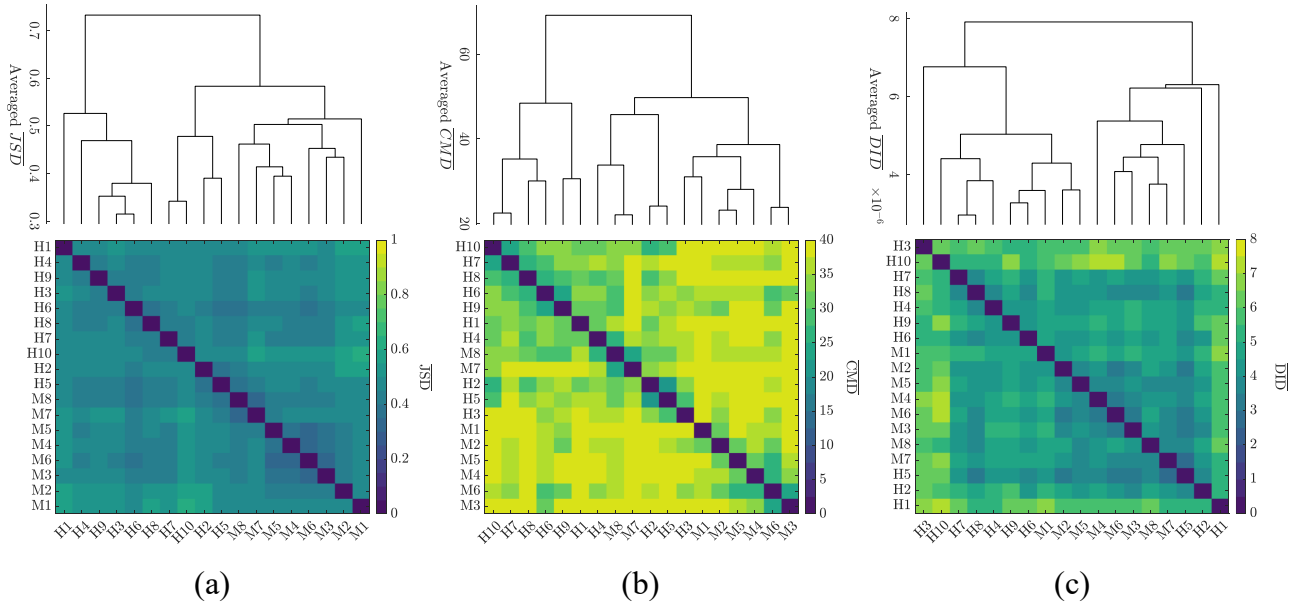


Figure 3. Evaluation of violin similarity based on $\overline{\text{JSD}}$ **(a)**, $\overline{\text{CMD}}$ **(b)** and $\overline{\text{DID}}$ **(c)**. The elements inside the matrices are obtained by averaging the frequency-dependent distance values. Pairs of similar violins are denoted with dark blue colors, while dissimilar violins are highlighted in yellow. Hierarchical clustering algorithms are employed to sort the elements inside the resulting matrices. The resulting dendrograms are reported above each distance matrix.

tion that is not captured by the other metrics.

A quantitative measure of the correlation between the metrics under analysis can be evaluated by performing linear regression on each point distribution. The regressed lines are denoted in red inside each 2D scatter plot. The regression accuracy is assessed in terms of the coefficient of determination R^2, which is related to the Pearson correlation coefficient in the case of simple linear regression. The resulting R^2 values are reported as an inset inside each plot. According to [45], the values range between 0 and 1, and moderate and strong correlation occurs for values greater than 0.3 and 0.6, respectively.

It can be noticed that the pair $(\overline{\text{NCC}}, \overline{\text{JSD}})$ shows moderate to strong correlation, with $\mathrm{R}^2 = 0.57$. Moderate correlations can be observed for $(\overline{\text{NCC}}, \overline{\text{CMD}})$, $(\overline{\text{JSD}}, \overline{\text{CMD}})$ and $(\overline{\text{JSD}}, \overline{\text{DID}})$, with $\mathrm{R}^2 = 0.42$, $\mathrm{R}^2 = 0.33$ and $\mathrm{R}^2 = 0.38$, respectively. Finally, no correlation occurs for $(\overline{\text{NCC}}, \overline{\text{DID}})$ and $(\overline{\text{CMD}}, \overline{\text{DID}})$, with $\mathrm{R}^2 = 0.09$ and $\mathrm{R}^2 = 0.03$, respectively.

3.3 Violins clustering based on similarity metrics

In order to group musical instruments that exhibit a sound emission with similar spatial characteristics, the proposed distance metrics can be used together with classical clustering methods. In this case, hierarchical clustering methods are employed, based on the generation of dendrograms [46]. In particular, the proposed similarity metrics are used for the iterative definition of the dendrogram. It is worth to underline, that the adopted clustering algorithm does not require any training data and it is applied directly on the computed similarities. Fig. 3 shows the distance matrices assessing the pairwise similarity between all the violins in the data set under study. The matrix elements in Fig. 3a, 3b and 3c are obtained using the frequency-averaged $\overline{\text{JSD}}$, $\overline{\text{CMD}}$ and $\overline{\text{DID}}$ values, respectively. Pairs of similar violins are highlighted with dark blue colors, while dissimilar violins are colored in yellow. The elements of each distance matrix are sorted according to the leaf order of a dendrogram tree. The Ward's method [47] is used to generate the tree branches, such that similar violins concentrate inside the matrix.

By inspecting the resulting distance matrices, it is noteworthy that the order of the elements in the matrix varies depending on the specific distance considered. However, expected groups of violins can be highlighted. In Fig. 3a, the subsets of historical and modern violins are clearly distinguished, being placed at the top-left and bottom-right corners of the $\overline{\text{JSD}}$ matrix, respectively. In particular, the twin violins (M7-M8) exhibit the minimum $\overline{\text{JSD}}$ value in the matrix and the remaining modern violins (M1-M6) cluster together. The same behavior occurs also in Fig. 3b and 3c, although at different locations inside the matrices.

Regarding the historical violins, H1 appears to be very different with respect to the rest of the data set. In particular, high values are encountered for $\overline{\text{JSD}}$ and $\overline{\text{DID}}$, which are related to the pattern shape and energy, respectively. Conversely, the same violin is more similar to other historical violins concerning the principal directions of radiation.

Fig. 4 shows the results of violin clustering based on the proposed overall metric DPD. On the left, the dendrogram computed with the Ward's method is shown, while on the right the resulting distance matrix is reported, with the elements sorted following the dendrogram hierarchy. Pairs of violins characterized by DPD values close to either zero or the maximum are colored in green or white, repsectively.

Typically, clusters can be extracted from the hierarchy of the dendrogram tree by applying a thresholding with re-

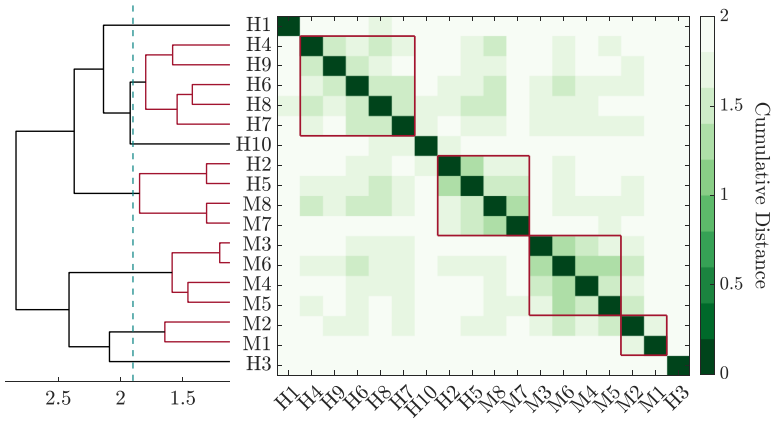


Figure 4. Violin clustering based on the proposed DPD metric. Small distance values correspond to pairs of similar violins and are highlighted in green, while pairs of dissimilar violins exhibit high distance values and are highlighted in white. The elements inside the matrix are sorted according to the dendrogram tree, shown on the left. Clustering is performed by thresholding the dendrogram tree. The threshold is denoted with a cyan line, while the resulting clusters are colored in red.

spect to the tree height. We decide to subdivide the dendrogram at a height equal to 1.9, i.e. the mean value between the height of the lowest branch in the tree and the height of its root, denoted with a vertical dashed cyan line. As a result, seven clusters are identified inside the data set: (i) three consisting of a single violin (i.e. H1, H10 and H3), (ii) one cluster made of five historical violins (i.e. H4-H6-H7-H8-H9), (iii) one cluster made of two historical violins and the twin violins (i.e. H2-H5-M7-M8), (iv) one cluster with four modern violins (i.e. M3-M4-M5-M6) and (v) one cluster with two modern violins (i.e. M1-M2).

The obtained clusters are coherent with the similarities extracted from the single proposed metrics. In particular, the distinction between historical and modern violins and the high similarity between the twin violins (M7-M8) are emphasized by the DPD. Moreover, hierarchical clustering based on DPD is able to recognize a cluster with five historical violins i.e. H4-H9-H6-H8-H7. Remarkably, the instruments belonging to this cluster have been made by the same luthier.

3.3.1 Visualization of the data collection through MDS

Given the similarity analysis of the violin directivity patterns based on the proposed DPD metric, the employment of MultiDimensional Scaling methods (MDS) allows one to easily visualize and navigate the collection of data [37]. In practice, MDS methods enable the mapping of the violins into a multidimensional space so that the similarities between the musical instruments in the data set are preserved.

Fig. 5 shows a 3D representation of the data set based on MDS. In this case, the coordinate system results from the use of Nonclassical MDS with the distance matrix shown in Fig. 4 as input. Each marker in the scatter plot corresponds to a violin, and the same marker color is used to denote violins belonging to the same cluster.

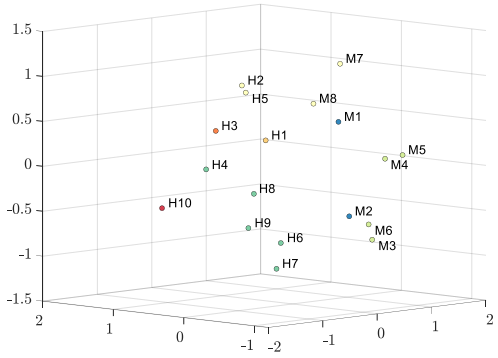


Figure 5. 3D representation of the violin data set based on Multidimensional Scaling. Nonclassical MDS is applied on the resulting DPD matrix to map the violins into a three-dimensional space. Each marker in the scatter plot corresponds to a violin. The same marker color is used for violins belonging to the same cluster.

4. CONCLUSION

In this paper, we tackle the problem of directivity patterns comparison by introducing a novel distance metric denoted as DPD, which is based on a combination of different similarity metrics and features of the patterns. This approach allows one to compactly compare the similarity of directivity patterns exploiting the different information provided by JSD, CMD and DID. The considered metrics are compared within each other and with respect to the well-known NCC, highlighting that they provide mutually uncorrelated information.

We analyzed a data set of directivity patterns of 18 violins divided between 10 historical and 8 modern instruments. Through the use of DPD, we were able to identify clusters of similar instruments among which a set of historical instruments made by the same maker and two "twin" violins. Finally, the MDS technique enabled the visualization of the violin data collection starting from the computed distances.

We foresee the application of the proposed approach for the retrieval of musical instruments based on directivity pattern characteristics. This opens new perspectives for the navigation of data sets of directivity patterns which can be used to provide a more realistic acoustic presence of musical instruments within spatial audio applications.

5. REFERENCES

[1] H. F. Olson, *Music, physics and engineering*. Courier Corporation, 1967, vol. 1769.

[2] J. Meyer, "Directivity of the bowed stringed instruments and its effect on orchestral sound in concert halls," *The Journal of the Acoustical Society of America*, vol. 51, no. 6B, pp. 1994–2009, 1972.

[3] ——, *Acoustics and the performance of music: Manual for acousticians, audio engineers, musicians, architects and musical instrument makers*. Springer Science & Business Media, 2009.

[4] ——, "The influence of the directivity of musical instruments on the efficiency of reflecting or absorbing areas in proximity to the orchestra," *Acta Acustica united with Acustica*, vol. 36, no. 3, pp. 147–161, 1976.

[5] G. Weinreich and E. B. Arnold, "Method for measuring acoustic radiation fields," *The Journal of the Acoustical Society of America*, vol. 68, no. 2, pp. 404–411, 1980.

[6] J. Pätynen, V. Pulkki, and T. Lokki, "Anechoic recording system for symphony orchestra," *Acta Acustica united with Acustica*, vol. 94, no. 6, pp. 856–865, 2008.

[7] J. Curtin, "Measuring violin sound radiation using an impact hammer," *J. Violin Soc. Am. VSA Papers, XXII*, no. 1, pp. 186–209, 2009.

[8] A. Canclini, F. Antonacci, S. Tubaro, and A. Sarti, "A methodology for the robust estimation of the radiation pattern of acoustic sources," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 211–224, 2020.

[9] N. R. Shabtai, G. Behler, M. Vorländer, and S. Weinzierl, "Generation and analysis of an acoustic radiation pattern database for forty-one musical instruments," *The Journal of the Acoustical Society of America*, vol. 141, no. 2, pp. 1246–1256, 2017.

[10] F. Otondo and J. H. Rindel, "The influence of the directivity of musical instruments in a room," *Acta acustica united with Acustica*, vol. 90, no. 6, pp. 1178–1184, 2004.

[11] J. Pätynen and T. Lokki, "Directivities of symphony orchestra instruments," *Acta Acustica united with Acustica*, vol. 96, no. 1, pp. 138–167, 2010.

[12] J. G. Tylka and E. Y. Choueiri, "Fundamentals of a parametric method for virtual navigation within an array of ambisonics microphones," *Journal of the Audio Engineering Society*, vol. 68, no. 3, pp. 120–137, 2020.

[13] M. Pezzoli, F. Borra, F. Antonacci, A. Sarti, and S. Tubaro, "Reconstruction of the virtual microphone signal based on the distributed ray space transform," in *26th European Signal Processing Conference (EUSIPCO)*. IEEE, 2018, pp. 1537–1541.

[14] M. Pezzoli, F. Borra, F. Antonacci, S. Tubaro, and A. Sarti, "A parametric approach to virtual miking for sources of arbitrary directivity," *IEEE Trans. on audio, speech, and language Process.*, vol. 28, pp. 2333–2348, 2020.

[15] R. Mehra, L. Antani, S. Kim, and D. Manocha, "Source and listener directivity for interactive wave-based sound propagation," *IEEE transactions on visualization and computer graphics*, vol. 20, no. 4, pp. 495–503, 2014.

[16] J. Ahrens and S. Bilbao, "Computation of spherical harmonic representations of source directivity based on the finite-distance signature," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 83–92, 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9257177/

[17] J. Klein and M. Vorländer, "Simulative investigation of required spatial source resolution in directional room impulse response measurements," in *EAA Spatial Audio Signal Processing Symposium*, 2019, pp. 37–42.

[18] B. N. Postma, H. Demontis, and B. F. Katz, "Subjective evaluation of dynamic voice directivity for auralizations," *Acta Acustica united with Acustica*, vol. 103, no. 2, pp. 181–184, 2017.

[19] L. M. Wang and M. C. Vigeant, "Evaluations of output from room acoustic computer modeling and auralization due to different sound source directionalities," *Applied Acoustics*, vol. 69, no. 12, pp. 1281–1293, 2008.

[20] D. Ackermann, C. Böhm, F. Brinkmann, and S. Weinzierl, "The acoustical effect of musicians' movements during musical performances," *Acta Acustica united with Acustica*, vol. 105, no. 2, pp. 356–367, 2019.

[21] A. Corcuera Marruffo and V. Chatziioannou, "A pilot study on tone-dependent directivity patterns of musical instruments," in *Audio Engineering Society Conference: AES 2022 International Audio for Virtual and Augmented Reality Conference*. Audio Engineering Society, 2022.

[22] C. Noufi, D. Markovic, and P. Dodds, "Reconstructing the dynamic directivity of unconstrained speech," *arXiv preprint arXiv:2209.04473*, 2022.

[23] B. F. Katz, F. Prezat, and C. d'Alessandro, "Human voice phoneme directivity pattern measurements," in *4th Joint Meeting of the Acoustical Society of America and the Acoustical Society of Japan*, 2006, p. 3359.

[24] B. Katz and C. d'Alessandro, "Directivity measurements of the singing voice," in *International Congress on Acoustics (ICA 2007)*, 2007, p. 6p.

[25] P. Kocon and B. B. Monson, "Horizontal directivity patterns differ between vowels extracted from running speech," *The Journal of the Acoustical Society of America*, vol. 144, no. 1, pp. EL7–EL12, 2018.

[26] S. Bellows and T. Leishman, "High-resolution analysis of the directivity factor and directivity index functions of human speech," in *Audio Engineering Society Convention 146*. Audio Engineering Society, 2019.

[27] J. D. Maynard, E. G. Williams, and Y. Lee, "Nearfield acoustic holography: I. theory of generalized holography and the development of NAH," *The Journal of the Acoustical Society of America*, vol. 78, no. 4, pp. 1395–1413, 1985.

[28] M. Olivieri, M. Pezzoli, F. Antonacci, and A. Sarti, "A physics-informed neural network approach for nearfield acoustic holography," *Sensors*, vol. 21, no. 23, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/23/7834

[29] D. Fernandez Comesana, T. Takeuchi, S. Morales Cervera, and K. R. Holland, "Measuring musical instruments directivity patterns with scanning techniques," in *25th International Congress on Sound and Vibration, ICSV 2019*, 2019.

[30] F. Hohl and F. Zotter, "Similarity of musical instrument radiation-patterns in pitch and partial," *Fortschritte der Akustik, DAGA, Berlin*, 2010.

[31] P. Guillon, R. Nicol, and L. Simon, "Head-related transfer functions reconstruction from sparse measurements considering a priori knowledge from database analysis: A pattern recognition approach," in *Audio Engineering Society Convention 125*. Audio Engineering Society, 2008.

[32] S. Moreau, J. Daniel, and S. Bertet, "3d sound field recording with higher order ambisonics–objective measurements and validation of a 4th order spherical microphone," in *120th Convention of the AES*, 2006, pp. 20–23.

[33] C. Molloy, "Calculation of the directivity index for various types of radiators," *The Journal of the Acoustical Society of America*, vol. 20, no. 4, pp. 387–405, 1948.

[34] T. Carpentier and A. Einbond, "Spherical correlation as a similarity measure for 3d radiation patterns of musical instruments," in *16ème Congrès Français d'Acoustique*. HAL Open Science, 2022.

[35] S. Weinzierl, M. Vorländer, G. Behler, F. Brinkmann, H. von Coler, E. Detzner, J. Krämer, A. Lindau, M. Pollow, F. Schulz *et al.*, "A database of anechoic microphone array measurements of musical instruments," 2017.

[36] J. Ahrens, "Database of spherical harmonic representations of sound source directivities," https://doi.org/10.5281/zenodo.3707708, Mar 2020.

[37] N. Saeed, H. Nam, M. I. U. Haq, and D. B. Muhammad Saqib, "A survey on multidimensional scaling," *ACM Computing Surveys (CSUR)*, vol. 51, no. 3, pp. 1–25, 2018.

[38] M. Pezzoli, A. Canclini, F. Antonacci, and A. Sarti, "A comparative analysis of the directional sound radiation of historical violins," *The Journal of the Acoustical Society of America*, vol. 152, no. 1, pp. 354–367, 2022.

[39] M. Olivieri, R. Malvermi, M. Pezzoli, M. Zanoni, S. Gonzalez, F. Antonacci, and A. Sarti, "Audio information retrieval and musical acoustics," *IEEE Instrum. Meas. Mag.*, vol. 24, no. 7, pp. 10–20, 2021.

[40] G. Weinreich, "Directional tone color," *The Journal of the Acoustical Society of America*, vol. 101, no. 4, pp. 2338–2346, 1997.

[41] J. Woodhouse, "The acoustics of the violin: a review," *Reports on Progress in Physics*, vol. 77, no. 11, p. 115901, 2014.

[42] A. Schmitz, T. Karolski, and L. Kobbelt, "Using spherical harmonics for modeling antenna patterns," in *2012 IEEE Radio and Wireless Symposium*. IEEE, 2012, pp. 155–158.

[43] J. G. Tylka, R. Sridhar, and E. Choueiri, "A database of loudspeaker polar radiation measurements," in *Audio Engineering Society Convention 139*. Audio Engineering Society, 2015.

[44] A. Jain, K. Nandakumar, and A. Ross, "Score normalization in multimodal biometric system," *Pattern Recognition*, vol. 38, pp. 2270–2285, 12 2005.

[45] J. Cohen, *Statistical power analysis for the behavioral sciences*. Routledge, 2013.

[46] G. Gruvaeus and H. Wainer, "Two additions to hierarchical cluster analysis," *British Journal of Mathematical and Statistical Psychology*, vol. 25, no. 2, pp. 200–206, 1972.

[47] S. Sharma, N. Batra *et al.*, "Comparative study of single linkage, complete linkage, and ward method of agglomerative clustering," in *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*. IEEE, 2019, pp. 568–573.

POLYRHYTHMIC MODELLING OF NON-ISOCHRONOUS AND MICROTIMING PATTERNS

George Sioros
University of Plymouth
georgios.sioros@plymouth.ac.uk

ABSTRACT

Computational models and analyses of musical rhythms are predominantly based on the subdivision of durations down to a common isochronous pulse, which plays a fundamental structural role in the organization of their durational patterns. Meter, the most widespread example of such a temporal scheme, consists of several hierarchically organized pulses. Deviations from isochrony found in musical patterns are considered to form an expressive, micro level of organization that is distinct from the structural macro-organization of the basic pulse. However, polyrhythmic structures, such as those found in music from West Africa or the African diaspora, challenge both the hierarchical subdivision of durations and the structural isochrony of the above models. Here we present a model that integrates the macro- and micro-organization of rhythms by generating non-isochronous grids from isochronous pulses within a polyrhythmic structure. Observed micro-timing patterns may then be generated from structural non-isochronous grids, rather than being understood as expressive deviations from isochrony. We examine the basic mathematical properties of the model and show that meter can be generated as a special case. Finally, we demonstrate the model in the analysis of micro-timing patterns observed in Brazilian samba performances.

1. INTRODUCTION

Isochrony has been a fundamental element of computational and cognitive models of musical rhythm, which are often inspired by the organization of rhythms found in Western classical music theory [1]. The advantages of isochrony as a structural foundation for the organization of music are numerous, from the potential to explain our ability to synchronize to music [2–5], to defining higher level qualities such as tempo [1, 6]. However, while periodicity is common in music, strict isochrony is almost never observed [7] and many studies document with empirical data the systematic durational patterns from music around the world, which do not fit into an isochronous structure, including the Viennese waltz [8], Brazilian Samba [9–11], Mali Jembe music [12] and the Norwegian Telespringar [13].

Generally, such patterns are understood on the basis of an underlying structural basic isochronous pulse and expressive deviations from it [14]. Although an isochronous pulse may not be directly observed and measured in the music signal, a steady beat may be inferred by the listener, most evidently when we bob our head or tap our feet to the music. Numerous beat tracking algorithms exist in the music information retrieval literature [15] and even cognitive models have been formalized that attempt to imitate this behavior [16, 17]. In most music-theoretical and cognitive models of rhythm, the beats of the basic isochronous pulse are grouped together, say every two or three, resulting in slower pulses that coincide with all faster ones, forming a hierarchical structure often referred to as meter [1, 18].

Typically, deviations from isochrony are modeled by processing repeating rhythmic patterns to derive statistical properties, such as the mean deviation from an isochronous pulse at each location of the repetition cycle (see for example [10]). Such statistical models only describe the processed recordings and cannot be generalized. They do, however, provide evidence that deviations from isochrony may be more than a mere expressive element of the performance, and are rather structural components of music [12].

Polyrhythmic music challenges the principle of hierarchical organization of rhythms. Polyrhythms are organized on the basis of multiple isochronous pulses of different periods that do not coincide [19–21]. Polyrhythmic elements are found in music around the world, with most representative examples coming from West Africa and the African diaspora [22, 23]. Even in groove-oriented music, where a strong sense of pulse is felt, short patterns that suggest an alternative pulse are common [24] and evidence indicate that they are central to experiencing groove [25]. Perhaps unsurprisingly, certain polyrhythmic music traditions also exhibit large and systematic deviations from isochrony [26]. It has been proposed that the deviations from isochrony and the polyrhythmic character of music from the African diaspora are related, one enhancing the other [9, p. 234, 23].

This paper presents a music-theoretical model that constructs non-isochronous grids from different isochronous pulses within a polyrhythmic context. Essentially, it attributes systematic non-isochronous patterns to underlying polyrhythmic structures that may be considered fundamental to the organization of the rhythmic patterns. In section 2, we present relevant music-theoretical concepts. Section 3, introduces a mathematical formalization of our model along with its key properties. In section 4, we employ the model to analyze Brazilian samba performance data taken from existing literature.

© G. Sioros. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** G. Sioros, "Polyrhythmic modelling of non-isochronous and microtiming patterns" in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.*, Milan, Italy, 2023.

2. BACKGROUND

2.1 Structural isochrony

In music theory, meter is formalized as a hierarchical structure that groups pulses periodically [18]. In typical mainstream Western music, the events are aligned across the voices in such a way that the salient moments coincide to form a metrical hierarchy (Figure 1). The different levels of the hierarchy correspond to various regularities found in the rhythm. While each level takes the form of a steady pulse, the various levels are stacked so that slower pulses coincide with all faster ones. Consequently, the periodicities in the articulated patterns in the music should also align in a similar fashion, although each voice may not necessarily articulate a specific metrical level. The periodic grouping of the beats has been formalized as a prime factorization problem [27–29]. Generative models of meter that can be implemented in computer algorithms have been developed as a set of transformations [30] and an abstract context-free grammar [31]

As a cognitive mechanism, meter is understood as oscillations that represent the attentional energy of the listener [3, 4, 32] or a predictive schema [33] that expresses our expectations about the timing of musical events [34]. The limitations of our cognition impose temporal limits to the pulses of the metrical hierarchy [1, p. 29, 35]. At the lower limit, the shortest pulse has a period of ~100ms, and at the upper limit, the longest pulse has a period of ~1.5s. The highest metrical salience is observed for pulses with a moderate period of ~700ms [6]. Tempo is then defined as the frequency of the most salient isochronous pulse of the metrical hierarchy.

The durations of the sound events are classified by the listener into discrete categories [2, p. 382, 36–39] that are influenced by the sensation of a pulse or meter evoked in them, so that a certain duration may be interpreted as a different category when listened in a different metrical framework [40]. A rhythmic pattern is essentially coded as a series of nominal durations that are a multiple of the basic isochronous pulse of the metrical hierarchy.

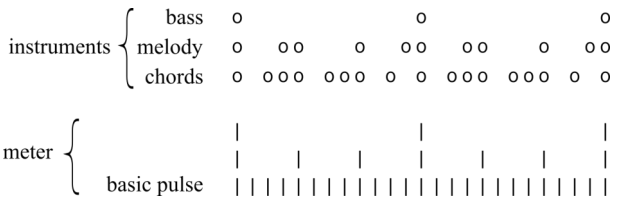


Figure 1: The metrical structure (bottom) emerging from The first two bars of "Conquest of Paradise" of composer Vangelis (top). Events are marked with (o). The three pulse levels of the metrical hierarchy (|) have periodicities of simple integer ratios 1:3:4 and are aligned with no phase differences.

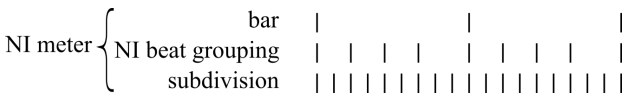


Figure 2: Example of a Non-Isochronous meter. The middle metrical level is non-isochronous and can be constructed as the Euclidean rhythm $E(4,9)$.

Despite the fundamental role that isochrony plays in the construction of meter, non-isochronous grouping of beats create non-isochronous metrical levels and meters (NI meters) [1, Ch. 7] (see Figure 2 for an example). Such groupings are based on the principle of maximal evenness [1, 41, 42] which is also the basis for the Euclidean rhythms that are encountered in many traditional rhythms [43, 44]. Such non-isochronous patterns are constructed by distributing a number of onsets k as evenly as possible over a number of beats n of an isochronous pulse. A Euclidean rhythm can then be denoted as $E(k,n)$ [43].

2.2 Micro-timing

While the metrical hierarchy determines durational categories for musical events, continuously variable timing determines an expressive level of organization of rhythms [14]. Systematic deviations from the nominal durations are typically measured in ms or as a percentage of the nominal beat duration to allow for an easier comparison between music segments with different tempi (Figure 3). The phenomenology of micro-timing has been discussed within the context of various music genres (see [9, 12, 45, 46] for some examples). Micro-timing modeling typically relies on statistical analysis of the timing of musical events over the duration of a performance to identify systematic, non-isochronous patterns [10, 47].

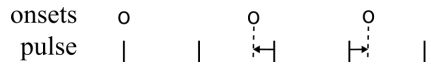


Figure 3: Onsets (o) may not exactly align with the isochronous pulse and have expressive (micro) timing deviations denoted with arrows (→).

2.3 Polyrhythms

Polyrhythms have been described in music theory as a form of metric dissonance [19, 20]. While consonant pulses align to give rise to the hierarchical structure of meter, dissonant pulses intertwine. Typically, polyrhythms consist of pulses with distinct beat durations that are not simple integer multiples of one another. The ratio between the beat durations of the two pulses determines the length of the repetition cycle of the entire polyrhythm. The number of beats of each pulse within a cycle of the polyrhythm n_1 and n_2 are related to the beat durations of the two pulses ΔT_1 and ΔT_2:

$$\frac{\Delta T_1}{\Delta T_2} = \frac{n_2}{n_1} \tag{1}$$

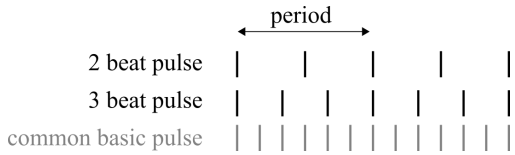


Figure 4: The 2|3 polyrhythm. A basic isochronous pulse subdivides both the 2-beat and the 3-beat pulses.

Polyrhythms can then be represented as $n_1|n_2$. Figure 4 depicts a polyrhythm in which 2 beats of one pulse have the same duration as 3 beats of a second pulse. The pulses that constitute a polyrhythmic structure may have a common faster subdivision with a beat duration longer than the perceptual threshold of 100ms, resulting in a type of grouping dissonance or polymeter [21].

2.4 Polyrhythms as flexible spaces

In principle, onsets are assumed to belong to one of the pulses of a polyrhythm, even if they are not perfectly aligned. However, it has been proposed that in music from the African diaspora, the intervals between the pulses of a 16|12 polyrhythm define a flexible space [23]. Events occurring between a beat from the 12-beat pulse and a beat from the 16-beat pulse may have a 'mixed' character, belonging at the same time to both pulses. Here, we extend this concept of 'mixed' character events to create non-isochronous grids that combine the character of both pulses of a polyrhythm. In this way, we formalize (micro) timing deviations from isochrony as a structural element of musical rhythms rooted in polyrhythms.

3. NON-ISOCHRONOUS GRIDS

3.1 Definition and construction

Non-isochronous pulses are constructed from two isochronous pulses of different periods. We will refer to the constructed non-isochronous pulses as grids (NI grids) to better distinguish them from isochronous pulses. The NI grid is formed by gradually changing the positions of the beats of one isochronous pulse towards a proximate beat of the other pulse. We call the first pulse the 'formative' pulse and the later pulse the 'target' pulse of the NI grid (Figure 5). The new beat positions for the NI grid are determined by:

$$N_i = F_i + S_i \cdot (T_j - F_i) \tag{2}$$

where i is the beat index of the formative pulse, F_i is the formative's beat time, T_j is the target's proximate beat time and S_i is the 'shift' as a fraction of the distance between the two beats taking values in the range [0, 1].

In principle, each beat may be shifted independently to form an NI pattern with multiple durations. Here, we examine the case of a uniform shift, where all formative beats are shifted by the same parameter S towards the nearest beat of the target pulse. Then, as S goes from 0 to 1, the formative pulse is being 'morphed' into the target pulse. While the relative shift S is uniform, the individual beat

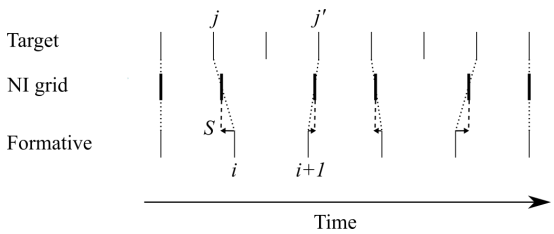


Figure 5: Construction of a non-isochronous grid by uniformly shifting the beats of a formative pulse towards the nearest beat positions of a target pulse.

displacements in time units (e.g. ms) will still have different values as the distance between the beats of the two pulses is not the same for all beats. Furthermore, S can also exhibit dynamic variations. In this sense, it should be understood as analogous to tempo, which can serve as a uniform parameter within a given time span and can also exhibit variations over the duration of a piece.

NI grids constructed by a uniform shift consist of only two beat classes, which we refer to as Short and Long for simplicity. If ΔF is the period of the formative pulse and ΔT the period of the target pulse, then the two beat classes' durations are:

$$\Delta N = (1 - S) \cdot \Delta F + S \cdot \Delta T \cdot k \tag{3}$$

where, k can take one of two values:

$$k_{Short} = \left\lfloor \frac{\Delta F}{\Delta T} \right\rfloor = q, k_{Long} = \left\lceil \frac{\Delta F}{\Delta T} \right\rceil = q + 1 \tag{4}$$

where $\lfloor \quad \rfloor$ and $\lceil \quad \rceil$ denote the floor and ceiling functions.

The durations of the Short and Long beats are limited within a NI grid. For $S = 0$, both the Short and Long beats have a duration equal to the period of the formative pulse ΔF, which is the upper limit for the Short beats and the lower limit for the Long beats. Conversely, for $S = 1$, the Short beats reach their shortest duration and the Long beats their longest duration which are integer multiples of the duration of the period of the target pulse ΔT. When $\Delta F < \Delta T$, i.e. $n_F > n_T$, then $k_{Short} = 0$ and $k_{Long} = 1$ and therefore the Short beats can reach a duration of 0.

By construction, the total number of beats of an NI grid is the same as the number of beats of the formative pulse. The number of Long beats of an NI grid can be calculated as the remainder of the division between the number of beats of the formative and target pulses:

$$n_{Long} = n_T \,\%\, n_F \tag{5}$$

3.2 Maximal evenness in NI grids

One of the key properties of the NI grids is that the Short and Long beats are evenly distributed; a direct consequence of the underlying polyrhythmic pulses being isochronous and the shift S being uniform. Different alignments of the pulses result in different rotations of the Short-Long beat pattern.

So far, we have examined shifts of the formative beats towards the nearest target beats. The above equations and the even distribution of the Short-Long beat classes also apply to shifts towards the next or previous target beat. Arbitrary combinations of the shifts, e.g. some towards the nearest beat and others towards the next beat, may also result in an even distribution of the two beat classes. However, this is guaranteed only when all formative beats follow the same rule. In Figure 6, the example of a 6 beat formative pulse and an 8 beat target pulse is shown. It follows from Eqn (5) that the number of Long beats in the resulting NI grid is 2, and the remaining 4 are Short. The different alignments of the formative and target pulses in Figure 6 produce the same Short-Long beat pattern but in different rotations.

Given a polyrhythm $n_F|n_T$ and its total duration, Eqns (3 - 5) can be used to calculate the durations and number of Long and Short beats in the corresponding NI grid as a

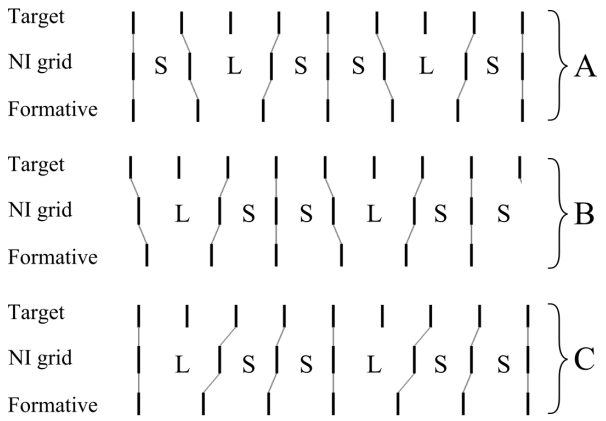


Figure 6: Three different alignments of a formative pulse with 6 beats and a target pulse of 8 beats produce the same pattern of evenly distributed Short-Long (L/S) beats, but in a different rotation. In A and B, all formative beats are shifted towards the nearest target beats. In C, the formative beats are all shifted forward, i.e. always towards the next target beat.

function of the shift S. The NI grid can then be produced directly by the Euclidean algorithm as a maximally even distribution of the two beat classes [1, 41, 43].

NI grids can be represented as polyrhythms with the additional parameter S: $NI(n_F|n_T, S)$. However, in contrast to polyrhythms, the order that the two pulses appear in the NI grid definition is important. The formative and target pulses are not equivalent as can easily be seen from Eqn (5), where the modulo operation is not commutative, and the fact that the number of beats of an NI grid is equal to the number of beats of the formative pulse, but not of the target pulse.

As a consequence of the even distribution of the Long and Short beats, every Euclidean pattern can be constructed as a NI grid (see Figure 7 for an example taken from [43]). In fact, it follows directly from the construction of Euclidean patterns that Euclidean patterns are equivalent to NI grids with a shift $S = 1$:

$$E(l, m) = NI(l \mid m, 1) \tag{6}$$

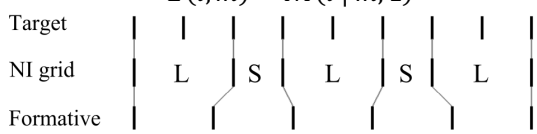


Figure 7: The Cuban cinquillo pattern is the Euclidean rhythm $E(5,8)$ and the NI grid $NI(5 \mid 8, 1)$.

Since the levels of a metrical hierarchy are evenly distributed, they can be constructed from NI grids. For NI meters, the process is similar to the construction of Euclidean patterns. However, typically, meters include isochronous pulses at all levels of their hierarchy and, therefore, these meters can be constructed from degenerate NI grids for which the Short and Long beats have the same duration.

4. APPLICATION TO SAMBA PERFORMANCES

Music events may be assigned to the beats of an NI grid. Similar to beat tracking, analyzing an observed durational pattern requires aligning event onsets with beats. As not all beats in an NI grid are necessarily articulated, there may be more than one NI grid that fits the rhythmic pattern. Information about the polyrhythmic structures that may be expected for the music at hand can help reduce the search space and find meaningful solutions. Here, we demonstrate the potential of the model in musical rhythm analysis by constructing NI grids that fit the durational patterns found in Brazilian samba.

Various rhythmic patterns of samba recordings have been reported in the literature [9–11, 13]. They are considered a characteristic feature of the performances and an integral part of the style, although the degree of deviation from isochrony as well as the specific non-isochronous patterns measured vary between studies. In [10], the same Samba rhythm was recorded at three different tempi. From the recordings, mean durations of the four events that make up the basic repetitive pattern were calculated. The results, which are summarized in Table 1, show that all three tempi follow the same general Medium-Short-Medium-Long durational pattern. The relative durations however are different at each tempo, with the fast and preferred tempi showing the most similarity and the slow tempo being more distinct and closer to isochrony with a characteristic lengthened last event. Additionally, the two Medium events (1 and 3) were reported to be significantly different between them, indicative of Medium-Short and Medium-Long duration [10].

Here we hypothesize that the observed non-isochronous patterns of Table 1 are the result of an underlying polyrhythmic structure and we attempt to reproduce them as: 1) NI grids with a 4-beat formative pulse and 3-beat target pulse (section 4.1), and 2) NI grids with a 5-beat formative pulse and a binary target pulse (section 4.2). In section 4.3, we use the NI grids to propose potential explanations for the differences in the observed durations between the three tempi.

4.1 The 4-beat formative pulse hypothesis

Our first hypothesis is that the basic Samba pattern (Table 1) emerges from an underlying 4|3 polyrhythm. From Eqn (5), it follows that the corresponding NI grid consists of 3

Tempo - BPM	Event number			
	1	2	3	4
Measured in ms				
Fast - 133	121 ±7.1	69 ±6.0	112 ±5.1	153 ±8.6
Preferred - 100	157 ±8.7	110 ±8.4	142 ±5.8	196 ±9.0
Slow - 69	212 ±9.7	198 ±12.4	206 ±6.6	256 ±9.5
Measured in percent of the total duration				
Fast - 133	27	15	25	34
Preferred - 100	26	18	24	33
Slow - 69	24	23	24	29

Table 1: Mean durations and standard deviations of the four events of the basic Samba pattern from [8, Tbl. 3].

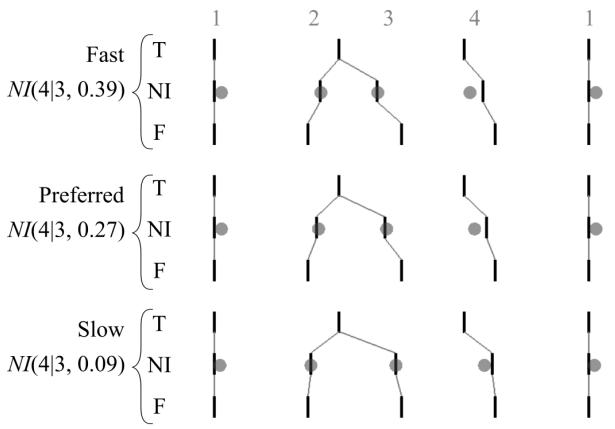


Figure 8: Potential NI grids with a 4-beat formative pulse for the Samba patterns of Table 1. A hypothetical alignment between the target (T) and formative (F) pulses is shown. The NI grids are specified on the left. The four events from Table 1 and the corresponding mean durations are indicated by grey circles and the integer numbers at the top.

Tempo NI grid	Event number			
	1	2	3	4
Fast *NI*(4|3, 0.39)	128.7 (**7.7**)	69 (0.0)	128.7 (**16.7**)	128.7 (**24.33**)
Preferred *NI*(4|3, 0.27)	165.0 (8.0)	110.0 (0.0)	165.0 (**23.0**)	165.0 (**31.0**)
Slow *NI*(4|3, 0.09)	224.7 (**12.7**)	198.0 (0.0)	224.7 (**18.7**)	224.7 (**31.3**)

Table 2: Theoretical durations in ms for the four samba events based on the NI grids of Figure 8. In parenthesis, the differences between the predicted durations and the observed mean durations for the respective events are shown for comparison with the standard deviations reported in Table 1. Difference greater than the respective standard deviations are shown in bold.

Long beats and a single Short beat. Since the NI grid has the same number of beats as the number of events in the Samba pattern, all four beats coincide with an event (Figure 8) and the shortest event duration (event 2) is aligned to the Short beat of the NI grid. Then, *S* is chosen so that it minimizes the difference between the theoretical beat durations and the mean event durations for the three different tempi. The results are summarized in Table 2.

The three different Samba patterns correspond to three different shifts *S*, so that events 1 and 2 of the Samba pattern are well aligned to the NI grid at all three tempi. However, since the NI grids consist of only two beat classes, events 1, 3 and 4 are matched to beats with the same duration and therefore this model cannot capture the characteristic longer 4th event and the difference in the durations of the two Medium events (1 and 3).

4.2 The 5-beat formative pulse hypothesis

Our second hypothesis is that the observed Medium-Short-Medium-Long pattern stems from the superposition of 5-beat and a binary pulse. Since only 4 of the 5 beats of the NI grid coincide with events, the two beat classes may produce three distinct event durations and in this way reproduce more accurately the event durations in the pattern (Figure 9, top). As in the previous hypothesis, the Short beat is aligned with event 2. The longer fourth event in this hypothesis spans two beats.

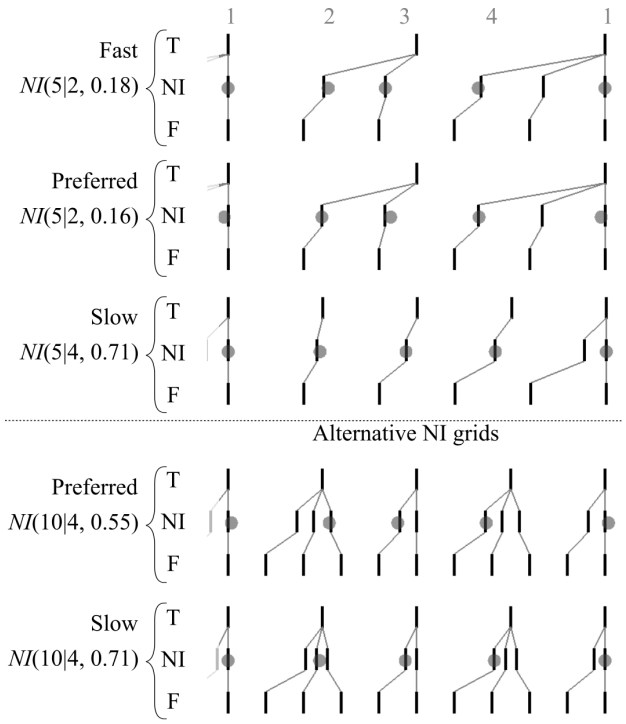


Figure 9: Potential NI grids with a 5-beat formative pulse (specified on the left) for the patterns in Table 1 (indicated here with grey circles) for the three different tempi. Two alternative NI grids are shown for the preferred and slow tempo duration patterns.

In the fast and preferred tempi, the length of event 4 is roughly double of the 2nd event and therefore we hypothesize that it spans two Short beats. Consequently, NI grid consists of 3 Short beats and 2 Long beats and therefore it is derived from a 5|2 polyrhythm. In the slow tempo, the difference between the duration of event 4 and the rest of the events is smaller. Our hypothesis is that this longer event spans two beats but not of equal durations. Since the first 3 events have similar durations, we hypothesize that a NI grid based on a 5|4 polyrhythm can reproduce this pattern, with the 3 Long beats aligned to the first 3 events. Finally, as previously, *S* is chosen to minimize the difference between the theoretical and observed mean durations. The results are summarized in Table 3.

The 5-beat formative pulse hypothesis reproduces more accurately the observed Samba pattern. The differences between the theoretical and observed durations are below the respective standard deviations, except for event 2 and 3 at the preferred tempo. This is indicative of the inability of the 5|2 model to capture the subtle difference between the two Medium events.

Introducing a subdivision to the Formative and Target pulses can address the above shortcoming of the model. A

Tempo NI grid	Event number			
	1	2	3	4
Fast *NI*(5|2, 0.18)	115.7 (5.3)	74.5 (5.5)	115.7 (3.7)	149.0 (3.9)
Preferred *NI*(5|2, 0.16)	150.8 (6.2)	101.2 (**8.8**)	150.8 (**8.8**)	202.3 (6.3)
Slow *NI*(5|4, 0.71)	205.3 (6.7)	205.3 (7.3)	205.3 (0.7)	256.0 (0.0)
Alternative NI grids				
Preferred *NI*(10|4, 0.55)	164.9 (7.9)	110.3 (0.3)	137.6 (4.4)	192.2 (3.8)
Slow *NI*(10|4, 0.71)	205.3 (6.7)	205.3 (7.3)	205.3 (0.7)	256.0 (0.0)

Table 3: Theoretical durations in ms for the four samba events based on the NI grids of Figure 9. In parenthesis, the differences between the theoretical durations and the observed mean durations for the respective events are shown. Differences greater than the respective standard deviations are shown in bold.

10|4 NI grid can reproduce the differences between the durations of event 1 and 3 (Figure 9 and Table 3, alternative NI grids). The period of the 10-beat Formative pulse at the this tempo is 61ms, which is below the perceptual threshold mentioned in section 2.1 for metrical subdivisions. Nevertheless, it may still be a plausible hypothesis considering the shortest event duration in the pattern at hand is 69ms. A similar hypothesis for the fast tempo would result in a period of 46ms for the Formative pulse, which was considered too fast within this context and was ommited. At the slow tempo, the alignment of the 10-beat Formative pulse is identical to the 5-beat one and thus offers no advantage.

4.3 Tempo dependence

The above hypotheses provide alternative explanations to those given in [10] for the tempo dependence of the basic samba pattern.

In the 4-beat formative hypothesis, we reproduce the patterns by gradually changing the shift *S*, from an almost purely binary pattern at slow tempo to a mixed character pattern at faster tempi. As the tempo becomes faster, the 4-beat pulse approaches the lower threshold for a metrical subdivision and events are pulled from their formative positions (period of 114ms) towards the ternary subdivision (period of 151ms), which is still significantly above the threshold.

In the 5-beat formative hypothesis, the tempo dependence is explained by the introduction of a subdivision in the target pulse at the slow tempo and change to the polyrhythmic structure from 5|2 to 5|4. At the preferred tempo, the small value of *S* indicates that events are mainly attracted to the faster 5-beat pulse (period of 121ms) and to a lesser extend to the slower binary subdivision (period of 303ms). As the tempo increases, S moves towards the binary pulse, possibly due to the 5-beat pulse crossing the 100ms threshold (91ms period). At the slow tempo, the binary pulse is subdivided, resulting in pulses with moderate periods (174ms for the 5-beat pulse and 218ms for the 4-beat pulse) and a more mixed character pattern.

5. CONCLUSION

In this paper, we formalize a novel model for non-isochronous and micro-timing rhythmic patterns that departs from theoretical and cognitive models that emphasize the hierarchical grouping of isochronous pulses prevalent in Western classical music theories. Instead, our model is rooted in non-hierarchical polyrhythmic relationships, such as those found in African polyphony. By incorporating polyrhythmic structures consisting of two pulses, our approach integrates both the structural/macro level and the expressive/micro level of musical rhythms, which are traditionally treated as separate. The resulting construct of non-isochronous (NI) grids unifies these levels within a novel framework. Non-isochronous groupings such as the Euclidean rhythms are then a special case of the model. While, NI grids can model systematic timing deviations from isochrony, not all deviations from isochrony can be accounted for by NI grids, and expressive timing may introduce micro-timing deviations to NI grids.

Our model is a music-theoretical one and is not intended to represent cognitive processes directly. However, some of its predictions may be relevant to music cognition. For example, it has previously been argued that only two beat classes, Long and Short, are perceptually relevant and that Medium duration events must be understood as expressive variants of these two beat classes [48]. A subsequent study showed that this is indeed the case in Mali Drum Ensemble music [49]. Our model makes similar predictions about the existence of only two beat classes, albeit for different reasons and with different implications for the observed patterns. To assess the perceptual relevance of these predictions, further analysis of musical performances and behavioral experiments are needed.

The potential of non-isochronous grids in music analysis was demonstrated in the example of Brazilian Samba. We developed two concrete hypotheses for the basic Samba pattern reported in the literature [10], which model the most salient features of the measured event durations and offer alternative explanations and interpretations for their tempo dependence. Polyrhythmic interpretations of the non-isochronous patterns observed in Samba performances have been hypothesized before, for example in [9]. However, such hypotheses are typically explored in a phenomenological and abstract, conceptual form. Our model provides the basis for a formalized method of determining the details of the polyrhythmic and micro-timing character of the observed durational patterns.

A future study will further test our preliminary hypotheses and compare them with other accounts of samba patterns. For example, we will investigate the possibility that some of the variation in measured durations may be due to a dynamic shift *S* that changes from one repetition of the pattern to the next. In addition, we will explore the development and evaluation of automated methods for discovering NI grids that can account for the durational patterns observed in music from various genres and music traditions.

6. REFERENCES

[1] J. London, *Hearing in Time: Psychological Aspects of Musical Meter*. New York: Oxford University Press, 2012. doi: 10.1093/acprof:oso/9780199744374.001.0001.

[2] H. Honing, 'Structure and Interpretation of Rhythm in Music', in *The Psychology of Music*, D. Deutsch, Ed., 3rd ed.Academic Press, Elsevier, 2012, pp. 367–404.

[3] E. W. Large and M. R. Jones, 'The dynamics of attending: How people track time-varying events.', *Psychological Review*, vol. 106, no. 1, pp. 119–159, 1999, doi: 10.1037//0033-295X.106.1.119.

[4] M. R. Jones, 'Musical time', in *The oxford handbook of music psychology*, S. Hallam, I. Cross, and M. Thaut, Eds., New York: Oxford University Press, 2008, pp. 81–92. doi: 10.1093/oxfordhb/9780199298457.013.0008.

[5] E. W. Large *et al.*, 'Dynamic models for musical rhythm perception and coordination', *Front. Comput. Neurosci.*, vol. 17, p. 1151895, May 2023, doi: 10.3389/fncom.2023.1151895.

[6] R. Parncutt, 'A perceptual model of pulse salience and metrical accent in musical rhythms', *Music Perception*, vol. 11, no. 4, pp. 409–464, 1994.

[7] N. Jacoby and J. H. McDermott, 'Integer Ratio Priors on Musical Rhythm Revealed Cross-culturally by Iterated Reproduction', *Current Biology*, vol. 27, no. 3, pp. 359–370, 2017, doi: 10.1016/j.cub.2016.12.031.

[8] A. Gabrielsson, 'Interplay between Analysis and Synthesis in Studies of Music Performance and Music Experience', *Music Perception*, vol. 3, no. 1, pp. 59–86, Oct. 1985, doi: 10.2307/40285322.

[9] L. Naveda, F. Gouyon, C. Guedes, and M. Leman, 'Microtiming Patterns and Interactions with Musical Properties in Samba Music', *Journal of New Music Research*, vol. 40, no. 3, pp. 225–238, Sep. 2011, doi: 10.1080/09298215.2011.603833.

[10] M. R. Haugen and A. Danielsen, 'Effect of tempo on relative note durations in a performed samba groove', *Journal of New Music Research*, vol. 49, no. 4, pp. 349–361, Aug. 2020, doi: 10.1080/09298215.2020.1767655.

[11] G. Guillot, 'Multi-level Anisochrony in Afro-Brazilian music', *GMTH Proceedings*, pp. 406–421, 2022, doi: 10.31751/p.200.

[12] R. Polak, 'Rhythmic Feel as Meter : Non-Isochronous Beat Subdivision in Jembe Music from Mali', *Music Theory Online*, vol. 16, no. 4, pp. 1–26, 2010.

[13] M. R. Haugen, 'Investigating Musical Meter as Shape: Two Case Studies of Brazilian Samba and Norwegian Telespringar', in *Proceedings of the 25th Anniversary Conference of the European Society for the Cognitive Sciences of Music*, E. Van Dyck, Ed., Ghent, Belgium, 2017, pp. 67–74.

[14] E. F. Clarke, 'Levels of structure in the organization of musical time', *Contemporary Music Review*, vol. 2, no. 1, pp. 211–238, Jan. 1987, doi: 10.1080/07494468708567059.

[15] M. E. P. Davies and S. Bock, 'Evaluating the Evaluation Measures for Beat Tracking', presented at the International Society for Music Information Retrieval Conference, 2014.

[16] M. J. Velasco and E. W. Large, 'Pulse Detection in Syncopated Rhythms Using Neural Oscillators', in *Proceedings of the 12th International Society for Music Information Retrieval Conference*, A. Klapuri and C. Leider, Eds., Miami, Florida, USA: University of Miami, 2011.

[17] E. W. Large, J. A. Herrera, and M. J. Velasco, 'Neural Networks for Beat Perception in Musical Rhythm', *Front. Syst. Neurosci.*, vol. 9, no. November, Nov. 2015, doi: 10.3389/fnsys.2015.00159.

[18] F. Lerdahl and R. Jackendoff, *A Generative Theory of Tonal Music*. Cambridge, Massachusetts: The MIT Press, 1983.

[19] M. Yeston, *The Stratification of Musical Rhythm*. New Haven, CT: Yale University Press, 1976.

[20] H. Krebs, *Metrical dissonance in the music of Robert Shumann*. New York, Oxford: Oxford University Press, 1999.

[21] V. K. Agawu, *Representing African music: postcolonial notes, queries, positions*. New York: Routledge, 2003.

[22] S. Arom, *African polyphony and polyrhythm*. Cambridge University Press, 1991. doi: 10.1017/CBO9780511518317.

[23] C. D. Stover, 'A theory of flexible rhythmic spaces for diasporic African music', PhD, University of Washington, 2009. doi: /10.7560/LAMR37202.

[24] A. Danielsen, *Presence and Pleasure: The Funk Grooves of James Brown and Parliament*. Middletown, CT: Wesleyan University Press, 2006.

[25] 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, Jun. 2022, doi: 10.1525/mp.2022.39.5.503.

[26] R. Polak, J. London, and N. Jacoby, 'Both isochronous and non-isochronous metrical subdivision afford precise and stable ensemble entrainment: A corpus study of malian jembe drumming', *Frontiers in Neuroscience*, vol. 10, no. JUN, pp. 1–11, 2016, doi: 10.3389/fnins.2016.00285.

[27] G. Sioros and C. Guedes, 'Syncopation as Transformation', in *Sound, Music and Motion*, M. Aramaki, O. Derrien, R. Kronland-Martinet, and S. Ystad, Eds., in Lecture Notes in Computer Science, vol. 8905. Springer International Publishing, 2014, pp. 635–658. doi: 10.1007/978-3-319-12976-1.

[28] C. Barlow and H. Lohner, 'Two essays on theory', *Computer Music Journal*, vol. 11, no. 1, pp. 44–60, 1987.

[29] C. Barlow, 'Corrections for Clarence Barlow' s Article: Two Essays on Theory', *Computer Music Journal*, vol. 11, no. 4, p. 10, 1987.

[30] G. Sioros, M. E. P. Davies, and C. Guedes, 'A generative model for the characterization of musical rhythms', *Journal of New Music Research*, vol. 47, no. 2, pp. 114–128, Mar. 2018, doi: 10.1080/09298215.2017.1409769.

[31] M. Rohrmeier, 'Towards a formalization of musical rhythm', in *Proceedings of the 21st International Society for Music Information Retrieval Conference*, Montréal, Canada, 2020, pp. 621–629.

[32] E. W. Large, 'Neurodynamics of Music', in *Music Perception*, 2010, pp. 201–231. doi: 10.1007/978-1-4419-6114-3.

[33] S. Koelsch, P. Vuust, and K. Friston, 'Predictive Processes and the Peculiar Case of Music', *Trends in Cognitive Sciences*, vol. 23, no. 1, pp. 63–77, 2019, doi: 10.1016/j.tics.2018.10.006.

[34] D. Huron, *Sweet anticipation: music and the psychology of expectation*. Cambridge, Massachusetts / London, England: The MIT Press, 2006.

[35] B. H. Repp, 'Rate Limits of Sensorimotor Synchronization', *Advances in Cognitive Psychology*, vol. 2, no. 2, pp. 163–181, Jan. 2006, doi: 10.2478/v10053-008-0053-9.

[36] E. F. Clarke, 'Categorical Rhythm Perception: an Ecological Perspective', in *Action and Perception in Rhythm and Music*, A. Gabrielsson, Ed., Stockholm: Royal Swedish Academy of Music, 1987, pp. 19–33.

[37] P. Desain and H. Honing, 'The Quantization of Musical Time: A Connectionist Approach', *Computer Music Journal*, vol. 13, no. 3, p. 56, Jan. 1989, doi: 10.2307/3680012.

[38] P. Desain and H. Honing, 'The formation of rhythmic categories and metric priming', *Perception*, vol. 32, no. 3, pp. 341–365, 2003, doi: 10.1068/p3370.

[39] M. L. A. Jongsma, P. Desain, and H. Honing, 'Rhythmic context influences the auditory evoked potentials of musicians and nonmusicians', *Biological Psychology*, vol. 66, no. 2, pp. 129–152, Apr. 2004, doi: 10.1016/j.biopsycho.2003.10.002.

[40] E. W. Large, 'Rhythm Categorization in Context', in *Large, Edward W. 'Rhythm categorization in context.' Proceedings of the International Conference on Music Perception and Cognition*, Keele, UK, 2000.

[41] J. Clough and J. Douthett, 'Maximally Even Sets', *Journal of Music Theory*, vol. 35, no. 1/2, pp. 93–173, 1991, doi: 10.2307/843811.

[42] R. Cohn, 'A Platonic Model of Funky Rhythms', *Journal of the Society for Music Theory*, vol. 22, no. 2, 2016, doi: 10.30535/mto.22.2.1.

[43] G. T. Toussaint, 'The Euclidean algorithm generates traditional musical rhythms', in *Proceedings of BRIDGES: Mathematical Connections in Art, Music and Science*, Banff, Alberta,, Canada, 2005, pp. 47–56.

[44] G. T. Toussaint, *The Geometry of Musical Rhythm: What Makes a 'Good' Rhythm Good?* Chapman and Hall/CRC, 2013.

[45] A. Danielsen, 'The Sound of Crossover: Microrhythm and Sonic Pleasure in Michael Jackson's "Don't Stop 'Til You Get Enough"', *Popular Music and Society*, vol. 35, no. 2, pp. 151–168, 2012, doi: 10.1080/03007766.2011.616298.

[46] D. Moelants, 'The Performance of Notes Inégales: The Influence of Tempo, Musical Structure, and Individual Performance Style on Expressive Timing', *Music Perception*, vol. 28, no. 5, pp. 449–460, Jun. 2011, doi: 10.1525/mp.2011.28.5.449.

[47] K. Hellmer and G. Madison, 'Quantifying Microtiming Patterning and Variability in Drum Kit Recordings: A Method and Some Data', *Music Perception: An Interdisciplinary Journal*, vol. 33, no. 2, pp. 147–162, Dec. 2015, doi: 10.1525/mp.2015.33.2.147.

[48] J. London, 'Commentary', *Music Theory Online*, vol. 16, no. 2, 2010.

[49] H. Neuhoff, R. Polak, and T. Fischinger, 'Perception and Evaluation of Timing Patterns in Drum Ensemble Music from Mali', *Music Perception*, vol. 34, no. 4, pp. 438–451, Apr. 2017, doi: 10.1525/mp.2017.34.4.438.

Papers – Session II

CLAMP: CONTRASTIVE LANGUAGE-MUSIC PRE-TRAINING FOR CROSS-MODAL SYMBOLIC MUSIC INFORMATION RETRIEVAL

Shangda Wu[1] **Dingyao Yu**[2] **Xu Tan**[2] **Maosong Sun**[1,3]
[1] Central Conservatory of Music, China [2] Microsoft Research Asia
[3] Tsinghua University, China
shangda@mail.ccom.edu.cn, {v-dingyaoyu, xuta}@microsoft.com, sms@tsinghua.edu.cn
https://ai-muzic.github.io/clamp

ABSTRACT

We introduce **CLaMP**: **C**ontrastive **La**nguage-**M**usic **P**re-training, which learns cross-modal representations between natural language and symbolic music using a music encoder and a text encoder trained jointly with a contrastive loss. To pre-train CLaMP, we collected a large dataset of 1.4 million music-text pairs. It employed text dropout as a data augmentation technique and bar patching to efficiently represent music data which reduces sequence length to less than 10%. In addition, we developed a masked music model pre-training objective to enhance the music encoder's comprehension of musical context and structure. CLaMP integrates textual information to enable semantic search and zero-shot classification for symbolic music, surpassing the capabilities of previous models. To support the evaluation of semantic search and music classification, we publicly release WikiMusicText (WikiMT), a dataset of 1010 lead sheets in ABC notation, each accompanied by a title, artist, genre, and description. In comparison to state-of-the-art models that require fine-tuning, zero-shot CLaMP demonstrated comparable or superior performance on score-oriented datasets. Our models and code are available at `https://github.com/microsoft/muzic/tree/main/clamp`.

1. INTRODUCTION

Symbolic Music Information Retrieval (MIR) is a field that deals with the automatic analysis and retrieval of music based on symbolic representations such as sheet music or MIDI files. Symbolic MIR has numerous practical applications, including music genre classification [1, 2], automatic music transcription [3, 4], and music recommendation systems [5]. However, traditional symbolic MIR approaches based on handcrafted features are often limited in their ability to capture the complex nature of music.

Deep learning has become increasingly popular in symbolic MIR [6–9] due to its ability to extract complex and abstract music features from large datasets. However, obtaining sufficient labelled data can be costly and time-consuming, as most labelled symbolic music datasets are small in size [10–12]. To address this issue, semantic search and zero-shot classification techniques can be used to retrieve and label extensive unlabelled data. These techniques enable the search for music by a given open-domain query (e.g., "*upbeat music with a fast tempo*"), or the automatic identification of music characteristics based on customized labels without the need for training data.

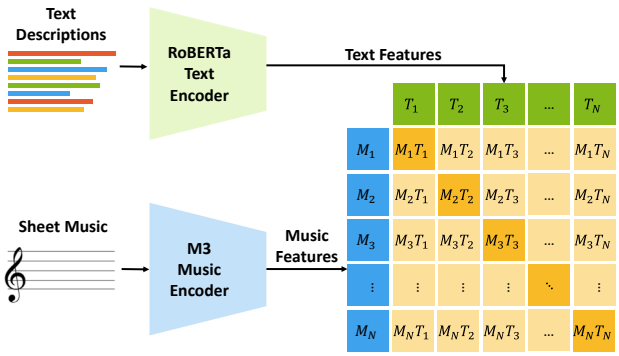


Figure 1. The architecture of CLaMP, including two encoders - one for music and one for text - trained jointly with a contrastive loss to learn cross-modal representations.

To enable semantic search and zero-shot classification for symbolic music, it is necessary to establish a connection between music and language. This can be achieved through the use of contrastive learning [13–17] and pre-training [18–20]. Contrastive learning trains models to learn a feature space where similar sample pairs are grouped and dissimilar pairs are separated, while pre-training involves training a model on a large dataset that can be fine-tuned or directly applied to a specific task.

In this paper, we introduce a solution for cross-modal symbolic MIR that utilizes contrastive learning and pre-training. The proposed approach, **CLaMP**: **C**ontrastive **La**nguage-**M**usic **P**re-training, is inspired by the success of vision-language models [13]. Unlike prior models that rely solely on symbolic music [9, 12, 21], CLaMP learns semantically rich representations of musical concepts from both sheet music and natural language. The contributions of this paper are as follows:

© S. Wu, D. Yu, X. Tan, and M. Sun. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).
Attribution: S. Wu, D. Yu, X. Tan, and M. Sun, "CLaMP: Contrastive Language-Music Pre-training for Cross-Modal Symbolic Music Information Retrieval", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.*, Milan, Italy, 2023.

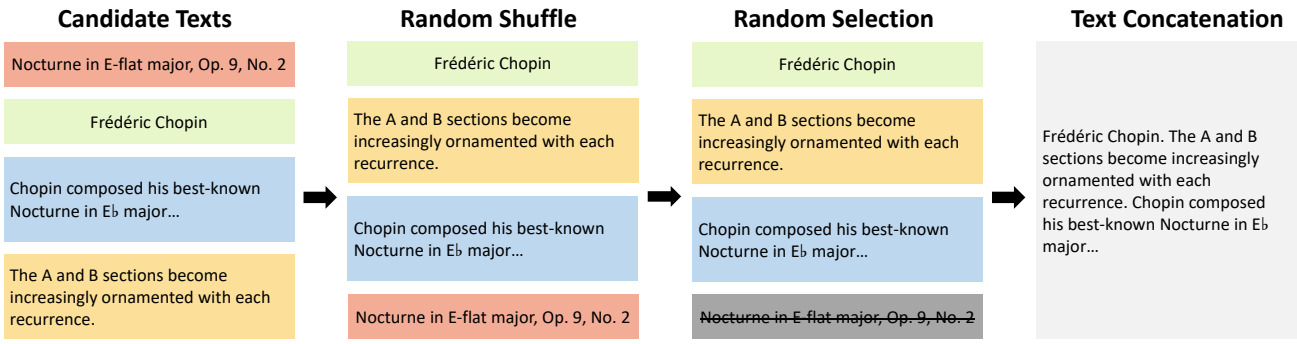


Figure 2. Text dropout is a data augmentation technique that involves a process in which candidate texts are shuffled randomly and then selected to form a concatenated text. In this example, three candidate texts were randomly selected and concatenated to produce the input text.

- CLaMP is a cross-modal model for symbolic MIR, which is pre-trained on WebMusicText (WebMT), a dataset of 1.4 million music-text pairs. To the best of our knowledge, this is the first model of its kind and it achieves comparable or better performance than existing state-of-the-art models without training.
- We propose multiple techniques to improve contrastive language-music pre-training. Our proposed techniques include applying text dropout as a data augmentation method, utilizing bar patching for efficient music representation, and implementing the masked music model pre-training objective.
- The cross-modal pre-training empowers CLaMP to perform tasks beyond the capabilities of unimodal models. It possesses unique features such as semantic search for desired music using open-domain text queries and zero-shot classification for new music.
- To facilitate the evaluation of semantic search and music classification, we release the WikiMusicText (WikiMT) dataset, which consists of 1010 music-text pairs sourced from Wikifonia and Wikipedia.

2. METHODOLOGY

This section presents CLaMP and its cross-modal symbolic MIR abilities. Additionally, we describe the WebMT dataset, which we created to pre-train our model.

2.1 Model Design

2.1.1 Contrastive Learning Objective

CLaMP jointly trains music and text encoders to represent the structural and semantic aspects of both modalities in a shared feature space. This is achieved using a batch construction method and objective [22, 23], as illustrated in Fig. 1, whereby the correct pairings of a batch of N music-text pairs are predicted. The music and text encoders employ global average pooling to obtain corresponding features from the last hidden states.

The objective of CLaMP is to minimize the distance between N paired music-text examples while maximizing the distance between $N^2 - N$ unpaired examples. We denote a batch of N music-text pairs as $(m_i, t_i)_{i=1}^{N}$, where m_i and t_i represent the i-th music and text inputs, respectively. The music and text encoders are represented as f_m and f_t. The contrastive loss for $(m_i, t_i)_{i=1}^{N}$ is defined as follows:

$$\mathcal{L}_{CL} = -\frac{1}{2N}\sum_{i=1}^{N}(\log \frac{\exp(f_m(m_i)\cdot f_t(t_i)/\tau)}{\sum_{j=1}^{N} \mathbb{1}_{i\neq j} \exp(f_m(m_i)\cdot f_t(t_j)/\tau)} + \log \frac{\exp(f_m(m_i)\cdot f_t(t_i)/\tau)}{\sum_{j=1}^{N} \mathbb{1}_{i\neq j} \exp(f_m(m_j)\cdot f_t(t_i)/\tau)}), \quad (1)$$

where τ is a temperature hyper-parameter that controls the sharpness of the softmax distribution, and $\mathbb{1}_{i\neq j}$ is an indicator function that equals 1 if $i \neq j$, and 0 otherwise. The two terms in Eq. 1 consider either music-to-text or text-to-music logits.

2.1.2 Text Encoder

CLaMP includes a text encoder to extract musically relevant features from the input text. To achieve optimal performance, a pre-trained language model is used to initialize the text encoder. Furthermore, text dropout is employed as a data augmentation technique to prevent overfitting and improve the generalization ability of the text encoder.

Pre-trained Language Model RoBERTa [24] is a transformer-based language model pre-trained on a large corpus of English text using the Masked Language Modeling (MLM) objective [18]. This model is designed to be fine-tuned on downstream tasks and has demonstrated excellent performance as a text encoder for the contrastive language-audio pre-training [25]. To improve training efficiency, we used DistilRoBERTa [26] instead, which has fewer parameters (82M) compared to RoBERTa-base (125M) while achieving comparable performance.

Text Dropout Text dropout is a data augmentation technique that encourages models to learn robust features from input texts. This technique involves using a dataset consisting of multiple paired candidate texts from various sources for each musical composition. Similar to [27], for a given composition with L candidates, text dropout shuffles the set of candidate texts and randomly selects K texts, where K is uniformly and randomly sampled from integers ranging from 1 to L. These selected texts are concatenated to form a single input text for the text encoder, as shown in

Table 1. The average number of tokens per lead sheet in the WikiMT dataset with different encoding methods.

Encoding	Bar Patching	ABC Notation	OctupleMIDI [9]
Tokens	**47.07±21.60**	749.16±379.56	469.09±256.43

Fig. 2. Text dropout offers a wider range of possible text combinations and allows the model to learn more complex and diverse textual features.

2.1.3 Music Encoder

The CLaMP music encoder is designed to understand the complex musical structure and context within ABC notation. As a text-based format for symbolic music, ABC notation incorporates a wide range of musical symbols commonly used in sheet music. To keep all musical information while shortening sequence length, the encoding process utilizes the bar patching technique. To optimize performance, the music encoder is specifically designed for symbolic music understanding based on bar patching.

Bar Patching The bar in musical notation groups phrases by defining a fixed number of beats and each bar can be read and played as a single unit. It is separated by vertical lines, providing reference points for locating positions within a score.

Previous models [28–31] for ABC notation utilized character-based tokenization, resulting in sequences that are too lengthy to process efficiently. On the other hand, MeasureVAE [32] demonstrated the feasibility of encoding scores at the bar-level for music generation. To improve the efficiency of processing, we proposed bar patching, inspired by patch-based techniques in computer vision [33].

Bar patching divides a score into several small segments corresponding to bars or headers (i.e. meta-information) in ABC notation. In our implementation, each patch is assigned a maximum of 64 characters, covering 98.8% of the headers or bars in the pre-training dataset. We add an `[END]` token at the end of each patch to indicate the end of the sequence. Patches with fewer than 64 are padded with `[PAD]` tokens, while those with over 64 characters are truncated. For the vocabulary, 95 ASCII printable characters and three special tokens (i.e., `[PAD]`, `[MASK]`, and `[END]`) are considered, resulting in a total of 98 tokens. Thus, each patch can be represented as a 64×98 matrix. These patches are then flattened and projected into 768 dimensions embeddings and used as input tokens, as illustrated in Fig. 3.

Bar patching effectively reduces the average sequence length of the encoded music to less than 10% of the original ABC notation, as shown in Table 1. This technique improves the efficiency of representing music and facilitates faster computation while preserving all musical information in the notation.

Masked Music Model The Masked Music Model (M3) is a self-supervised model for symbolic MIR based on bar patching representation. The primary concept of M3 is to introduce random noise to certain patches of the input music, and then reconstruct the characters in the noise-added bar patches based on the context. This pre-training enables M3 to learn from unlabelled musical data, making it useful for initializing the CLaMP music encoder.

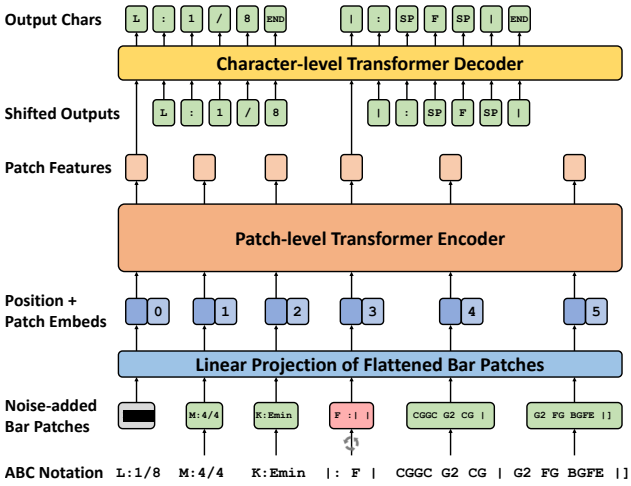


Figure 3. The masked music model architecture, where the encoder takes in a sequence of patches, and the decoder reconstructs character information of noise-added patches.

M3 is based on an asymmetric encoder-decoder architecture, similar to MAE [34], as shown in Fig. 3. It uses an encoder to extract contextualized features of individual patches, along with a decoder, which is lightweight and autoregressively reconstructs the characters for each patch. After pre-training, the decoder is discarded and the encoder is used to initialize the music encoder of CLaMP.

The pre-training objective is inspired by MLM [18]. We first randomly select $M\%$ of the bar patches in the input music, and then the noise is added in three different ways:

- *Masking:* 80% of the selected bar patches are replaced with a special patch filled with `[MASK]` tokens. This encourages the model to learn to fill in missing information and understand the relationship between different musical elements.
- *Shuffling:* 10% of the selected bar patches are randomly shuffled internally. For example, a bar patch "`|: F |`" may be randomly shuffled to "`F :| |`" as shown in Fig. 3. This forces the model to learn the patterns and structures within bar patches.
- *Unchanged:* 10% of the selected bar patches are left unchanged. This can narrow down the gap between pre-training and fine-tuning.

M3 is trained to predict the original characters in the noise-added bar patches based on contextualized patch features. The model is optimized using the cross-entropy loss, which compares the predicted characters with the ground truth characters. The final objective is to minimize the average loss over all the noise-added bar patches in the training set. By denoising these bar patches, M3 learns to capture the dependencies and relationships between different musical elements and structures, allowing it to extract meaningful features from ABC notation.

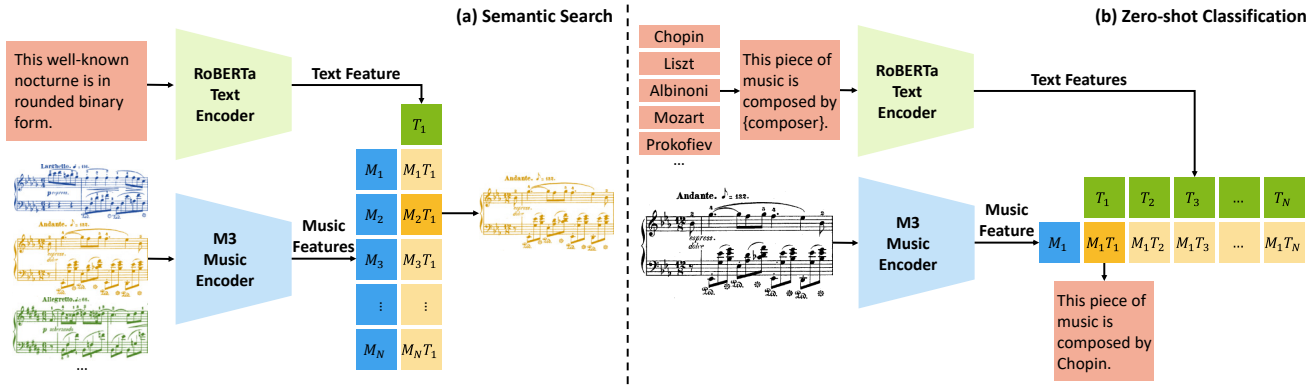


Figure 4. The processes of CLaMP performing cross-modal symbolic MIR tasks, including semantic search and zero-shot classification for symbolic music, without requiring task-specific training data.

2.2 Cross-Modal Symbolic MIR

CLaMP is capable of aligning symbolic music and natural language, which can be used for various cross-modal retrieval tasks, including semantic search and zero-shot classification for symbolic music.

Semantic search is a technique for retrieving music by open-domain queries, which differs from traditional keyword-based searches that depend on exact matches or meta-information. This involves two steps: 1) extracting music features from all scores in the library, and 2) transforming the query into a text feature. By calculating the similarities between the text feature and the music features, it can efficiently locate the score that best matches the user's query in the library.

Zero-shot classification refers to the classification of new items into any desired label without the need for training data. It involves using a prompt template to provide context for the text encoder. For example, a prompt such as "`This piece of music is composed by {composer}.`" is utilized to form input texts based on the names of candidate composers. The text encoder then outputs text features based on these input texts. Meanwhile, the music encoder extracts the music feature from the unlabelled target symbolic music. By calculating the similarity between each candidate text feature and the target music feature, the label with the highest similarity is chosen as the predicted one.

2.3 WebMusicText Dataset

To facilitate the learning of relationships between natural language and symbolic music, we developed a dataset named WebMusicText (WebMT) by crawling an extensive collection of music-text pairs from the web. Our dataset comprises 1,448,750 pairs of music-text data, where all music files are in score-oriented formats (e.g., MusicXML, LilyPond, and ABC notation). To reduce the disparity between scores in different notations, we first converted all music files to MusicXML and then to ABC notation [1]. In addition, to avoid information leakage, we removed any natural language (e.g., titles, composers, and lyrics) in ABC notation. The text parts of each pair were obtained from corresponding meta-information (e.g., title and composer) or user comments, and are all in English. WebMT features diverse musical compositions, from monophonic folk music to polyphonic orchestral music, which enables the model to learn a wide range of musical information.

[1] `https://wim.vree.org/svgParse/xml2abc.html`

3. EXPERIMENTS

3.1 Settings

3.1.1 Models

- **MusicBERT [9]**: This model combines unsupervised pre-training with supervised fine-tuning, which achieved state-of-the-art results. MusicBERT is available in two settings: *`MusicBERT-S/1024`* ($\text{MusicBERT}_{\text{small}}$), and *`MusicBERT-B/1024`* ($\text{MusicBERT}_{\text{base}}$). MusicBERT-S/1024 consists of 4 layers and was pre-trained on the small-scale Lakh MIDI Dataset (LMD, 148,403 pieces) [35], while MusicBERT-B/1024 has 12 layers and was pre-trained on the large-scale Million MIDI Dataset (MMD, 1,524,557 pieces). Both models have a maximum length of 1024.

- **M3**: Our proposed music encoder is used to compare the performances of unimodal and multimodal models trained on the same dataset (i.e., WebMT). M3 comes with two settings: *`M3-S/512`* and *`M3-S/1024`*, with maximum lengths of 512 and 1024, respectively. In the following experiments, both settings use the 6 encoder layers only.

- **CLaMP**: Several variants were tested to verify the effectiveness of the proposed techniques for improving contrastive language-music pre-training. These include *`CLaMP-S/512`* which is the full model, *`CLaMP-S/512 (w/o TD)`* which removes text dropout, *`CLaMP-S/512 (w/o M3)`* which has a randomly initialized music encoder, and *`CLaMP-S/512 (w/o M3, BP)`* which removes both M3 and bar patching, and uses char-level tokenization to encode raw ABC notation instead. *`CLaMP-S/1024`* was included to verify the effectiveness of an extended maximum length.

3.1.2 Pre-training

The text encoder was initialized using DistilRoBERTa [26], with a maximum length of 128, and the music encoder was initialized using two settings: M3-S/512 and M3-S/1024. A length of 512 resulted in truncating 17.29% of compositions in WebMT, while a length of 1024 reduced truncation to 7.7%. Both models were trained for 40 epochs with 6 encoder layers and 3 decoder layers, an embedding size of 768, and a noise ratio of 45%. Based on these two M3 encoders, we developed CLaMP-S/512 and CLaMP-S/1024. Both of them were trained for 20 epochs, using the AdamW optimizer [36] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, and a weight decay coefficient of 0.01. The batch size is set to 640, and the temperature $\tau = 0.2$. The training process was accelerated and memory was saved by using mixed precision [37].

3.1.3 Evaluation Datasets

We introduce WikiMusicText (WikiMT) [2], a new dataset for the evaluation of semantic search and music classification. It includes 1010 lead sheets (melodies with harmonies) in ABC notation sourced from Wikifonia, each accompanied by a title, artist, genre, and description. The title and artist information is extracted from the score, whereas the genre labels are obtained by matching keywords from the Wikipedia entries and assigned to one of the 8 classes that loosely mimic the GTZAN genres [38]. The description is obtained by utilizing BART-large [39] to summarize and clean the corresponding Wikipedia entry. Additionally, following WebMT, the natural language information within the ABC notation is removed.

In addition to WikiMT, we use two other datasets to evaluate music classification: VGMIDI and Pianist8. VGMIDI [11] includes 204 score-oriented MIDI arrangements that were classified according to the valence-arousal model. Pianist8 [12] contains symbolic piano performances of 411 pieces from 8 composers with distinct styles, which were automatically transcribed from audio using a model presented in [8].

3.1.4 Metrics

We use the following three metrics to evaluate the effectiveness of models in various downstream tasks:

- Mean Reciprocal Rank (MRR) is used to evaluate ranking systems. This metric calculates the average of the reciprocal ranks of the correct answers, which measures the effectiveness of the ranking.
- Hit Ratio at K (HR@K) measures the accuracy of the model by checking if the correct item is among the top K recommendations, which is often used in recommendation systems.
- F1-macro score is a metric that assesses the overall effectiveness of a classification model. It is computed using the arithmetic mean (i.e., unweighted mean) of all the per-class F1 scores.

[2] https://huggingface.co/datasets/sander-wood/wikimt

Table 2. Semantic search performance of CLaMP on WikiMT (1010 music-text pairs) under different settings.

Setting	MRR	HR@1	HR@10	HR@100
S/512	**0.2561**	**0.1931**	**0.3693**	**0.7020**
S/1024	0.2016	0.1436	0.3109	0.6554
S/512 (w/o TD)	0.1841	0.1248	0.2911	0.6188
S/512 (w/o M3)	0.1262	0.0802	0.1960	0.5119
S/512 (w/o M3, BP)	0.0931	0.0525	0.1584	0.4426

3.2 Results

3.2.1 Semantic Search

In the semantic search evaluation, we assessed different versions of CLaMP for semantic search, aiming to test the efficacy of contrastive language-music pre-training techniques. The pre-training dataset WebMT and the evaluation dataset WikiMT have no overlap, thus guaranteeing the validity of our evaluation results. In addition, as semantic search requires no additional training for this dataset, it demonstrates the generalizability of CLaMP.

Table 2 shows that our full model (CLaMP-S/512) outperforms all other models across all metrics. Interestingly, we discovered that increasing the maximum sequence length to 1024 (CLaMP-S/1024) did not lead to an improvement in performance. We attribute this to the fact that all lead sheets in the WikiMT dataset, once encoded with bar patching, have a length smaller than 512, which limits the potential advantages of the longer sequence length of CLaMP-S/1024. We also observed that the removal of the proposed techniques from CLaMP had a considerable negative impact on semantic search performance. Notably, the removal of M3 pre-training had the greatest effect on model performance, followed by text dropout and bar patching.

In conclusion, our evaluation of CLaMP on WikiMT shows that CLaMP-S/512 with all proposed contrastive language-music pre-training techniques is the most effective for the semantic search task. This highlights the importance of these techniques for effective pre-training and semantic search tasks. Additionally, increasing the sequence length (CLaMP-S/1024) did not improve the model's performance. These results emphasize the significance of using appropriate pre-training techniques in language-music models and suggest that a longer sequence length may not necessarily result in better outcomes.

3.2.2 Music Classification

The goal of the classification evaluation is to assess how well the zero-shot CLaMP models perform compared to other fine-tuned models. In addition, to evaluate pre-trained models, linear probes are used to train a linear classifier for the classification based on the features from pre-trained models. Despite being less powerful and relying on pre-trained model features, linear classifiers offer a valuable means of quantitatively assessing feature quality [40].

Table 3. Classification performance of different models on three datasets: WikiMT (1010 pieces, 8 genres), VGMIDI (204 pieces, 4 emotions), and Pianist8 (411 pieces, 8 composers).

Model	WikiMT		VGMIDI [11]		Pianist8 [12]	
	F1-macro	*Accuracy*	*F1-macro*	*Accuracy*	*F1-macro*	*Accuracy*
Linear Probe MusicBERT-S/1024	0.2401	**0.3507**	0.4662	0.5350	0.8047	0.8102
Linear Probe MusicBERT-B/1024	0.1746	0.3219	0.5127	0.5850	**0.8379**	**0.8413**
Zero-shot CLaMP-S/512	**0.2660**	0.3248	**0.5217**	**0.6176**	0.2180	0.2512
Zero-shot CLaMP-S/1024	0.2248	0.3406	0.4678	0.5049	0.1509	0.2390
Linear Probe M3-S/512	0.2832	0.3990	0.5991	0.6667	0.6773	0.6909
Linear Probe M3-S/1024	0.3079	0.4020	0.5966	0.6522	0.6844	0.6958
Linear Probe CLaMP-S/512	**0.3452**	0.4267	**0.6453**	**0.6866**	0.7067	0.7152
Linear Probe CLaMP-S/1024	0.3449	**0.4416**	0.6345	0.6720	**0.7271**	**0.7298**

WikiMT was converted into the MIDI format using music21 [41] to be compatible with MusicBERT. In contrast, for VGMIDI and Pianist8, we employed MuseScore3's batch conversion tool [3] to convert the scores into the MusicXML format, which were then converted into ABC notation for use with M3 and CLaMP.

We conducted 5-fold cross-validation with the same folds to assess all linear probe models, using identical fine-tuning settings and a batch size of 10 to ensure consistency, given the limited size of the evaluation datasets. The linear probe CLaMP models used the music encoder only, while the text encoder was discarded. In the zero-shot classification setting, CLaMP had no previous exposure to these evaluation datasets during pre-training. We utilized manually designed prompts for the zero-shot CLaMP models.

The top half of Table 3 presents the comparison of the performance between linear probe MusicBERT and zero-shot CLaMP. The results found that the zero-shot CLaMP models demonstrated comparable or even superior performance compared to the linear probe MusicBERT models on WikiMT and VGMIDI datasets. Interestingly, the smaller zero-shot CLaMP-S/512 outperformed the larger linear probe MusicBERT-B/1024, indicating that the pre-training of CLaMP has enabled it to learn more generalizable features that are useful for zero-shot music classification. However, this trend was not observed on Pianist8, where MusicBERT models performed much better than zero-shot CLaMP models. This difference in performance can be attributed to the source of the datasets, as WikiMT and VGMIDI primarily focus on score information, whereas Pianist8 contains performance MIDI data derived from audio. Since both CLaMP and M3 were trained exclusively on score information, they lack knowledge of performance MIDI. However, we noticed that the performances of linear probe CLaMP models on Pianist8 significantly improved after fine-tuning compared to the zero-shot ones. This suggests that incorporating ABC notation from performance MIDI into the pre-training of CLaMP may enhance its ability to comprehend such data.

[3] `https://musescore.org/en/project/batch-convert`

The linear probe CLaMP models show better performance compared to the linear probe M3 models, as indicated in the bottom half of Table 3, despite being pre-trained on the same dataset with the same architecture. This is attributed to the use of contrastive learning, which aligns the music encoder of CLaMP with the text modality, thus implicitly introducing textual information to the music encoder. Furthermore, we found that CLaMP-S/1024 performed better on Pianist8 than CLaMP-S/512, suggesting that a larger maximum length is beneficial for models to learn performance MIDI.

In summary, our evaluation demonstrates that zero-shot CLaMP performs comparably to state-of-the-art models in music classification. Furthermore, the incorporation of contrastive learning and textual information enhances the music encoder's performance, resulting in better classification accuracy when compared to M3 which employed the same architecture. These results highlight the potential of CLaMP as a pre-training framework for symbolic MIR.

4. CONCLUSIONS

This paper introduces CLaMP, a pre-trained model that utilizes contrastive language-music pre-training techniques to build cross-modal representations between natural language and symbolic music. The model was trained on a dataset containing 1.4 million music-text pairs and has demonstrated unique abilities of semantic search and zero-shot classification for symbolic music. Compared to state-of-the-art models that require fine-tuning, zero-shot CLaMP exhibits comparable or superior performance in score-oriented music classification tasks without any training. However, the current version of CLaMP has limited comprehension of performance MIDI, and still has room for improvement. Future research will aim to expand its capabilities by scaling it up and pre-training it on larger datasets that incorporate a wider range of symbolic music formats beyond score-oriented ones. We expect that its cross-modal representations will facilitate research on new topics in music analysis, retrieval, and generation, and provide a foundation for the development of innovative systems and applications that integrate music and language.

5. REFERENCES

[1] I. Karydis, "Symbolic music genre classification based on note pitch and duration," in *Advances in Databases and Information Systems, 10th East European Conference, ADBIS 2006, Thessaloniki, Greece, September 3-7, 2006, Proceedings*, ser. Lecture Notes in Computer Science, Y. Manolopoulos, J. Pokorný, and T. K. Sellis, Eds., vol. 4152. Springer, 2006, pp. 329–338. [Online]. Available: https://doi.org/10.1007/11827252_25

[2] C. Kofod and D. O. Arroyo, "Exploring the design space of symbolic music genre classification using data mining techniques," in *2008 International Conferences on Computational Intelligence for Modelling, Control and Automation (CIMCA 2008), Intelligent Agents, Web Technologies and Internet Commerce (IAWTIC 2008), Innovation in Software Engineering (ISE 2008), 10-12 December 2008, Vienna, Austria*, M. Mohammadian, Ed. IEEE Computer Society, 2008, pp. 43–48. [Online]. Available: https://doi.org/10.1109/CIMCA.2008.223

[3] J. P. Bello, G. Monti, and M. B. Sandler, "Techniques for automatic music transcription," in *ISMIR 2000, 1st International Symposium on Music Information Retrieval, Plymouth, Massachusetts, USA, October 23-25, 2000, Proceedings*, 2000. [Online]. Available: http://ismir2000.ismir.net/papers/bello_paper.pdf

[4] C. Raphael, "Automatic transcription of piano music," in *ISMIR 2002, 3rd International Conference on Music Information Retrieval, Paris, France, October 13-17, 2002, Proceedings*, 2002. [Online]. Available: http://ismir2002.ismir.net/proceedings/02-FP01-2.pdf

[5] C. Walshaw, "A statistical analysis of the abc music notation corpus: Exploring duplication," 2014.

[6] C. Hawthorne, I. Simon, R. Swavely, E. Manilow, and J. H. 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*, J. H. Lee, A. Lerch, Z. Duan, J. Nam, P. Rao, P. van Kranenburg, and A. Srinivasamurthy, Eds., 2021, pp. 246–253. [Online]. Available: https://archives.ismir.net/ismir2021/paper/000030.pdf

[7] J. Gardner, I. Simon, E. Manilow, C. Hawthorne, and J. H. Engel, "MT3: multi-task multitrack music transcription," in *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. [Online]. Available: https://openreview.net/forum?id=iMSjopcOn0p

[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 Trans. Audio Speech Lang. Process.*, vol. 29, pp. 3707–3717, 2021. [Online]. Available: https://doi.org/10.1109/TASLP.2021.3121991

[9] 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, C. Zong, F. Xia, W. Li, and R. Navigli, Eds., vol. ACL/IJCNLP 2021. Association for Computational Linguistics, 2021, pp. 791–800. [Online]. Available: https://doi.org/10.18653/v1/2021.findings-acl.70

[10] 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, DLfM 2018, Paris, France, September 28, 2018*, K. R. Page, Ed. ACM, 2018, pp. 34–37. [Online]. Available: https://doi.org/10.1145/3273024.3273035

[11] L. N. Ferreira and J. Whitehead, "Learning to generate music with sentiment," 2019.

[12] Y. Chou, I. Chen, C. Chang, J. Ching, and Y. Yang, "Midibert-piano: Large-scale pre-training for symbolic music understanding," *CoRR*, vol. abs/2107.05223, 2021. [Online]. Available: https://arxiv.org/abs/2107.05223

[13] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," in *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 2021, pp. 8748–8763. [Online]. Available: http://proceedings.mlr.press/v139/radford21a.html

[14] W. Kim, B. Son, and I. Kim, "Vilt: Vision-and-language transformer without convolution or region supervision," in *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 2021, pp. 5583–5594. [Online]. Available: http://proceedings.mlr.press/v139/kim21k.html

[15] J. Li, D. Li, C. Xiong, and S. C. H. Hoi, "BLIP: bootstrapping language-image pre-training for unified vision-language understanding and generation," in *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, ser. Proceedings of Machine Learning Research, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvári, G. Niu, and S. Sabato, Eds., vol. 162. PMLR, 2022, pp. 12 888–12 900. [Online]. Available: https://proceedings.mlr.press/v162/li22n.html

[16] B. Elizalde, S. Deshmukh, M. A. Ismail, and H. Wang, "CLAP: learning audio concepts from natural language supervision," *CoRR*, vol. abs/2206.04769, 2022. [Online]. Available: https://doi.org/10.48550/arXiv.2206.04769

[17] 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," *CoRR*, vol. abs/2208.12415, 2022. [Online]. Available: https://doi.org/10.48550/arXiv.2208.12415

[18] 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)*, J. Burstein, C. Doran, and T. Solorio, Eds. Association for Computational Linguistics, 2019, pp. 4171–4186. [Online]. Available: https://doi.org/10.18653/v1/n19-1423

[19] 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 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., 2020. [Online]. Available: https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html

[20] 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

[21] 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*, J. H. Lee, A. Lerch, Z. Duan, J. Nam, P. Rao, P. van Kranenburg, and A. Srinivasamurthy, Eds., 2021, pp. 722–729. [Online]. Available: https://archives.ismir.net/ismir2021/paper/000090.pdf

[22] K. Sohn, "Improved deep metric learning with multi-class n-pair loss objective," in *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, D. D. Lee, M. Sugiyama, U. von Luxburg, I. Guyon, and R. Garnett, Eds., 2016, pp. 1849–1857. [Online]. Available: https://proceedings.neurips.cc/paper/2016/hash/6b180037abbebea991d8b1232f8a8ca9-Abstract.html

[23] 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

[24] 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. [Online]. Available: http://arxiv.org/abs/1907.11692

[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," *CoRR*, vol. abs/2211.06687, 2022. [Online]. Available: https://doi.org/10.48550/arXiv.2211.06687

[26] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," *ArXiv*, vol. abs/1910.01108, 2019.

[27] 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, pp. 1–5.

[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.

[30] S. Wu and M. Sun, "Tunesformer: Forming tunes with control codes," *CoRR*, vol. abs/2301.02884, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2301.02884

[31] ——, "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

[32] A. Pati, A. Lerch, and G. Hadjeres, "Learning to traverse latent spaces for musical score inpainting," 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. 343–351. [Online]. Available: http://archives.ismir.net/ismir2019/paper/000040.pdf

[33] 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. [Online]. Available: https://openreview.net/forum?id=YicbFdNTTy

[34] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. B. Girshick, "Masked autoencoders are scalable vision learners," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*. IEEE, 2022, pp. 15 979–15 988. [Online]. Available: https://doi.org/10.1109/CVPR52688.2022.01553

[35] C. Raffel, "Learning-based methods for comparing sequences, with applications to audio-to-midi alignment and matching," Ph.D. dissertation, PhD Thesis, 2016.

[36] 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

[37] P. Micikevicius, S. Narang, J. Alben, G. F. Diamos, E. Elsen, D. García, B. Ginsburg, M. Houston, O. Kuchaiev, G. Venkatesh, and H. Wu, "Mixed precision training," in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. [Online]. Available: https://openreview.net/forum?id=r1gs9JgRZ

[38] B. L. Sturm, "An analysis of the GTZAN music genre dataset," in *Proceedings of the second international ACM workshop on Music information retrieval with user-centered and multimodal strategies, MIRUM '12, Nara, Japan, October 29 - November 02, 2012*, C. C. S. Liem, M. Müller, S. K. Tjoa, and G. Tzanetakis, Eds., 2012, pp. 7–12. [Online]. Available: https://doi.org/10.1145/2390848.2390851

[39] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, D. Jurafsky, J. Chai, N. Schluter, and J. R. Tetreault, Eds. Association for Computational Linguistics, 2020, pp. 7871–7880. [Online]. Available: https://doi.org/10.18653/v1/2020.acl-main.703

[40] Y. M. Asano, C. Rupprecht, and A. Vedaldi, "A critical analysis of self-supervision, or what we can learn from a single image," in *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. [Online]. Available: https://openreview.net/forum?id=B1esx6EYvr

[41] 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, August 9-13, 2010*, J. S. Downie and R. C. Veltkamp, Eds. International Society for Music Information Retrieval, 2010, pp. 637–642. [Online]. Available: http://ismir2010.ismir.net/proceedings/ismir2010-108.pdf

GENDER-CODED SOUND: ANALYSING THE GENDERING OF MUSIC IN TOY COMMERCIALS VIA MULTI-TASK LEARNING

Luca Marinelli **György Fazekas** **Charalampos Saitis**
C4DM, Queen Mary University of London, UK
{l.marinelli, c.saitis, george.fazekas}@qmul.ac.uk

ABSTRACT

Music can convey ideological stances, and gender is just one of them. Evidence from musicology and psychology research shows that gender-loaded messages can be reliably encoded and decoded via musical sounds. However, much of this evidence comes from examining music in isolation, while studies of the gendering of music within multimodal communicative events are sparse. In this paper, we outline a method to automatically analyse how music in TV advertising aimed at children may be deliberately used to reinforce traditional gender roles. Our dataset of 606 commercials included music-focused mid-level perceptual features, multimodal aesthetic emotions, and content analytical items. Despite its limited size, and because of the extreme gender polarisation inherent in toy advertisements, we obtained noteworthy results by leveraging multi-task transfer learning on our densely annotated dataset. The models were trained to categorise commercials based on their intended target audience, specifically distinguishing between masculine, feminine, and mixed audiences. Additionally, to provide explainability for the classification in gender targets, the models were jointly trained to perform regressions on emotion ratings across six scales, and on mid-level musical perceptual attributes across twelve scales. Standing in the context of MIR, computational social studies and critical analysis, this study may benefit not only music scholars but also advertisers, policymakers, and broadcasters.

1. INTRODUCTION

The purpose of this study is to analyse gender-coding in a context where music is secondary to other modalities and serves a clear purpose, such as in advertisement. Our aim is to investigate how music may be employed to reinforce traditional gender roles in toy commercials, and we propose an automatic method for analyzing this phenomenon. [1] Our overarching research objective is to provide a basis for a theory of message production. Specifically, a theory of the effects that message producers, their decision-making, or their unconscious gender biases have on the selection and composition of sound and music in toy adverts. For this goal, we propose an integrative approach combining content analytical (CA) variables, music perceptual ratings, and multimodal affective ratings to annotate toy commercials, and using multi-task learning (Fig. 1) to analyse the gendering of their soundtracks.

1.1 Gendered music styles as cognitive schemas

Empirical studies have demonstrated that gender and sex impact the perception and processing of music [1–3]. However, the idea that sex determines fixed differences in brain structure has been questioned due to potential misinterpretations, overestimations, and publication bias [4]. Gender schemas, instead, are *learned* cognitive networks of associations that guide an individual's behavior by assimilating or rejecting gender-appropriate ideas and activities [5,6]. Schemas guide an individual's perception, information processing, and memory retention, as they prevent information overload by organising one's perceptual experience into a coherent and intelligible whole [6, 7].

Popular music genres have been themselves theorised as cognitive schemas containing extramusical concepts that can be primed when a subject is exposed to the genre's music [8, 9]. Such schemas are formed through repeated exposure to the multimodal discourse encompassing music, which is to some extent globalised, but that also varies from culture to culture as a result of glocalisation. [2] processes [9] Schema theory has also been used in literary reading and analysis to explain organised "bundles of information and features" [10, p. 106-7] such as literary genres (e.g., science fiction, fantasy, horror).

While gendered language patterns in text and lyrics [11] may be relatively more straightforward to interpret, gender-coding in sound and music ensues from the historical sedimentation, in musical practice, of multimodal associations between gendered meanings in language, visual images, and musical structures [12]. For example, instruments have been consistently associated with masculinity or femininity, even when their sound is presented in isolation and not visually linked to the actual object [13, 14]. Sergeant and Himonides [15, 16] investigated whether in Western art music individual sounds or their organization

[1] https://github.com/marinelliluca/gender_coded_sound_ismir2023

© L. Marinelli, C. Saitis, and G. Fazekas. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).
Attribution: L. Marinelli, C. Saitis, and G. Fazekas, "Gender-coded sound: Analysing the Gendering of Music in Toy Commercials via Multi-task Learning", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

[2] A lexical blend of globalization and localism.

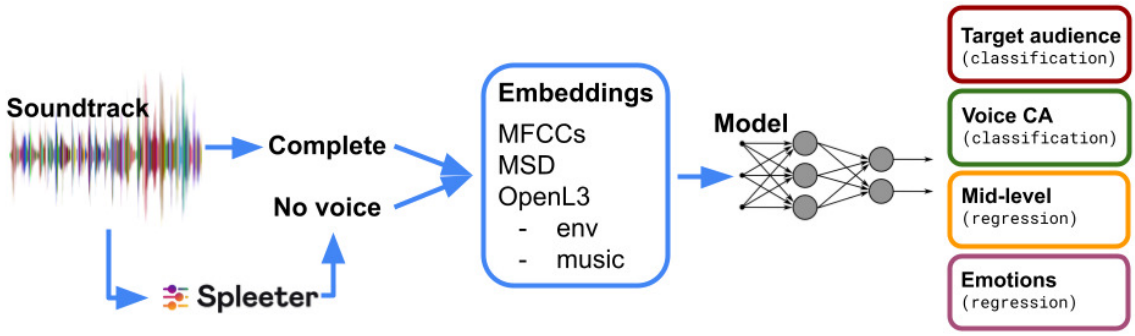


Figure 1. Brief overview of the experimental pipeline.

within a composition could infer the sex or gender of the performer or composer. Even though they found no correlation between the gender of the composers or performers and the gendering of music, raters agreed on the gendering of music, which was associated with features such as tempo, minor/major key, and tonal weight or density. Tagg [17] studied the reception of gendered meanings in TV theme tunes and also found high agreement among participants. Several musical dimensions, such as average tempo, rhythmic and dynamic regularity, and presence of active bass lines, may contribute to conveying gendered meanings. In a subsequent investigation, Tagg and Clarida [18] found that musical pieces linked to female characters were more prone to be classified as quiet and calm. Wang and Horvát [19] computationally extracted twelve descriptors of musical parameters and perceptual features for over 200k songs by more than 8k globally distributed artists across a multitude of popular music genres. They found statistically significant differences for eleven out of twelve musical parameters with regard to the gender of the composers, suggesting the existence of measurable, supra-genre, gendered music styles in the global music industry.

Some of these studies appear to contradict each other, [3] while at the same time sharing the same fallacy, in that feminine and masculine patterns in the performance and composition of music should be considered on a par with distinct gendered styles in spoken language, such as Lakoff's 'women's talk' [20]. As such, these differences should not be understood in terms of a causal relationship between the gender of the artists and gendered musical patterns. Individuals may have a tendency to use forms of expression that they deem appropriate with regard to their identity, but given the performative nature of gender we cannot possibly generalise this behaviour (i.e., even strong correlation is not causation), as this would end up reinforcing gender stereotypes and their power relations.

Gender schemas therefore mediate our perception of music, and this relationship appears to be bidirectional. At the same time, music-primed schemas can alter our perception of other people's ethnicity, rural/urban background, age, expertise [8], and even gender [21, 22]. We thus posit that not only gender roles and stereotypes can be understood in terms of schemas, but also that masculine and feminine music styles can be viewed as *music-primed gender schemas*, which to some extent overlap with the former. We also presume that different music-primed schemas might exist for other intersectional factors, such as class.

1.2 Gendered toy marketing

Gender polarisation in TV advertising aimed at children has been consistently found in a large body of studies spanning over 40 years [23–25]. Differences in commercials targeted at girls, boys, and mixed audience have been found in terms of: sound (voices, background music and sound effects), language, transitions and camera work, setting, interactions and activities, and colours.

Specifically in terms of sound and music, Welch et al. [23] noted that in general the sex of the voice-over matched the target audience of the commercials, but that male narrating voices also occurred more often in mixed audience commercials, and subsequent research confirmed the same trend [25]. They also found that commercials targeted at boys had more noise, louder music, and more sound effects. Another study [24] conversely found that music used in girls' advertisements is generally softer and more likely to have a sung narration style. Whereas, Johnson and Young [26] identified what they called "gender exaggeration:" male voice-overs tend to be exceedingly deep, growl-like or aggressive, whereas female voice-overs are often very high-pitched and singsong.

By interpreting music as an inherently multimodal discourse, a critical analysis of gender markers in children's TV adverts can help to investigate the relation between music and hegemonic discourses on gender; and to promote further research towards a commercial and contemporary musical semiotics of gender. Analysing music in gendered advertising aimed at children allows a privileged glance into the birthplace of music-primed gender schemas.

1.3 Automatic discourse processing

Discourse analysis is an umbrella term that refers to approaches developed across diverse academic disciplines. This includes disciplines that first developed models for understanding discourse, such as linguistics, social semiotics and conversation analysis. But it also refers to other

[3] [19] found significant correlations between the gender of the composers and characteristics of their music, while [16] did not.

All $N = 606$		Feminine $N = 163$	Masculine $N = 149$	Mixed $N = 200$
	Type	$\chi^2(6, N = 512) = 89.02, p = .000$		
5.6%	Sung	9.8%	None	2.5%
18.8%	Spoken and sung	36.8%	6.0%	17.0%
67.7%	Spoken	52.8%	81.2%	75.5%
7.9%	No voices	0.6%	12.8%	5.0%
	Age	$\chi^2(6, N = 512) = 39.51, p = .000$		
79.5%	Adults	76.7%	79.2%	83.0%
5.9%	Children and adults	8.0%	6.0%	6.5%
6.6%	Children	14.7%	2.0%	5.5%
7.9%	No voices	0.6%	12.8%	5.0%
	Gender	$\chi^2(6, N = 512) = 332.1, p = .000$		
39.8%	Feminine	95.7%	2.0%	29.5%
46.9%	Masculine	1.8%	83.9%	54.5%
5.4%	Feminine and masculine	1.8%	1.2%	11.0%
7.9%	No voices	0.6%	12.8%	5.0%
	Gender exaggeration	$\chi^2(6, N = 512) = 243.6, p = .000$		
16.5%	Exagg. feminine	44.8%	None	8.0%
15.5%	Exagg. masculine	None	40.3%	7.0%
60.1%	All normal sounding	54.6%	47.0%	80.0%
7.9%	No voices	0.6%	12.8%	5.0%

Table 1. Contingency tables of voice-related content analytical variables with χ^2 tests of independence. The column "All" includes commercials without actors or presenter (94).

approaches that apply and extend these models of understanding to their particular academic field, such as cognitive psychology, literary criticism and *artificial intelligence* [27]. Research on discourse processing, an endeavour of natural language processing (NLP), is already at a stage where machine learning approaches are able, for example, to automatically detect social attitudes and political stances in online news or social media [28, 29].

Beyond textual discourse and NLP, denotative meanings in images and videos can be easily captured by machine learning techniques [30,31]. However, works that try to address connotative meanings or the rhetoric of multimedia content are still in their infancy and such approaches are often not even framed as pertaining to discourse or semiotic analysis. Dinkov et al. [32] predicted the political ideological bias (left, centre, right) of media outlets using text, metadata, and audio (via speech processing techniques) from YouTube channels, but not visual content. Ye et al. [33] predicted the messages that image and video advertisements convey by explicitly modeling symbolic associations (e.g., gun for "danger") and combining cues from multiple modalities, including the loudness in video soundtracks. Notably, none of these studies leveraged approaches and tools from music information retrieval.

1.4 Multi-task learning in MIR

In multi-task learning we train a single model to perform multiple related tasks simultaneously, leveraging shared information among tasks, which results in several benefits. Böck et al. [34] simultaneously modelled tempo estimation and beat tracking of musical audio, showing state-of-the-art performance for both tasks. Wu et al. [35] combined multi-task and self-supervised learning, resulting in improved performance. Chowdhury et al. [36] proposed a VGG-style deep neural network to predict emotional characteristics of music based on mid-level perceptual features (e.g., melodiousness and tonal stability) and found that the loss in performance was negligible when compared to predicting emotions directly. Further improvements were obtained by training jointly on the mid-level and emotion annotations, with the small loss in performance justified by the gain in explainability of the predictions. Our study expands upon this foundation by incorporating emotions and perceptual features, while also adding more granular structure to facilitate a comprehensive understanding of the gendering of music in multimodal contexts.

2. DATASET

Our hierarchical data collection framework comprised CA variables at the lower level, music-focused ratings from experts at the middle level, and multimodal affective ratings at the highest level of subjectivity. Mid-level perceptual features, which describe relevant and instantly identifiable musical characteristics, exhibit high consistency across listeners and can be predicted from the acoustic signal. These features also correlate with music's affective dimensions [37]. The emotion ratings were collected from adults rather than children because adults are better equipped to capture the commercials' intended emotional impact. Fur-

thermore, research indicates that children exhibit adult-like emotion recognition capabilities by age 11 [38].

2.1 Sampling method

In March 2022, we collected a sample of 5614 videos from the official YouTube channel of Smyths Toys Superstores, a major UK toy retailer. To ensure comparability with previous studies [39,40], we selected only high-quality videos intended for television and excluded those without audio, formatted for mobile phones, or with substantial on-screen text. Additionally, we excluded advertisements featuring toddlers and pre-schoolers as these are actually targeted at parents. To minimise duplicates, we removed videos with the same title from our sample.

Given that we are interested in understanding the gendering of sound and music in the toy industry at large, we needed to enforce some balance across gender targets. We thus performed a *preliminary* classification of 1778 commercials based on their intended target audience (feminine, masculine or mixed audience) using simple heuristics regarding the gender of the majority of presenters featuring in the commercial, the colour coding of the video and ultimately the category of the product. This resulted in 780 'feminine', 509 'masculine', and 489 'mixed audience' commercials. A final sample of 606 commercials, spanning over 10 years from 2012 to 2022, was obtained by randomly sampling from each category 202 videos.

2.2 Content analysis (manual annotation)

The *gender orientation* (also *target* audience) of the commercials was determined by the gender of the actors/presenters. Following [26], in order to account for tokenism, whenever a presenter of the other gender was included in the background or for just a few seconds, these were considered token gender representations and not explicit market orientations. All fictional characters, even when realistic (e.g. from a video game), were not considered as actors/presenters and the corresponding commercials were coded as having no actors. Whenever commercials featured exclusively character 'dismemberment' (e.g., showing only hands without a face or head) [41] these were also coded as having no actors.

Four distinct items describing the sound of the voices in the commercial were collected using a coding schema based on Verna's research [42]. But unlike the original work, we coded for all voices in the commercial, both diegetic and non-diegetic. The reason for this choice is that there is no way to reliably distinguish between diegetic and non-diegetic sounds purely based on the audio signal. Commercials were coded in terms of *type of voices* ("Spoken", "Sung", "Both spoken and sung", "No voices"), then in terms of *voices age* ("Adults" which included young adults, "Children", "Children and adults", "No voices"), *gender exaggeration* of the voices ("All normal sounding", "Exaggeratedly masc.", "Exaggeratedly fem.", "No voices"), and finally in terms of *voice gender* ("Feminine", "Masculine", "Feminine and masculine", "No voices"). In order to determine the reliability of each variable, 15% of the commercials was double-coded by two coders independently. For all variables we obtained Krippendorff's alpha levels above .80 (with 'gender orientation' and 'gender of the voices' exceeding .90), and therefore met the standards of reliability required for this type of analysis [43]. Out of 606 commercials analyzed, 163 were targeted at a feminine audience, 149 at a masculine audience, 200 at a mixed audience and 94 featured no actors or presenters. Contingency tables of the voice variables are shown in Table 1.

2.3 Music-focused and emotion ratings

Participants in our study were paid between £7 and £8 per hour (depending on their completion time) on Prolific.co. In order to minimise the effects of careless responding, a low-effort metric was computed by summing the length of all long strings for each participant, and those that scored above two standard deviations from the average value were screened out during data collection, as it was performed in batches of 50 participants. For 600 of the videos, we collected between five and six ratings on each music and emotion scale. At an initial stage, the remaining 6 videos were used as controls (i.e., were rated by all participants), but we do not leverage them as such in the current study.

Musically trained participants (at least three years of experience with an instrument) rated the soundtracks of the commercials on 15 music-focused bipolar scales [44, 45]: Electric/Acoustic, Distorted/Clear, Loud/Soft, Many/Few instruments, Heavy/Light, High/Low pitch, Punchy/Smooth, Wide/Narrow pitch variation, Harmonious/Disharmonious, Clear melody/No melody, Complex/Simple rhythm, Repetitive/Non-repetitive, Dense/Sparse, Fast/Slow tempo, and Strong/Weak beat. We collected a total of 4560 ratings from 152 participants from the UK (75 M, 77 F, aged 40 ± 14). Given that our focus is on music, but soundtracks consist of speech, music and sound effects, our question was formulated as follows: "The following are a series of perceptual attributes of music. You are asked to evaluate the *music in the background* in terms of the adjectives on each side of the scale."

To annotate the perceived affect of videos, we drew from the aesthetic emotions scale [46, AESTHEMOS], which was devised from an extensive review of emotion measures from different domains such as music, literature, film, painting, advertisements, design, and architecture, and is thus ideal, in its flexibility, for our use with multimodal stimuli. Given that our focus is on music and sound, in a preliminary study we limited our choice to a subset of 10 AESTHEMOS items that intersect with the 13 music emotions listed by Cowen et al. [47]. Of these, we kept only seven scales which showed significant discriminant capabilities: Happy or Delightful, Amusing or Funny, Beauty or Liking, Calm or Relaxing, Energising or Invigorating, Angry or Aggressive, and Triumphant or Awe-inspiring. We used a single unipolar item for each subscale, instead of two. We collected a total of 4530 ratings from 151 participants from the UK (76 M, 75 F, aged 39 ± 13). Given that our aim is to analyse the intended emotional profile, our question was formulated as follows:

Embeddings	Voice	Target F1	Secon. F1	Avg. R^2 emo	Avg. r emo	Avg. R^2 mid	Avg. r mid
mfcc	no	.79 ± .08	.66 ± .07	.02 ± .17	.36 ± .11	.14 ± .16	.48 ± .09
mfcc	yes	.78 ± .10	.65 ± .07	.06 ± .16	.38 ± .11	.13 ± .15	.43 ± .10
msd	no	.87 ± .05	.66 ± .06	.25 ± .11	.54 ± .08	.35 ± .14	.62 ± .09
msd	yes	.95 ± .04	.79 ± .05	.26 ± .15	.56 ± .09	.30 ± .12	.58 ± .09
openl3_env	no	.91 ± .05	.72 ± .06	.34 ± .10	.61 ± .08	.41 ± .10	.66 ± .07
openl3_env	yes	.95 ± .04	.77 ± .05	.34 ± .13	.62 ± .08	.35 ± .12	.62 ± .08
openl3_music	no	.87 ± .09	.71 ± .06	.31 ± .16	.56 ± .19	.39 ± .16	.64 ± .15
openl3_music	yes	.91 ± .11	.76 ± .10	.29 ± .17	.56 ± .19	.31 ± .14	.59 ± .13

Table 2. Mean and standard deviation from 5x repeated 5-fold cross-validation. 'Target' refers to the gender orientation of ads (binary); secondary tasks involve voice-related content analytical variables. 'No' represents models trained on voice-separated accompaniments, while 'Yes' indicates models trained on entire soundtracks.

"Toys commercials are targeted at an audience mainly consisting of children and aim at evoking the following emotions. Pay attention to both sound and images and rate each *intended* emotion accordingly."

2.4 Between-targets ANOVA

We first performed between-targets (i.e., gender targets of the commercials) one-way analyses of variance for each of the music-focused and emotion scales. When ANOVA assumptions were violated, we performed a Kruskal-Wallis H-test instead. Highly significant polarisation ($p < .001$) emerged for twelve of the mid-level music perceptual scales, with stark contrasts observed between feminine and masculine-targeted commercials, and commercials targeted at mixed audiences generally registering in-between values. Masculine adverts were more Electric than Acoustic, more distorted, disharmonious and with a less clear melodic contour than feminine ones. They also were more dense in terms of instrumentation, more Punchy, with stronger beats, and therefore were generally louder and heavier. Also in terms of rhythmic complexity, they were more complex than feminine-targeted commercials. Thus a clear picture emerges, as the soundtracks in boys' adverts are significantly more *abrasive* than those in girls' ads.

Similarly, stark contrasts ($p < .001$) were observed between feminine and masculine-targeted commercials for all affective scales, with commercials targeted at mixed audiences often registering in-between values. Commercials targeted at boys were the least "Happy or delightful", the least "Amusing or funny", "Calm or relaxing", and registered the lowest values on the scale "Beauty or liking". They instead were the most "Energising or invigorating", "Angry or aggressive", and "Triumphant or awe-inspiring". Apart from the scale "Amusing or funny", which scored the highest values within mixed audiences commercials, adverts targeted at girls displayed an opposite behaviour to those for boys. For example, they were the most "Calm or relaxing" and the least "Angry or aggressive". As previously highlighted with the music-focused scales, masculine-targeted commercials appear again to be significantly more abrasive than the feminine ones.

We report a more in-depth analysis in an upcoming publication. In this paper, we exclude "Amusing or funny" from further analyses due to poor correlation with the mid-level features. We also exclude the three non-significant mid-level scales: Wide/Narrow pitch variation, Repetitive/Non-repetitive, and Fast tempo/Slow tempo.

3. MACHINE LEARNING PIPELINE

Our machine learning framework is a multi-task learning model implemented in PyTorch (Fig. 1). It was trained to simultaneously learn mid-level features regression, emotion regression, and all the CA variables (classes). These tasks share an initial hidden layer with 128 units and then branch out into separate sub-tasks. Each sub-task has its own hidden layer with 128 units and an output layer with dimensions corresponding to the specific task.

To avoid the jingle of the retailer in the last 5 seconds of most soundtracks, we trimmed them accordingly. Then with Spleeter [48] we separated voices and accompaniments. Features were extracted in non-overlapping chunks across the trimmed soundtrack and then averaged across the chunks. We computed 20-band MFCCs using `librosa` [49], along with their delta and delta-deltas, yielding 60-dimensional embeddings. A reimplementation of a state-of-the-art model trained on the million song dataset (MSD) [50] provided 256-dimensional embeddings. OpenL3 features were computed using the provided `conda` package [51], generating 512-dimensional embeddings for both environmental and music models.

The proposed model employs an equally weighted, combined loss function, incorporating the mean squared error for the mid-level features and emotion regression tasks, and cross-entropy loss for the classification tasks. The model was trained jointly on all tasks. We also used a model checkpoint and early stopping with a patience of 30 epochs (maximum of 200). Repeated 5-fold cross-validation was performed (10% test, 10% validation, for 5 repetitions, i.e. 25 "folds", as the random seed was not set) and utilised the AdamW optimizer instead of Adam for regularization. Further optimising the network to surpass the already remarkable results, as well as conducting ablation studies to evaluate the various components and design choices, is beyond the scope of our investigation.

Embeddings	Voice	Target F1	Secon. F1	Avg. R^2 emo	Avg. r emo	Avg. R^2 mid	Avg. r mid
mfcc	no	.52 ± .05	.67 ± .06	.04 ± .16	.37 ± .11	.14 ± .14	.48 ± .09
mfcc	yes	.48 ± .04	.67 ± .05	.05 ± .15	.38 ± .11	.15 ± .14	.46 ± .09
msd	no	.62 ± .05	.67 ± .06	.23 ± .15	.54 ± .10	.36 ± .12	.64 ± .07
msd	yes	.67 ± .06	.80 ± .05	.29 ± .12	.57 ± .08	.33 ± .10	.60 ± .07
openl3_env	no	.59 ± .06	.72 ± .06	.30 ± .12	.59 ± .08	.42 ± .10	.66 ± .07
openl3_env	yes	.66 ± .07	.77 ± .06	.34 ± .11	.61 ± .07	.35 ± .10	.62 ± .07
openl3_music	no	.64 ± .07	.73 ± .06	.32 ± .12	.60 ± .08	.43 ± .11	.68 ± .07
openl3_music	yes	.67 ± .04	.78 ± .05	.35 ± .12	.61 ± .08	.37 ± .10	.63 ± .07

Table 3. Same as Table 2, but results refer to ternary 'Target' classification.

4. RESULTS

Tables 2 and 3 reveal once again stark differences between the soundtracks of commercials designed for feminine and masculine audiences (the value "no" corresponds to models trained on the voice-separated accompaniments). In fact, the binary classification task on the soundtracks including voice achieves an impressively high Target F1 score of .95 ± .04 using the MSD and OpenL3 env embeddings. It is also worth noting that even without voice, the soundtracks still contain enough information to classify the commercials with a high degree of accuracy, with the OpenL3 env embedding achieving a Target F1 score of .91 ± .11. In a way, the dataset is so gendered that it can be considered a toy dataset in all senses.

Upon closer examination of the R^2 and r emotions metrics, we observe that they are relatively low across all experiments compared to mid-level metrics. This contrasts with previous research [36] where mid-level correlations were lower than those of emotions, as in our case the R^2 mid-level and r mid-level metrics are generally higher, with the OpenL3 embeddings performing the best.

When comparing Tables 2 and 3, the high performance of the MSD and both OpenL3 embeddings on the binary task, suggests that there are no significant differences in the soundtracks of mixed-audience commercials compared to those targeted at feminine or masculine audiences. This confirms the results from the analysis of variance and highlights the ability of these embeddings to perform similarly across different target audiences. Overall, we found that the OpenL3 embeddings performed better than others across all tasks, indicating superior generalizability, as already shown in previous research especially in the context of limited training examples [52]. However, the relatively low R^2 and r for emotions suggest that there is still room for improvement, possibly through multimodal fusion.

It is noteworthy that human voice plays a critical role in conveying higher-level connotations, as performance on the classification tasks *and* especially the emotion regressions generally improves when voices are present. Additionally, improvement in mid-level regressions on the accompaniments of the soundtracks (no voice) indicates that participants in the data collection were able to focus on the background of the soundtracks, as they were asked to.

Although the MFCCs are the worst performing, their discriminative power on the target task and the decent performance on the mid-level features regression highlight the underlying "simplicity" of the task, in terms of the strong collinearity due to the degree of gender-polarization inherent in the dataset.

5. CONCLUSION

By examining the performance of different musical embeddings in classifying commercials targeted at different audiences, and by providing explainable inference of the target of the commercials, in terms of affective and of music perceptual features, this study sheds light on the role of music in gendered marketing strategies. Such approach has significant implications for advertisers, policymakers, and broadcasters, who recently faced a public backlash against the gendered marketing of toys and other products. [4] Furthermore, the study highlights the importance of considering the role of music when regulating marketing strategies and developing more inclusive and diverse advertising campaigns. Our results suggest that gendered music styles in toy commercials emerge as a result of deliberate marketing strategies, as such styles reflect gender stereotypes that are "ludicrously old-fashioned and offensively out of touch" [53] and still prevalent in the industry.

By bringing together music analysis, machine learning, and critical analysis, this study illustrates the potential of interdisciplinary approaches, contributing to the emerging field of computational social studies. It highlights the importance of considering the role of music, among other modalities, in shaping societal norms and values and the need for greater awareness and accountability in the use of such affordances in marketing and other industries.

Future research can build on these findings by further investigating the relationship between gendered music and advertising strategies in different industries and contexts, exploring the impact of gendered music on consumer behavior and societal perceptions of gender, and developing new methodologies for creating more inclusive and diverse marketing campaigns. The results also emphasise the potential for the development of multimodal approaches to enhance the models' performance on these tasks.

[4] https://www.bbc.co.uk/news/world-us-canada-46613032

6. ACKNOWLEDGMENTS

This work was supported by UK Research and Innovation [grant number EP/ S022694/1], and was partially conducted during the first author's Enrichment Scheme placement at the Alan Turing Insitute. The authors would like to express their gratitude to Professor Petra Lucht for her invaluable guidance at early stages of this study.

7. REFERENCES

[1] J. J. Kellaris and S. P. Mantel, "The influence of mood and gender on consumers' time perceptions," *ACR North American Advances*, 1994.

[2] G. T. Toney and J. B. Weaver, "Effects of gender and gender role self-perceptions on affective reactions to rock music videos," *Sex Roles*, 1994.

[3] J. Meyers-Levy and R. Zhu, "Gender differences in the meanings consumers infer from music and other aesthetic stimuli," *Journal of Consumer Psychology*, 2010.

[4] G. Rippon, "Do women and men have different brains?" *New Scientist*, 2019.

[5] S. L. Bem, "Gender schema theory: A cognitive account of sex typing." *Psychological review*, 1981.

[6] E. Leung, "Gender schemas," in *Encyclopedia of Personality and Individual Differences*. Springer, 2020.

[7] M. G. Boltz, "Musical soundtracks as a schematic influence on the cognitive processing of filmed events," *Music Perception*, 2001.

[8] M. Shevy, "Music genre as cognitive schema: Extramusical associations with country and hip-hop music," *Psychology of music*, 2008.

[9] S. Kristen and M. Shevy, "A comparison of german and american listeners' extra musical associations with popular music genres," *Psychology of Music*, 2013.

[10] P. Stockwell, *Cognitive poetics: An introduction*, 2nd ed. Routledge, 2019.

[11] L. Betti, C. Abrate, and A. Kaltenbrunner, "Large scale analysis of gender bias and sexism in song lyrics," *arXiv preprint arXiv:2208.02052*, 2022.

[12] N. Dibben, "Gender identity and music," in *Musical identities*. New York: Oxford University Press, 2002.

[13] C. A. Elliot and M. Yoder-White, "Masculine/feminine associations for instrumental timbres among children seven, eight, and nine years of age," *Contributions to Music Education*, 1997.

[14] L. M. Stronsick, S. E. Tuft, S. Incera, and C. T. McLennan, "Masculine harps and feminine horns: Timbre and pitch level influence gender ratings of musical instruments," *Psychology of Music*, 2018.

[15] D. C. Sergeant and E. Himonides, "Gender and the performance of music," *Frontiers in psychology*, 2014.

[16] ——, "Gender and music composition: A study of music, and the gendering of meanings," *Frontiers in psychology*, 2016.

[17] P. Tagg, "An anthropology of stereotypes in tv music?" *Swedish Musicological Journal*, vol. 71, 1989.

[18] P. Tagg and B. Clarida, "Title tune gender and ideology," in *Ten little title tunes: towards a musicology of the mass media*. Huddersfield: The Mass Media Music Scholars' Press, NY., 2003.

[19] Y. Wang and E.-Á. Horvát, "Gender differences in the global music industry: Evidence from musicbrainz and the echo nest," in *Proceedings of the International AAAI Conference on Web and Social Media*, 2019.

[20] R. Lakoff, "Language and woman's place," *Language in society*, 1973.

[21] H.-B. Brosius and H. M. Kepplinger, "Der einfluß von musik auf die wahrnehmung und interpretation einer symbolisierten filmhandlung," *Rundfunk und Fernsehen*, 1991.

[22] A.-K. Herget, "On music's potential to convey meaning in film: A systematic review of empirical evidence," *Psychology of Music*, 2021.

[23] R. L. Welch *et al.*, "Subtle sex-role cues in children's commercials." *Journal of Communication*, 1979.

[24] J. Lewin-Jones and B. Mitra, "Gender roles in television commercials and primary school children in the uk," *Journal of children and media*, 2009.

[25] B. Mitra and J. Lewin-Jones, "Colin won't drink out of a pink cup," in *The handbook of gender, sex, and media*. Wiley Online Library, 2012.

[26] F. Johnson and K. Young, "Gendered voices in children's television advertising," *Critical Studies in Media Communication*, 2002.

[27] D. Schiffrin, D. Tannen, and H. E. Hamilton, "Introduction to the first edition," *The handbook of discourse analysis*, 2015.

[28] Y. Feng, H. Chen, and L. He, "Consumer responses to femvertising: a data-mining case of dove's "campaign for real beauty" on youtube," *Journal of Advertising*, 2019.

[29] G. Wiedemann, "Text mining for discourse analysis: An exemplary study of the debate on minimum wages in germany," *Quantifying approaches to discourse for social scientists*, 2019.

[30] A. Miech, D. Zhukov, J.-B. Alayrac, M. Tapaswi, I. Laptev, and J. Sivic, "Howto100m: Learning a text-video embedding by watching hundred million narrated video clips," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

[31] S. Islam, A. Dash, A. Seum, A. H. Raj, T. Hossain, and F. M. Shah, "Exploring video captioning techniques: A comprehensive survey on deep learning methods," *SN Computer Science*, 2021.

[32] Y. Dinkov, A. Ali, I. Koychev, and P. Nakov, "Predicting the leading political ideology of youtube channels using acoustic, textual, and metadata information," *arXiv preprint arXiv:1910.08948*, 2019.

[33] K. Ye, N. H. Nazari, J. Hahn, Z. Hussain, M. Zhang, and A. Kovashka, "Interpreting the rhetoric of visual advertisements," *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[34] 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.

[35] H.-H. Wu, C.-C. Kao, Q. Tang, M. Sun, B. McFee, J. P. Bello, and C. Wang, "Multi-task self-supervised pre-training for music classification," in *ICASSP 2021-2021*. IEEE, 2021.

[36] S. Chowdhury, A. Vall, V. Haunschmid, and G. Widmer, "Towards explainable music emotion recognition: The route via mid-level features," in *Proceedings of the 20th ISMIR Conference, Delft, The Netherlands*, 2019.

[37] A. Aljanaki and M. Soleymani, "A data-driven approach to mid-level perceptual musical feature modeling," in *Proceedings of the 19th ISMIR Conference, Paris, France*, 2018.

[38] P. G. Hunter, E. G. Schellenberg, and S. M. Stalinski, "Liking and identifying emotionally expressive music: Age and gender differences," *Journal of Experimental Child Psychology*, 2011.

[39] M. S. Larson, "Interactions, activities and gender in children's television commercials: A content analysis," *Journal of Broadcasting & Electronic Media*, 2001.

[40] S. G. Kahlenberg and M. M. Hein, "Progression on nickelodeon? gender-role stereotypes in toy commercials," *Sex roles*, 2010.

[41] E. Goffman, *Gender advertisements*. New York: Harper Colophon Books, 1976.

[42] M. E. Verna, "The female image in children's tv commercials," *Journal of Broadcasting & Electronic Media*, vol. 19, no. 3, 1975.

[43] K. A. Neuendorf, "Content analysis—a methodological primer for gender research," *Sex roles*, 2011.

[44] V. Alluri and P. Toiviainen, "Exploring perceptual and acoustical correlates of polyphonic timbre," *Music Perception*, 2010.

[45] K. L. Whiteford, K. B. Schloss, N. E. Helwig, and S. E. Palmer, "Color, music, and emotion: Bach to the blues," *i-Perception*, 2018.

[46] I. Schindler, G. Hosoya, W. Menninghaus, U. Beermann, V. Wagner, M. Eid, and K. R. Scherer, "Measuring aesthetic emotions: A review of the literature and a new assessment tool," *PloS one*, 2017.

[47] 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*, 2020.

[48] 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.

[49] 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*, 2015.

[50] M. Won, S. Chun, and X. Serra, "Toward interpretable music tagging with self-attention," *arXiv preprint arXiv:1906.04972*, 2019.

[51] J. 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, 2019.

[52] S. Grollmisch, E. Cano, C. Kehling, and M. Taenzer, "Analyzing the potential of pre-trained embeddings for audio classification tasks," in *2020 28th European Signal Processing Conference (EUSIPCO)*. IEEE, 2021.

[53] C. Fine and E. Rush, ""Why does all the girls have to buy pink stuff?" The ethics and science of the gendered toy marketing debate," *Journal of Business Ethics*, 2018.

A DATASET AND BASELINES FOR MEASURING AND PREDICTING THE MUSIC PIECE MEMORABILITY

Li-Yang Tseng **Tzu-Ling Lin** **Hong-Han Shuai**
Jen-Wei Huang **Wen-Whei Chang**
National Yang Ming Chiao Tung University, Hsinchu, Taiwan
{liyangtseng.ee10, tzulinglin.11 ,hhshuai, admsd.ee10, wwchang}@nycu.edu.tw

ABSTRACT

Nowadays, humans are constantly exposed to music, whether through voluntary streaming services or incidental encounters during commercial breaks. Despite the abundance of music, certain pieces remain more memorable and often gain greater popularity. Inspired by this phenomenon, we focus on measuring and predicting music memorability. To achieve this, we collect a new music piece dataset with reliable memorability labels using a novel interactive experimental procedure. We then train baselines to predict and analyze music memorability, leveraging both interpretable features and audio mel-spectrograms as inputs. To the best of our knowledge, we are the first to explore music memorability using data-driven deep learning-based methods. Through a series of experiments and ablation studies, we demonstrate that while there is room for improvement, predicting music memorability with limited data is possible. Certain intrinsic elements, such as higher valence, arousal, and faster tempo, contribute to memorable music. As prediction techniques continue to evolve, real-life applications like music recommendation systems and music style transfer will undoubtedly benefit from this new area of research.

1. INTRODUCTION

Music memorability is essential and has a wide range of commercial applications. For instance, content creators and marketing teams can use unique visual aids or audio components to captivate target audiences and distinguish themselves from other information sources [1, 2]. Sound logos, such as Netflix's iconic "ta-dum," are designed to engage listeners and promote brand recognition. In the realm of cognition literature, numerous studies have sought to understand the factors that contribute to music memorability [3–6]. For instance, [5, 6] bridged the gap between cognitive science and MIR by examining whether implicit or explicit memory for a single tune is impacted by the type of encoding task and variations in timbre, tempo and structure.

However, music memorability remains a relatively unexplored area, particularly from a data-driven standpoint. Research related to music memorability includes the study of involuntary musical imagery (INMI) [7, 8], also known as "earworms," which refers to the phenomenon where fragments of music become mentally lodged on repeat. For instance, Jakubowski et al. proposed a model that can determine whether a piece of music may induce the INMI effect by using statistical analysis and a random forest model [8]. However, the mechanism of INMI differs from music memorability since the former is a passive process while the latter can be active, e.g., everyone remembers how to sing "Happy Birthday," but the song may not qualify as an earworm. Another line of prior studies [9–12] investigating the intrinsic memorability of multimedia content have predominantly focused on computer vision, with their findings suggesting that data-driven approaches can effectively determine memorability levels. Motivated by these studies, we break new ground in exploring music memorability from a data-driven perspective by compiling a novel dataset and employing machine learning techniques.

Specifically, to expand the scope of memorability detection and recognition in music information retrieval (MIR), we establish a new research domain called music memorability regression (MMR), which aims to predict a memorability score for a given music piece. We create an experimental procedure as shown in Figure 1 to collect a new dataset, the YouTube Music Memorability (YTMM) dataset, where memorability scores are determined by the percentage of participants who can recall the music piece after a certain period. This dataset provides reliable and consistent music memorability scores across all participants, paving the way for further research in the field. We also propose several baseline approaches for predicting music memorability, including feature engineering using hand-crafted music-related features and transfer learning techniques. These baselines not only demonstrate the potential of machine learning in addressing music memorability but also serve as a foundation for future work.

Despite the promise of machine learning in tackling music memorability by predicting memorability scores, its "black box" characteristics hinder the interpretation of machine decisions in MIR tasks. A straightforward approach would be to compute correlations without relying on black-

© L.-Y. Tseng, T.-L. Lin, H.-H. Shuai, J.-W. Huang, and W.-W. Chang. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** L.-Y. Tseng, T.-L. Lin, H.-H. Shuai, J.-W. Huang, and W.-W. Chang, "A dataset and Baselines for Measuring and Predicting the Music Piece Memorability", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

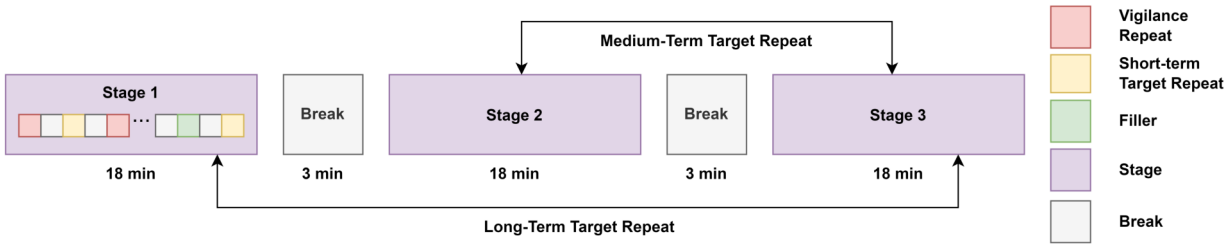


Figure 1. The music memory game, which allows data annotators to label music memorability scores reliably. The experiment is divided into three stages, each with a 3-minute long break in between. Each 18-minute stage is composed of multiple 5-second music pieces and short breaks.

box prediction models to glean insights about the relationship between memorability and musical features. However, given the complexity of analyzing music memorability, using a single feature results in an extremely low correlation with memorability, leading to inconclusive findings. One alternative would be to explore all possible feature combinations when calculating correlations, but the sheer number of combinations, e.g., $2^{20} - 1$ for just 20 features, renders this approach impractical. A/B testing could be used to determine which type of music is more memorable, but it suffers from similar drawbacks, such as being time-consuming and unable to account for all variables that may impact the experiment's outcome. To make machine learning models reveal their "black box" characteristics, researchers are increasingly adopting explainable artificial intelligence (XAI) [13] for deeper insights. Building on previous interpretability analyses in audio processing [14, 15], we utilize Shapley Additive Explanations (SHAP) [16], a game-theoretic approach that clarifies the output of machine learning models, to identify the key components of memorable music.

Our main contributions are as follows: first, we present the new YTMM dataset with objective annotations of memorability scores, which will be publicly available for future research; second, we propose several deep learning baseline models for MMR; and finally, we explore the potential characteristics of memorable music pieces while providing interpretability for these deep learning-based methods.

2. RELATED WORK

In addition to the cognition literature on music memorability [5, 6], there are several related yet distinct terms, such as Involuntary Musical Imagery (INMI) or "earworms"—fragments of music that involuntarily come to mind [7]. Studies have examined earworms through interviews, environmental and psychological conditions leading to INMI, and the impact of melodic features and song popularity on spontaneous musical imagination [8]. Crucial differences between INMI and music memorability include: 1) INMI involves uncontrollable mental repetition, while memorability requires conscious recall; and 2) the stimuli in [8] are highly familiar to participants, whereas our study selects audios unfamiliar to most annotators to mitigate the influence of individual listening histories on memorability. Another related concept is hook catchiness [17–20], which refers to the most easily recalled fragment of a musical piece. However, our focus lies in predicting the memorability of different music pieces rather than assessing the impact of various segments within the same tune on catchiness prediction and recognition. Furthermore, we ensure our stimuli consist solely of pure instrumental music clips to prevent any textual information from lyrics influencing music memorability.

Moreover, while deep learning has achieved significant success in supervised MIR tasks, it often demands large-scale annotated data. However, collecting useful annotations for MIR tasks can be costly, as it typically requires expertise and domain knowledge [21]. To tackle this challenge, various data augmentation and training strategies have been proposed [21–24]. For instance, McFee et al. [21] apply transformations such as pitch shifting, time-stretching, and adding background noise to the original waveform. Cubuk et al. [22] mask both time and frequency content to expand the input space in automatic speech recognition (ASR) and MIR tasks. To enhance learning robustness with limited data, Wu et al. [23] extract general music representations using a multi-task pre-trained encoder, inspired by speech processing research [25, 26]. Similarly, Castellon et al. [24] employ transfer learning from existing music generation architectures. However, not all the aforementioned methods are simultaneously open-source, computationally inexpensive, and interpretable. Therefore, in this paper, we focus on applying signal processing approaches like masking, with further details provided in Section 4.

3. DATASET CONSTRUCTION FOR MUSIC MEMORABILITY

In this section, we discuss the details of our dataset collection process and how music memorability is measured.

3.1 Audio Collection

To construct a dataset with objective music memorability scores, we first ensure that the audio samples are unbiased. We randomly select music by querying music-related videos using the YouTube API with random query keys, avoiding any specific music genre preference. Addi-

Task Type	# of Audios	# of Repetition (min)	# of Repetition (max)	# of Repetition (avg)	# of Repetition (std)
Filler	65	-	-	-	-
Vigilance	21	5	10	6.5	1.08
Short-Term Target	88	10	49	25.23	10.81
Medium-Term Target	41	61	131	110.33	16.32
Long-Term Target	20	155	276	222.05	36.64

Table 1. Details of different audio tasks in the music memory game.

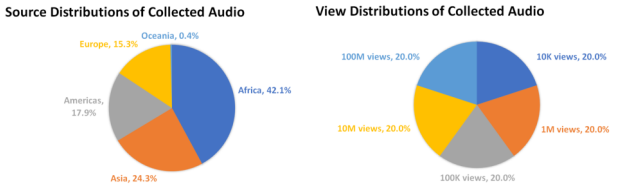


Figure 2. Distributions of the audio published location and the distributions of the audio views in the final dataset.

tionally, we manually filter the music to confirm that the queried videos contain pure music content, excluding instrument tutorials or gadget unboxings. Next, we conduct a pilot study to verify that the selected audios are unfamiliar to most of the annotators in our target user group. Considering the annotators' nationalities might not be as varied as the music collection's, and language can be a memorable yet non-music-related element, we only use the intro part of each song. This approach helps eliminate other potential variables affecting music memorability. Also, the volume across all audio clips is normalized to minimize any memorable attributes unrelated to the music itself. Loudness normalization ensures the music is remembered based on its inherent qualities rather than its loudness.

We use only a segment of each audio for two reasons: i) to better eliminate confounding factors, such as vocal timbre, and ii) to shorten the period of annotations and prevent fatigue. We achieve this by truncating audios into structurally meaningful segments and applying proper time-stretching to alter the duration of an audio signal to a fixed length without distorting the audio. The segmentation process is supervised by an expert with a professional music education background. Note that time-stretching not only reduces modeling complexity but also prevents annotators from memorizing the audio based on its duration.

Ultimately, we collect 235 structurally meaningful 5-second audios with labeled music memorability scores. Our goal is to determine which types of music pieces are more likely to be memorized, rather than focusing on entire music clips, which are more complex and involve additional factors. This research can facilitate various applications, such as Netflix's iconic "ta-dum" sound. The collected data can be found in the supplementary materials. Figure 2 illustrates the distribution of the collected audios concerning their published geographical locations and total views on YouTube, with view counts ranging from 10K to 100M.

3.2 The Music Memory Game

To annotate the memorability of the collected musical data, we follow the setting of image memory game [9] to design a novel music listening experiment. During the experiment, the recruited data annotators are asked to listen to 235 music pieces in total and answer whether the audio is repeated in the experiment or not. From a cognitive view, we define music memorability as long-term musical salience and the extent to which a musical piece continues to be remembered over time. In the music memory game, music memorability is measured as the tendency to correctly recognize a music piece when encountering it again in the experiment among all users. Specifically, let $x_j^{(i)}$ denote whether the i-th music piece can be recalled by the j-th data annotator, *i.e.*, 1 if the annotator recognized the i-th music piece. The memorability score of music i, denoted by $m^{(i)}$, is then calculated by:

$$m^{(i)} = \frac{1}{n_i} \sum_{j=1}^{n_i} x_j^{(i)}, \; x_j^{(i)} \in \{0, 1\} \tag{1}$$

where n_i is the total number of data annotators for the i-th music.

To make the ground truth unbeknownst to all participants, music excerpts are split into three task categories: "vigilance", "target", and "filler". Targets and vigilance targets are both repeated in the experiment, while the former are collected to be the true labels and the latter is used to make sure participants are attentive when labeling data. Moreover, fillers are used to stuff the spacing between the first and second repetition of a target and therefore is only presented once. The overview of the music memory game experimenting procedure is shown in Figure 1. The target-vigilance-filler split details can be found in Table 1. Rigorous criteria are enforced to monitor the performances of data annotators and preserve the quality of memorability labels. Specifically, annotations from users who detect vigilance repetition with an accuracy lower than 60% are automatically discarded. Furthermore, to prevent gathering biased memorability, all annotators only engage in labeling once. We recruited a total of 218 users from campus, with 163 clearing the vigilance accuracy level, 17% of passed annotators having professional music education backgrounds, and over 98% being between the ages of 20 and 29.

Differing from previous works on image memorability, our experiment is composed of three similar stages with breaks inserted in between. The reasons for using stages and breaks are two-fold. First, audios are sequential, there-

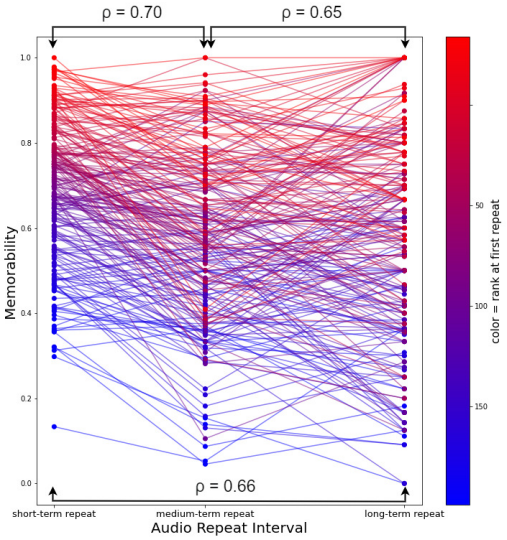


Figure 3. Memorability scores at various stages. The color symbolizes the rank of short-term memorability, while the lines represent stage relationships. The plot also shows Spearman's rank correlations ρ between memorabilities measured at each stage.

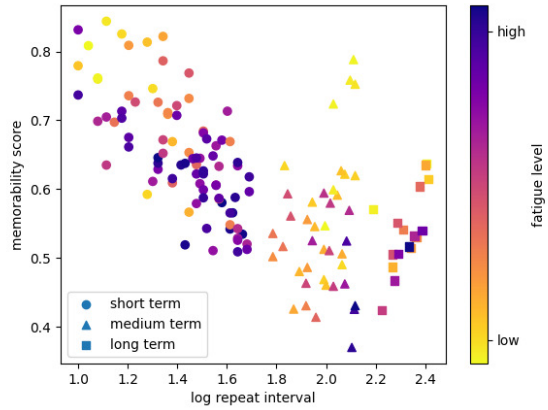


Figure 4. Relations between memorability score and target repeat interval in log scale. The hue represents the level of fatigue.

fore it is more exhausting to label the memorability score to audios as compared to static images. Second, it usually takes some time for the earworm phenomenon to happen when listening to music. Hence, we assume memorability should be invariant even after encountering breaks that probably would reset the memory. The results of relations between repeat interval and memorability score are shown in Figure 3, where the lines exhibiting memorability scores across short-term, medium-term, and long-term repeats. The results manifest that the memorability score is indeed independent of the sequential context. Therefore, it is easier to memorize truly memorable pieces of music even after long breaks. The fact that Spearman's rank correlation [27] between short-term, medium-term, and long-term are all greater than 0.64 also proves that the rank of memorability score is preserved across variant repeat intervals.

3.3 Labels and Consistency Analysis

To assure that collected labels are universal across all data annotators, we evaluate the human consistency according to previous work [9] by randomly splitting all participants into 2 groups and examining how well the memorability scores measured in the first groups matched the ones measured in the second group by averaging Spearman's rank correlation [27] between randomly separated two halves of the participants 25 times. The average Spearman's rank correlation coefficient ρ is 0.83, indicating the consistency of the collected data.

Figure 4 shows the scattering plot of music memorability and repeat interval. The graph demonstrates that music possesses a linear relation between memorability score and log-scaled repeat interval. Please note that the fatigue level is another factor in the plot that also contributes to the memorability score of audio. The fatigue level, defined as the amount of audios listened without a 3-minute break, is a direct result caused by staging experiment and participating in taking a break in the middle since listening to more music at one time without resting reduces participants' ability to identify repeated music pieces. The setting of inserting audio to random positions in the experiment procedure adds more context diversity to the process of memorizing music, thus making the labeled memorability scores more robust.

4. MUSIC MEMORABILITY PREDICTION

4.1 Learning with Handcrafted Features

Although feature extractions for deep learning models can be data-driven without being handcrafted, leading to a better result given sufficient training data, handcrafted features provide interpretable information for more insights. Therefore, we propose handcrafted features that can more accurately depict the low-level acoustic features or high-level semantic features of musical clips as shown in Table 2. For the low-level acoustic features that can be directly derived from the audio signal of music segments, we leverage the harmony, rhythm and timbre since they are most easily recognizable fragments of a piece of music [17] and describe the fundamental elements of a tune. Moreover, zero crossings and zero crossing rate are also extracted since they give the impressions into the frequency content of a signal. On the other hand, high-level semantic features are more abstract descriptions. Since the previous works in Psychology [28,29] mention the link between music emotion and memory, we introduce valence and arousal, which represent the mood of music pieces as features.

Another high-level feature is genre, which describes

Level	Category	Feature Implementation
Low-level	Harmony	mean, std of 12 pitch class
	Rhythm	beat per minute (bpm)
	Timbre	mean, std of 4-tracks (Vocals, Bass, Drums, Others)
	Zero Crossing	# of zero crossings & avg, median of zero crossings rate
High-level	Mood	valence, arousal
	Genre	Music, Musical Instrument

Table 2. Explainable handcraft features.

how likely a clip is belong to a certain type of music. Specifically, due to the unstable performance of existing algorithms for detecting sequences of chord labels, we employ chromagram (chroma) [30] as a representation of harmony patterns. To extract timbre information, the Mel-Frequency Cepstral Coefficient (MFCCs) is widely utilized. Although MFCCs is representative for timbre, its components are difficult to grasp intuitively. As a result, we treat MFCCs as a raw feature and find an alternative solution by first separating source audios into four components using source separation software Spleeter [31], and calculating their respective amplitudes to represent the characteristics of different instruments and frequency ranges. For the rhythmic pattern, although Tempogram [32] captures the underlying rhythmic pattern of raw audios, it is unable to provide precise insights to concretely measure the audio's groove. Therefore, we instead utilize beat per minute (bpm) to represent general rhythm characteristics. We also use static valence and arousal values to describe perceived music moods, which are predicted by using Support Vector Regression (SVR) with a linear kernel trained on the PMEmo dataset [33]. For genre features, we use the predicted music tagging and instruments from the downstream task of PANN [34]. Finally, SVR and Multilayer Perceptron (MLP) are employed as predictors to link audio features to memorability scores.

4.2 End-to-End Deep Learning

Deep learning [35] is featured by its ability to directly learn meaningful information from raw data, as opposed to using hand-crafted features. As a result, we also test end-to-end models to find if their feature-learning process improves performance. Our model uses spectrograms in Mel-scale as inputs, similar to previous end-to-end MIR tasks. Moreover, transfer learning [36], which applies previously learned knowledge to new data, has been found to significantly increase learning performance by skipping costly data-labeling procedures. Here, we use the self-supervised pre-trained Audio Spectrogram Transformer (SSAST) [37] since SSAST has been proved to achieve state-of-the-art results on numerous audio tasks, including audio event classification, keyword spotting, mood recognition, and speaker identification, after being trained on a vast amount of unlabeled data.

Method	Corr.	MSE	MSE STD
chroma + MLP	0.1740	0.0326	-
MFCCs + MLP	0.1179	0.0353	-
convnet features [38] + MLP	0.1889	0.0314	-
EHC features + SVR	**0.2988**	0.0339	0.0128
EHC features + SVR + PS	0.2084	0.0391	0.0129
EHC features + MLP	0.2656	0.0263	0.0058
EHC features + MLP + PS	0.2388	0.0275	0.0059
mel-spectrograms + SSAST	0.0124	0.0298	0.0061
mel-spectrograms + SSAST + PS	0.2658	0.0265	0.0074

Table 3. Spearman's rank correlation and MSE loss between predicted and ground truth music memorability score using different models. Note that EHC features stand for explainable handcrafted features, PS stands for pitch shift data augmentation, and Corr. represents Spearman's rank correlation.

5. EXPERIMENT RESULTS

Evaluation Metrics. Spearman's rank correlation and mean squared error (MSE) loss are used as the metrics to evaluate the performance of music memorability prediction. The former indicates the ability to rank the relative memorability of different audios, while the latter indicates the absolute error of the predicted results.

Different Baselines. Here, we leverage Chroma and MFCCs along with their respective derivatives as two hand-crafted feature representations and fit the ground truth by Multilayer Perceptron (MLP) as two simple baselines. Moreover, we also use the convnet model as a baseline since it is the most referenced and available work in general music representation. The convnet model [38] utilizes CNNs for music tagging in the pre-training stage, and the extracted features serve as the representation for downstream tasks. Finally, Self-Supervised Audio Spectrogram Transformer (SSAST) [37] is also used as the baseline, which is a Transformer-based model with more parameters as compared to CNNs. SSAST pretrains the model with joint discriminative and generative masked spectrogram patch modeling.

Implementation Details. All the feature classifiers are pretrained without finetuning on the self-collected dataset. To handle the instability stemming from the limited labeled data, we normalize labels by subtracting the mean value, *i.e.*, predicting a relative value instead of an absolute value. For MLP and SSAST models, the learning rates are respectively set to 5e-5 and 5e-6 with the Adam optimizer [39]. We also conduct additional feature selection on the handcrafted features to improve the convergence of the MLP/SVR model (only select 25 features) due the small number of data samples. In addition, techniques including frequency masking, band stop filtering, and reverberation [26] are used to augment data, together with the pitch shifting augmentation. The results are reported by the average of the 10-fold outputs.

5.1 Prediction Results.

Table 3 compares the results of different prediction models, where SVR and MLP take explainable handcrafted

Model	Top-k feature selection	Corr.	MSE
MLP	k = 40 (no feature selection)	0.2160	0.0272
MLP	k = 35	0.2324	0.0270
MLP	k = 25	**0.2656**	**0.0263**
MLP	k = 20	0.2018	0.0271
SVR	k = 40 (no feature selection)	0.2168	**0.0324**
SVR	k = 35	0.2291	0.0340
SVR	k = 25	**0.2988**	0.0339
SVR	k = 20	0.2630	0.0354

Table 4. Spearman's rank correlation and MSE loss for MLP/SVR models with different top-k feature selection.

features as inputs, and SSAST takes mel-spectrograms as inputs. The results indicate that chroma and MFCCs produce the worst results due to the ineffective feature extraction. For convnet features, the performance is better than chroma and MFCCs due to the pretraining. However, the amount of training data is too small to finetune the model on the music memorability regression task. SSAST outperforms other baselines since it incorporates the prior knowledge of spectrograms pre-trained by using advanced methods. Finally, Explainable Handcrafted Features (EHC) method produces the best correlation results by combining both low- and high-level features that help improve music memorability. These quantitative findings manifest that data-driven MIR tasks are notably reliant on huge data quantities to be resilient and general.

5.2 Ablation Study

Table 4 shows an ablation study on feature selection for handcrafted features, indicating that selecting top-25 features leads to the best overall correlation results. Moreover, Table 3 also shows an ablation study on extra pitch shifting for data augmentations. Small pitch shifts (less than 5 semitones) make the altered audio seem natural to the human ear according to [40]. Therefore, we add semitone shifts of -5 to 5 to our data. The results manifest that pitch shifting is effective for the models that take sequence information into account because applying mean pooling across time on harmony features in non-sequential models like SVR and MLP just forces the model to forecast the same value using multiple static chroma information. This may confuse the model on harmony characteristics. On the other hand, models with sequential information, such as SSAST, learn pitch-invariance after pitch shifting. The performance of SSAST notably decreases without pitch shift data augmentation, possibly due to its data-hungry nature as a Transformer-based model, *i.e.* requiring more data for optimal parameter tuning.

5.3 Interpretability

We attempt to gain insight into the intrinsic memory utilizing XAI methodologies. One post-hoc strategy for expressing black box models in a human-interpretable manner is SHAP [16]. Specifically, SHAP explanations are obtained by perturbing a specific instance in the data and observing the impact of these perturbations on the black-box

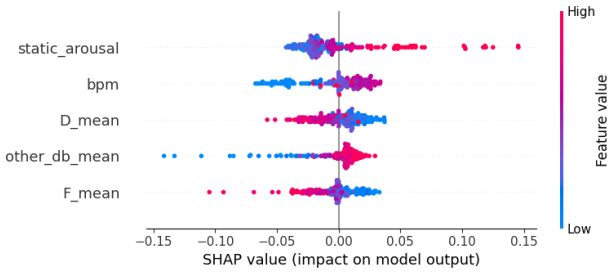


Figure 5. SHAP summary of the SVR model with RBF kernel [41]. The most important features are listed in decreasing order and the fact that feature value rises after the SHAP value shows a positive relationship between the two.

model's output. As such, SHAP allows us to explore the factors that the model considers when determining memorability. Figure 5 visualizes the directionality impact of the top-5 features in SHAP, where the x-axis stands for SHAP value and each point is a SHAP value of a sample for a feature. Red color and blue colors respectively indicate higher and lower values of a feature. As such, we can observe the feature directionality impact based on the distribution. For example, the first row shows that a higher arousal value leads to high memorability scores, while a lower arousal value can lead to both high and low memorability scores. The important factors for the predictor among the EHC features include arousal, bpm, harmony (the feature "D mean") and the timbre features extracted from the source other than vocals, drums, and bass (the feature "other db mean"). According to Psychology research [28], normal individuals without brain damage find it easier to recognize musical excerpts with high arousal. The melodies are the main constituent elements of the source "others" after applying 4-stem Spleeter separation. This finding supports our understanding that we often focus on the main melody in music, and thus the chorus or hook of the song with outstanding melody usually represents the entire song.

6. CONCLUSION AND FUTURE WORK

In this work, we explore the novel task of music memorability regression (MMR) using a data-driven approach. The consistency of our newly proposed YouTube Music Memorability (YTMM) dataset supports our hypothesis that music memorability indeed exists and can be predicted. Furthermore, we investigate the use of feature engineering and self-supervised learning for predicting music memorability, highlighting the importance of prior knowledge and other training approaches, such as label normalization, for improving results with limited data. We make the dataset available online to encourage further research and development in the field of MMR. In the future, we plan to: 1) scale the dataset to better represent the memorability of full music structures, 2) investigate the potential of transfer learning trained on music-oriented datasets to enhance our current baselines, and 3) study the personalization issue since music memorability can be strongly related to the past musical experience of individuals.

7. ACKNOWLEDGEMENT

This work was supported in part by the National Science and Technology Council of Taiwan under Grant NSTC-109-2221-E-009-114-MY3.

8. REFERENCES

[1] M. Alexomanolaki, C. Loveday, and C. Kennett, "Music and memory in advertising: Music as a device of implicit learning and recall," *Music, Sound, and the Moving Image*, vol. 1, no. 1, pp. 51–71, 2007.

[2] S. Hecker, "Music for advertising effect," *Psychology & Marketing*, vol. 1, no. 3-4, pp. 3–8, 1984.

[3] B. Snyder and R. Snyder, *Music and memory: An introduction*. MIT press, 2000.

[4] D. B. Ramsay, I. Ananthabhotla, and J. A. Paradiso, "The intrinsic memorability of everyday sounds," *AES International Conference on Immersive and Interactive Audio*, 2019.

[5] A. R. Halpern and D. Müllensiefen, "Effects of timbre and tempo change on memory for music," *Quarterly Journal of Experimental Psychology*, vol. 61, no. 9, pp. 1371–1384, 2008.

[6] D. Müllensiefen and A. R. Halpern, "The role of features and context in recognition of novel melodies," *Music Perception: An Interdisciplinary Journal*, vol. 31, no. 5, pp. 418–435, 2012.

[7] S. McCullough Campbell and E. H. Margulis, "Catching an earworm through movement," *Journal of New Music Research*, vol. 44, no. 4, pp. 347–358, 2015.

[8] K. Jakubowski, S. Finkel, L. Stewart, and D. Müllensiefen, "Dissecting an earworm: Melodic features and song popularity predict involuntary musical imagery." *Psychology of Aesthetics, Creativity, and the Arts*, vol. 11, no. 2, p. 122, 2017.

[9] P. Isola, J. Xiao, D. Parikh, A. Torralba, and A. Oliva, "What makes a photograph memorable?" *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 7, pp. 1469–1482, 2013.

[10] A. Khosla, A. S. Raju, A. Torralba, and A. Oliva, "Understanding and predicting image memorability at a large scale," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[11] R. Dubey, J. Peterson, A. Khosla, M.-H. Yang, and B. Ghanem, "What makes an object memorable?" in *Proceedings of the ieee international conference on computer vision*, 2015, pp. 1089–1097.

[12] S. Shekhar, D. Singal, H. Singh, M. Kedia, and A. Shetty, "Show and recall: Learning what makes videos memorable," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2017, pp. 2730–2739.

[13] A. Adadi and M. Berrada, "Peeking inside the black-box: a survey on explainable artificial intelligence (xai)," *IEEE access*, vol. 6, pp. 52 138–52 160, 2018.

[14] C.-Y. Li, P.-C. Yuan, and H.-Y. Lee, "What does a network layer hear? analyzing hidden representations of end-to-end asr through speech synthesis," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6434–6438.

[15] J. V. Jeyakumar, J. Noor, Y.-H. Cheng, L. Garcia, and M. Srivastava, "How can i explain this to you? an empirical study of deep neural network explanation methods," *Advances in Neural Information Processing Systems*, vol. 33, pp. 4211–4222, 2020.

[16] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 4765–4774. [Online]. Available: http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf

[17] J. A. Burgoyne, D. Bountouridis, J. van Balen, H. Honing *et al.*, "Hooked: a game for discovering what makes music catchy," in *Proceedings of the 14th Society of Music Information Retrieval Conference (ISMIR)*, 2013.

[18] J. V. Balen, J. A. Burgoyne, D. Bountouridis, D. Müllensiefen, and R. C. Veltkamp, "Corpus analysis tools for computational hook discovery," in *ISMIR*, 2015, pp. 227–233.

[19] J. Van Balen *et al.*, "Audio description and corpus analysis of popular music," Ph.D. dissertation, Utrecht University, 2016.

[20] I. R. Korsmit, J. A. Burgoyne, and H. Honing, "If you wanna be my lover... a hook discovery game to uncover individual differences in long-term musical memory," in *Proceedings of the 25th Anniversary Conference of the European Society for the Cognitive Sciences of Music*, 2017.

[21] B. McFee, E. J. Humphrey, and J. P. Bello, "A software framework for musical data augmentation." in *ISMIR*, vol. 2015, 2015, pp. 248–254.

[22] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, "Randaugment: Practical automated data augmentation with a reduced search space," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 702–703.

[23] H.-H. Wu, C.-C. Kao, Q. Tang, M. Sun, B. McFee, J. P. Bello, and C. Wang, "Multi-task self-supervised pre-training for music classification," in *ICASSP 2021-2021 IEEE International Conference on Acoustics,*

Speech and Signal Processing (ICASSP). IEEE, 2021, pp. 556–560.

[24] R. Castellon, C. Donahue, and P. Liang, "Codified audio language modeling learns useful representations for music information retrieval," in *ISMIR*, 2021.

[25] S. Pascual, M. Ravanelli, J. Serrà, A. Bonafonte, and Y. Bengio, "Learning Problem-Agnostic Speech Representations from Multiple Self-Supervised Tasks," in *Proc. of the Conf. of the Int. Speech Communication Association (INTERSPEECH)*, 2019, pp. 161–165. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2019-2605

[26] M. Ravanelli, J. Zhong, S. Pascual, P. Swietojanski, J. Monteiro, J. Trmal, and Y. Bengio, "Multi-task self-supervised learning for robust speech recognition," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6989–6993.

[27] C. Spearman, "The proof and measurement of association between two things," *The American journal of psychology*, vol. 100, no. 3/4, pp. 441–471, 1987.

[28] S. Samson, D. Dellacherie, and H. Platel, "Emotional power of music in patients with memory disorders: Clinical implications of cognitive neuroscience," *Annals of the New York Academy of Sciences*, vol. 1169, no. 1, pp. 245–255, 2009.

[29] P. Vuilleumier and W. Trost, "Music and emotions: from enchantment to entrainment," *Annals of the New York Academy of Sciences*, vol. 1337, no. 1, pp. 212–222, 2015.

[30] C. Harte and M. Sandler, "Automatic chord identifcation using a quantised chromagram," in *Audio Engineering Society Convention 118*. Audio Engineering Society, 2005.

[31] 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, 2020.

[32] A. T. Cemgil, B. Kappen, P. Desain, and H. Honing, "On tempo tracking: Tempogram representation and kalman filtering," *Journal of New Music Research*, vol. 29, no. 4, pp. 259–273, 2000.

[33] K. Zhang, H. Zhang, S. Li, C. Yang, and L. Sun, "The pmemo dataset for music emotion recognition," in *Proceedings of the 2018 acm on international conference on multimedia retrieval*, 2018, pp. 135–142.

[34] 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, 2020.

[35] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[36] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[37] Y. Gong, C.-I. J. Lai, Y.-A. Chung, and J. Glass, "Ssast: Self-supervised audio spectrogram transformer," in *AAAI Conference on Artificial Intelligence*, 2022.

[38] K. Choi, G. Fazekas, M. Sandler, and K. Cho, "Transfer learning for music classification and regression tasks," in *The 18th International Society of Music Information Retrieval (ISMIR) Conference 2017, Suzhou, China*. International Society of Music Information Retrieval, 2017.

[39] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR (Poster)*, 2015.

[40] J. Thickstun, Z. Harchaoui, D. P. Foster, and S. M. Kakade, "Invariances and data augmentation for supervised music transcription," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 2241–2245.

[41] I. Steinwart, D. Hush, and C. Scovel, "An explicit description of the reproducing kernel hilbert spaces of gaussian rbf kernels," *IEEE Transactions on Information Theory*, vol. 52, no. 10, pp. 4635–4643, 2006.

EFFICIENT NOTATION ASSEMBLY IN OPTICAL MUSIC RECOGNITION

Carlos Penarrubia[1] **Carlos Garrido-Munoz**[1]
Jose J. Valero-Mas[2] **Jorge Calvo-Zaragoza**[1]
[1] U. I. for Computing Research, University of Alicante, Spain
{carlos.penarrubia, carlos.garrido, jorge.calvo}@ua.es
[2] Music Technology Group, Universitat Pompeu Fabra, Spain
josejavier.valero@upf.edu

ABSTRACT

Optical Music Recognition (OMR) is the field of research that studies how to computationally read music notation from written documents. Thanks to recent advances in computer vision and deep learning, there are successful approaches that can locate the music-notation elements from a given music score image. Once detected, these elements must be related to each other to reconstruct the musical notation itself, in the so-called *notation assembly* stage. However, despite its relevance in the eventual success of the OMR, this stage has been barely addressed in the literature. This work presents a set of neural approaches to perform this assembly stage. Taking into account the number of possible syntactic relationships in a music score, we give special importance to the efficiency of the process in order to obtain useful models in practice. Our experiments, using the MUSCIMA++ handwritten sheet music dataset, show that the considered approaches are capable of outperforming the existing state of the art in terms of efficiency with limited (or no) performance degradation. We believe that the conclusions of this work provide novel insights into the notation assembly step, while indicating clues on how to approach the previous stages of the OMR and improve the overall performance.

1. INTRODUCTION

Optical Music Recognition (OMR) is the field of research that enables the automatic reading of music notation from scanned documents [1]. OMR has become increasingly important due to its potential for a better preservation of music archives, while also facilitating new data to the wealth of Music Information Retrieval algorithms that rely on symbolic formats [2, 3].

As in many other fields, deep learning brought about a drastic change in the performance of the proposed approaches for OMR [4]. As we will mention in the next section, tasks that used to be a difficult barrier are now feasible and successful models are known, e.g., staff detection [5] or the identification of musical symbols in the image [6]. However, although these tasks are the first obstacles of an OMR system, they are not enough to complete the process. Once the graphic elements have been identified, it is necessary to reconstruct the musical notation itself by inferring the syntactic relationships that exist between such elements, namely *notation assembly*.

To account for all existing relations, the retrieval is usually performed in a pairwise fashion among all the identified graphic units. On this note, given the (typically large) density of elements within music score images, the task exhibits a high computational complexity that complicates its integration in an end-user application. Therefore, in addition to accuracy, one must carefully take into account the efficiency of this type of schemes.

This work addresses the efficient estimation of all the syntactic relations among the elements of a music score using neural network. More precisely, we propose and assess two approaches to address this task in an efficient manner: one that is based on classifying each pair of elements employing a series of numerical features, while the other uses asymmetric kernels [7], which can be computed with high parallelization and provide results very fast. In our experiments, using the well-known MUSCIMA++ corpus, we will compare the trade-off between effectiveness and efficiency that these methods provide and discuss the experimental outcomes. In addition, assuming that the previous stages of the process may contain errors, we also assess the robustness of the assembly proposals by intentionally degrading the estimations of these precedent phases. This analysis is expected to provide useful insights for the adequate design of notation assembly methods in OMR.

The remainder of the paper is as follows: in Section 2, we provide some background on the field of OMR; in Section 3, we present the problem and the proposed approaches; in Section 4, the complete experimental setup is described; in Section 5, results are reported and discussed; and, finally, the main conclusions of the work are summarized in Section 6.

2. RELATED WORK

Traditionally, OMR has been considered a multi-stage process [8]. The legacy pipeline distinguishes four stages:

© C. Penarrubia, C. Garrido-Munoz, J.J. Valero-Mas, and J. Calvo-Zaragoza. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** C. Penarrubia, C. Garrido-Munoz, J.J. Valero-Mas, and J. Calvo-Zaragoza, "Efficient Notation Assembly in Optical Music Recognition", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

(i) *image preprocessing*, including tasks such as binarization [9], distortion correction, or stave separation [10]; (ii) *music symbol detection*, including steps such as staff-line removal [11], connected-component search, and classification [12]; (iii) *notation assembly*, where the independent components are related to each other to reconstruct the musical notation [13]; and (iv) *encoding*, in which the recognized notation is exported to a specific language that can be stored and further processed by computational means [14].

With the rise of deep learning, many of these steps have been reformulated as machine learning problems to be solved by neural networks [15, 16]. Also, many stages have been merged, giving rise to models that are capable of locating and categorizing the musical elements of the given image in a single step. This task has been the subject of extensive recent research [6, 17–19]. Alternatively, the so-called *holistic* or *end-to-end* approaches that seek to perform the entire pipeline in a single step have also been proposed, often with some prior pre-processing such as staff segmentation [14, 20, 21].

Although end-to-end approaches seem promising, so far they have only been successfully implemented for monodic music collections, where there is a clear left-to-right reading order. This is useful in many of the historical music heritage, such as plainchant or mensural music, where the different voices (if any) typically appear on different pages or sections, and staves are therefore monodic in the graphical sense. However, to deal with the common western modern notation, multi-stage OMR approaches seem to be the only ones capable of dealing with such complexity [4].

However, despite the aforementioned recent advances in the detection of music symbols with deep learning, there are hardly any proposals that complete the notation assembly stage employing machine learning techniques. To our knowledge, the only existing work that focuses on the retrieval of relationships using learning techniques is that of Pacha et al. [22]. In such work, for each pair of nodes, a single image is built with different channels: one that depicts the area of the image that contains both nodes, another that depicts the same region but only shows the first node, and a last one that depicts also the region of interest but only with the second node. A Convolutional Neural Network (CNN) is then trained to recognize whether or not there is a relationship between the nodes involved in this three-channel image. Despite the reported good results, the approach is tremendously inefficient, since it requires the independent construction and classification of an image for each pair of nodes. As we will see below, this scheme entails a huge computational complexity that makes it infeasible to use in practice.

In this paper, we especially focus on providing a solution to the notation assembly stage with a level of efficiency that enables its use in a real system, while keeping good accuracy figures.

3. METHODOLOGY

This paper follows the formulation proposed in previous works [19, 22, 23], where it is assumed that the computational reading of a music score, in the context of OMR, can be described by retrieving a graph structure from the image. In this graph, the atomic notation elements (referred to as "primitives") represent nodes, while edges denote the relationships between them. Here, we are particularly interested in the retrieval of the edges, once the nodes have been detected somehow (for instance, with the existing approaches mentioned in the previous section). Note that, instead of relying on case-specific heuristics, we frame the task within a learning-based formulation due to its inherent capability of modeling any relationship among the primitives as far as there exists a set of annotated reference data. Therefore, the formulation is general and can be used as long as there is a training set consistent with the envisioned model for the music-notation graph.

3.1 Formulation

A graph is a mathematical structure that models pairwise relationships between elements—referred to as nodes or vertices—through its edges. Here, we aim to retrieve the edges (relationships) between each pair of nodes in music scores, where each node represents a music primitive—e.g., a notehead, a stem, or an accidental. [1] The formal definition of the problem is as follows.

We assume that for a given music score $\mathbf{s}$ there exists a graph $g_{\mathbf{s}}$ that represents its symbolic music notation. The graph is defined as a pair (V, E), where V denotes the set of nodes and E denotes the set of edges. Two nodes $v_i, v_j \in V$ are connected if there exists an edge $e_{i,j} = (v_i, v_j) \in E$.

In the context of OMR, information about the set of symbols V corresponds to the *music symbol detection* stage of the OMR pipeline. Although they are still far from perfect, there are approaches in the literature that address this stage (cf. Sect. 2). Therefore, we here assume that there exists a function that maps $\mathbf{s}$ onto set V. Typically, each symbol $v_i \in V$ is further represented as a set of features with, at least, the following information: primitive class and coordinates within the image score. The problem we address from now on is how to get the set E given V, which corresponds to the *notation assembly* stage of the OMR pipeline.

The problem can be considered as a binary classification task in which a model predicts the class between each pair of nodes v_i, v_j present in the score. In this regard, $e_{i,j}$ is labeled as a 1 if there is a relationship between v_i and v_j, and 0 otherwise. The prediction of the relationship—henceforth, $\hat{e}_{i,j}$—can be represented as a function $\varphi(v_i, v_j)$ that takes the two nodes' features as input and computes the probability of connection, i.e., $\mathrm{P}(e_{i,j} = 1)$. Figure 1 depicts a general outline of the methodology adopted in this work.

[1] Hereafter, we use the terms "node", "symbol", and "primitive" interchangeably: a graphical element placed in the music score with certain attributes.

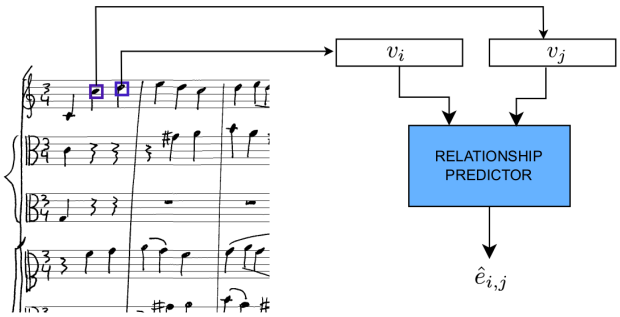


Figure 1: General schema of the methodology for retrieving the edges of the music notation graph.

3.2 Approaches

From the formulation given above, it is important to emphasize that the complexity of predicting each possible edge belongs to $O(|V|^2)$. Therefore, the approaches to φ must take into account the computational cost to make the task feasible in practice. This is a driving criterion for our approaches below since, with a sufficient number of primitives in a score, no system depicting the aforementioned complexity would be practical in a real-world scenario.

We here propose two shallow neural architectures that take a pair of nodes and predict the class of the relationship. These two neural architectures are: (i) a Multilayer Perceptron (MLP) architecture that takes the input of each node's features concatenated; and (ii) an asymmetric kernel model.

3.2.1 MLP architecture.

In this method, the features—attributes—of the nodes are first concatenated, forming a single feature vector that contains the entire information from the pair of nodes. Then, this vector is passed through a series of layers of an MLP. The final layer implements a function σ that models the probability of the two input nodes being connected:

$$\hat{e}_{i,j} = \sigma(\varphi_{\text{MLP}}([v_i, v_j]))$$

3.2.2 Asymmetric kernels.

In this second scheme, our proposed neural architecture learns an asymmetric kernel (AsymK) function [7]. This function is defined by $k(v_1, v_2) = (\langle \phi_{k_1}(v_1), \phi_{k_2}(v_2) \rangle)$, where $\langle \cdot, \cdot \rangle$ is the dot product of two N-dimensional points in two Hilbert spaces—features spaces. In this work, we use this asymmetric kernel as a similarity function between the two mapped features to distinct Hilbert spaces as:

$$\hat{e}_{i,j} = \sigma(\langle \phi_{k_1}(v_i), \phi_{k_2}(v_j) \rangle)$$

In this approach, $\phi_{k_1}(v_1), \phi_{k_2}(v_2)$ are kernels implemented as dense neural layers that map the initial node features onto two different (asymmetric) spaces that suit the task at hand. After computing the similarity score, a σ function is applied to obtain probabilities between 0 and 1.

Note that, since the embeddings are calculated only once per node, the scheme is remarkably efficiency. For each possible relationship, it is only necessary to compute the dot product between node embeddings and apply the σ function. That is why the complexity is much lower than the previous approach.

3.2.3 Loss function.

In both neural architectures proposed, the objective is to minimize the binary cross-entropy (BCE) loss function

$$\mathcal{L}_{\text{BCE}} = \sum_{e_{i,j} \in E} e_{i,j} \log(\hat{e}_{i,j}) + (1 - e_{i,j}) \log(1 - \hat{e}_{i,j}) \tag{1}$$

where $\hat{e}_{i,j}$ corresponds to the probability predicted by the model and $e_{i,j}$ is the ground-truth data for the edge (1 for a positive relationship, 0 otherwise).

4. EXPERIMENTS

In this section, we describe the experimental setup for evaluating the neural architectures proposed. More precisely, the rest of the section presents the corpus considered for the experiments, the contemplated figures of evaluation, the implementation details of the two neural proposals, and the feature descriptions used.

4.1 Data

The experiments were carried out using the MUSCIMA++ dataset [23]. This corpus provides 140 handwritten music scores with manual annotations of the different musical symbols—primitives defined by the symbol bounding box and the corresponding class label—and existing relationships among them. The dataset provides the direction of the edges; in our work, however, an undirected edge is assumed between two nodes that are connected regardless of the specific direction (undirected graph). Figure 2 depicts an example from this corpus.

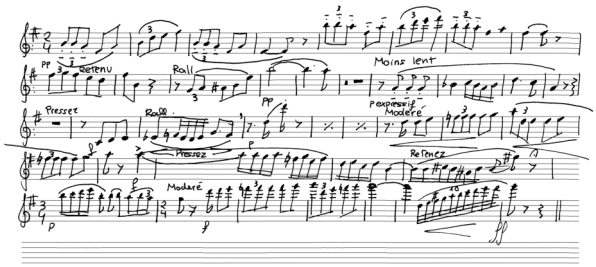


Figure 2: Example of a music score extracted from the MUSCIMA++ dataset.

Concerning the data partitioning, we follow a 5-fold cross-validation scheme. At each iteration, 60% of the dataset is used for training, 20% is used for validation, and 20% is used as test.

Finally, it must be highlighted that each music sheet depicts an average value of 734 primitives, which constitutes a large number of relations to be modeled. Due to this,

and as aforementioned, efficiency must be considered in the design of practical notation assembly strategies.

4.2 Figures of merit

We consider a two-fold assessment of the proposed approaches, *i.e.*, we evaluate their recognition capabilities as well as efficiency rates. These criteria are now detailed.

In terms of recognition performance, as in previous works considering the same evaluation corpus [22, 23], we resort to the F-measure (F_1) metric. Note that, instead of providing the average scores for the two classes, the figures reported exclusively refer to the positive relationships with the aim of measuring the quality of the retrieval.

Concerning the efficiency assessment, we measure the computation time in the prediction phase of the methods. Since this metric depends on the computational capabilities of the device used, all methods are run over the same machine to avoid any possible bias. [2] Moreover, each experiment is repeated 10 times, being the average processing time the one reported as the efficiency score.

4.3 Neural architectures

Regarding the MLP architectures, we consider two implementations with a varying number of layers and weights to balance the trade-off between efficiency and representational power:

- $\text{MLP}_{64,512}$: A three-layered fully-connected network comprising two hidden layers with 64 and 512, respectively, with *Rectifier Linear Unit* (ReLU) activations and a single output unit to compute the score of the binary classification.
- MLP_{32}: A two-layered fully-connected network comprising a 32-unit hidden layer and ReLu activation and a single unit as output.

Concerning the AsymK, ϕ_{k_1}, ϕ_{k_2} are implemented as two different 4-layered MLP comprising 512, 1024, 512, and 256 units, respectively, with ReLU activation. The idea is to generate two 256-dimensional embeddings—two points in different Hilbert spaces—to then compute the similarity through the dot product.

In all cases, the last operation is implemented as a *sigmoid* activation function to understand the output as a probability of a positive relationship. This probability is eventually thresholded considering a value of 0.5 to convert it in an actual decision.

Regarding optimization, all models were trained for 200 epochs using the Adam optimizer [24] with a learning rate of 10^{-3}.

4.4 Feature description

As aforementioned, the music-object detection stage of an OMR process retrieves, at least, the position (coordinates) of the detected object in the image sheet together with its estimated class label. For our relationship prediction approaches, we consider that each vertex ($v_i \in V$) is represented only by these features, being the inclusion of additional information left as future work.

Delving on the features considered, the spatial (position) information is directly encoded using four normalized values that denote the top-left and right-bottom corners of the bounding box. Conversely, the class information is processed by a 16-dimension learnable embedding layer to obtain an adequate representation for the task. Therefore, every single node is finally represented as a 20-dimensional feature vector.

5. RESULTS

Having introduced the different neural proposals as well as the experimental procedures, this section presents and discusses the results obtained. To establish a reference in the effectiveness that can be obtained for this task, we include the results of Pacha et al. [22], measured under the same experimental conditions as the rest of the methods in the work. [3]

The rest of the section separately studies and analyzes the two individual aspects considered, *i.e.*, performance efficiency and the ability to retrieve syntactic relationships between primitives.

5.1 Performance efficiency

Focusing first on the temporal aspect of the strategies, Table 1 shows the per-page average execution time of the contemplated notation assembly strategies. Note that, since this evaluation disregards the correctness of the estimation but simply assesses its temporal cost, all experiments are performed considering the ground-truth annotations.

Table 1: Efficiency results in terms of the per-page absolute execution time (in milliseconds) on the MUSCIMA++ corpus for the different notation assembly methods assessed. Each value corresponds to the average execution time obtained with 10 different iterations over all test samples.

	AsymK	$\mathbf{MLP_{32}}$	$\mathbf{MLP_{64,512}}$	**CNN** [22]
Execution time (ms)	< 0.5	55	176	$> 1.5 \cdot 10^6$

As can be observed, the existing CNN method [22] proves to be the least efficient among the considered strategies due to the large execution time it exhibits (roughly, 25 minutes per page). Such a point directly disables its possible integration in any practical system that comprises user interaction.

[2] The experiment was run over 8 cores of i7-7700K CPU at 4.20GHz with 16 GB of RAM memory, with no explicit parallelization or GPU speed-up.

[3] All experiments have been run considering the Python language (version 3.8), being the PyTorch (version 1.12.1) and PyTorch-lightning frameworks (1.9.1) particularly contemplated for reproducing the architecture proposed in Pacha et al. [22].

Oppositely, the different neural proposals presented in the work remarkably outperform these low-efficiency figures, achieving execution times in the order of a few milliseconds per page. More in detail, the AsymK stands as the most efficient strategy of the proposed ones as it reports figures several orders of magnitude faster than the MLP_{32} and $\text{MLP}_{64,512}$. Note that this is because AsymK exploits the parallelization of the dot product operation and the independent node processing while the two other proposals require more computation because of the classification framework they are based on.

It must be pointed out that the presented neural architectures depict execution times several orders of magnitude faster than the reference state-of-the-art method. In this regard, while the CNN strategy may be further optimized, the difference with the AsymK case—the most efficient strategy—must be considered as insurmountable.

5.2 Recognition capability

Let us now move to compare the estimation goodness of the different notation assembly strategies. As aforementioned, these methods take as input the result from a given framework that detects the primitives in the music score, *i.e.*, an object detection strategy.

Taking this into consideration, we will not consider any existing object detection approach as starting point, since other additional issues should be taken into account which are outside the scope of this work—*e.g.*, which object detection strategy to use, what confidence level to actually retrieve an object, or how to evaluate cases where a node is missing or has been predicted with the wrong label. Note that, since these questions are not related to the retrieval of the relationships themselves, any decision in this regard might bias the analysis of the models for the targeted notation assembly stage.

Nevertheless, to cover a greater number of possibilities in an agnostic way to the considered music-object detection step, we will simulate inaccuracies in the location process of the nodes. Specifically, we will consider a set of ranges for the Intersection over Union (IoU) metric between the original objects—the ground-truth annotations—and those generated for this experiment.[4] For that, assuming that the MUSCIMA++ corpus depict an IoU $= 1$ (ground-truth annotation), we will progressively perturb the location of the objects—*i.e.*, altering the coordinates of the bounding boxes—so that the overall IoU metric degrades to the range IoU $\in [0.85, 0.95]$. This range will decrease in steps of 0.1 (i.e., [0.75–0.85], [0.65–0.75], ..., [0.05–0.15]) to simulate scenarios depicting more limited symbol detection methods. In this way, our study focuses on general advantages and limitations of the notation assembly models, which can be then considered for developing more adequate pipelines for the previous stages. Figure 3 shows examples of how node locations are perturbed at some of the IoU levels considering the proposed strategy.

Considering this experimental set-up, Figure 4 shows the recognition rates in terms of F_1 achieved by the different neural schemes contemplated with respect to increasing IoU conditions (x-axis).

For the ideal scenario of perfect object-detection retrieval (IoU $= 1$), the reference CNN method [22] reports the highest recognition rate among all the schemes, with a value of $\text{F}_1 = 93.0\%$. However, the $\text{MLP}_{64,512}$ proposal shows slightly lower figures to the reference strategy—$\text{F}_1 = 91.9\%$—thereby proving itself as a competitive alternative to the CNN-based method in terms of accuracy. In relation to the AsymK and MLP_{32} proposals, these two strategies depict the least competitive results among the ones studied. However, since the MLP_{32} case shows a more competitive performance than the AsymK method, the former may be deemed as an intermediate case among the best-performing strategies—CNN and $\text{MLP}_{64,512}$—and the AsymK approach.

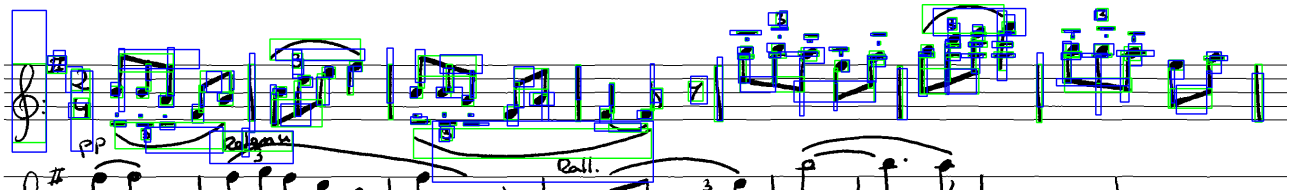
As the music-object detection becomes more realistic (IoU < 1), the neural models (except for CNN, which will be discussed below) do not degrade ostensibly but exhibit certain robustness up to reasonable IoU cases (above 0.5).[5] Digging deeper into the curves, the most relevant phenomenon is that, although at the higher ranges the CNN approach maintains the best accuracy, it decays much faster than the MLPs. Specifically, from IoU $\in$ [0.75–0.85], the $\text{MLP}_{64,512}$ outperforms it in terms of F_1, while maintaining the clear advantage in efficiency reported in the previous section. Furthermore, the rather shallow MLP_{32} approach also outperforms the CNN from IoU $\in$ [0.55–0.65], which are still likely values for music-object detection. These results reflect the adequacy of the efficient approaches proposed in this work, which are not only efficient enough to be used in practice but also keep greater robustness against very common distortions in previous stages to that of notation assembly in the OMR pipeline. In contrast, the AsymK shows a similar trend to the other efficient approaches, so from the perspective of this experiment it maintains the same advantages and disadvantages already discussed above (even higher efficiency but very poor retrieval).

As a last point, it must be noted that the results reported in this work may be considered as a turning point for the development of novel approaches to music-object detection in a complete OMR workflow. For example, using the efficient approaches of this work, one can prioritize retrieving most of the objects at the cost of slightly losing some location accuracy. An object that is not detected is impossible to relate correctly, but if it is detected, even if inaccurately it might connect successfully with other nodes (see Fig. 4).

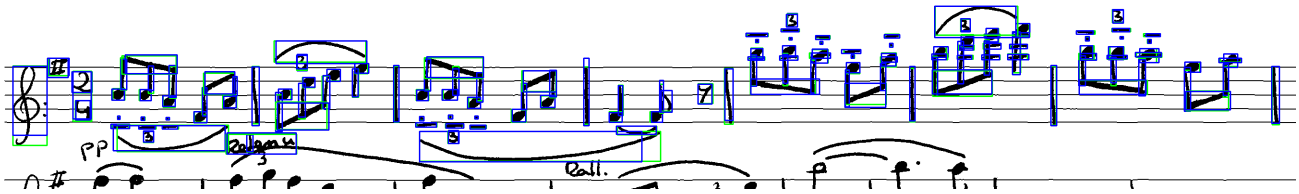
As a final note, the whole experiments clearly prove that there is no single strategy capable of optimizing both contemplated criteria at the same time: high recognition rates imply large execution times (*e.g.*, the CNN method [22],

[4] The IoU estimates the degree of overlap between two sets (in this case, areas of two music-notation objects) as the ratio of their intersection and their union.

[5] While it is true that the models obtain very poor results for the lowest ranges, this is not relevant in practice because 0.5 is the minimum IoU threshold for most object detectors to consider a correct retrieval.



(a) Simulated symbol detection performance of IoU $\in [0.45 - 0.55]$.



(b) Simulated symbol detection performance of IoU $\in [0.85 - 0.95]$.

Figure 3: Examples obtained with our proposal under different simulated symbol-detection scenarios based on the overall IoU. The green and blue bounding boxes respectively denote the ground truth and the modified ones.

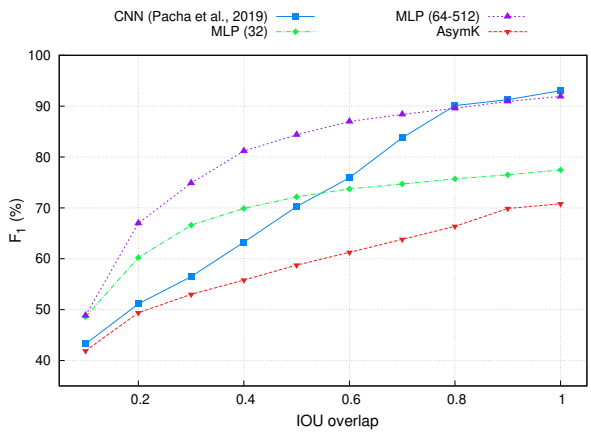


Figure 4: Results in terms of the F_1 metric for the compared note assembly strategies for different object detection performance rates based on the IoU score.

which results impractical in real-world applications) while faster strategies show more limited recognition rates (for instance, the AsymK case). In this regard, the proposed MLP-based architectures seem to provide an adequate balance between the two evaluation criteria, being particularly relevant to the $\text{MLP}_{64,512}$ one as it shows a remarkable temporal efficiency with a slightly worse performance than the highest attainable recognition results by the CNN case.

6. CONCLUSION

Optical Music Recognition (OMR) represents the research field that studies how to computationally read music notation from written documents. Generally, these strategies comprise an initial phase in which the music-notation elements from a given image are located—symbol detection—followed by a notation assembly stage that estimates the relations among these elements to reconstruct the musical notation itself. However, while there exist a large number of approaches that address the former process, the latter one has been scarcely addressed in the related literature.

This work frames in this particular assembly stage. Considering the high number of possible relationships in a music score, this work proposes two neural architectures to address this task in an efficient manner: (i) a strategy based on a Multilayer Perceptron (MLP) scheme; and (ii) a model based on asymmetric kernels. The results obtained with the MUSCIMA++ benchmark corpus [23] show that the MLP-based approach achieves recognition rates comparable to those of the reference strategy by Pacha et al. [22] with considerably less computational cost. Moreover, the asymmetric kernel approach, while proven to be extremely fast, exhibits a noticeable loss of accuracy with respect to the highest attainable one. In addition, these results also prove MLP-based schemes as remarkably robust when facing adverse symbol detection scenarios compared to the state-of-the-art method.

Several avenues of future research are opened: on the one hand, it would be important to estimate the relevance of each error produced since it has not been yet studied what errors—missing positive relationships or predicting non-existing relationships—and what type of elements involved cause the most impact on the eventual OMR system. On the other hand, this work has considered the given labeling of the MUSCIMA++; however, it has not been explored in depth whether this annotation scheme is actually adequate for these learning algorithms. More consistent or easy-to-learn annotations may be possible, as long as the goal of correctly encoding music notation is still met. Besides, just as the music-object detection step has been integrated into a single process, it would be beneficial to train end-to-end models that take into account both object detection and notation assembly. In this way, the model could leverage contextual and semantic information, provided by the notation assembly stage, when detecting objects that would be otherwise difficult or impossible. Finally, it would be also relevant to carry out user studies to assess the usefulness of these efficient approaches in real-world OMR scenarios.

7. ACKNOWLEDGMENT

Work produced with the support of a 2021 Leonardo Grant for Researchers and Cultural Creators, BBVA Foundation. The Foundation takes no responsibility for the opinions, statements and contents of this project, which are entirely the responsibility of its authors.

8. REFERENCES

[1] D. Bainbridge and T. Bell, "The challenge of optical music recognition," *Computers and the Humanities*, vol. 35, pp. 95–121, 2001.

[2] C. Müller, *Between Digital Transformation in Libraries and the Digital Humanities: New Perspectives on Librarianship*. De Gruyter, 2020, pp. 379–384.

[3] E. R. Miranda, *Handbook of artificial intelligence for music*. Springer, 2021.

[4] J. Calvo-Zaragoza, J. Hajic Jr., and A. Pacha, "Understanding Optical Music Recognition," *ACM Comput. Surv.*, vol. 53, no. 4, pp. 77:1–77:35, 2020.

[5] 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.

[6] A. Pacha, J. Hajič, and J. Calvo-Zaragoza, "A baseline for general music object detection with deep learning," *Applied Sciences*, vol. 8, no. 9, p. 1488, 2018.

[7] W. Wu, J. Xu, H. Li, and S. Oyama, "Asymmetric kernel learning," *Technical Report, Microsoft Research*, 2010.

[8] 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, no. 3, pp. 173–190, 2012.

[9] J. A. Burgoyne, L. Pugin, G. Eustace, and I. Fujinaga, "A comparative survey of image binarisation algorithms for optical recognition on degraded musical sources," in *Proceedings of the 8th International Conference on Music Information Retrieval*, 2007, pp. 509–512.

[10] 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.

[11] J. dos Santos Cardoso, A. Capela, A. Rebelo, C. Guedes, and J. P. da Costa, "Staff Detection with Stable Paths," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 6, pp. 1134–1139, 2009.

[12] A. Rebelo, G. A. Capela, and J. S. Cardoso, "Optical recognition of music symbols - A comparative study," *Int. J. Document Anal. Recognit.*, vol. 13, no. 1, pp. 19–31, 2010.

[13] F. Rossant and I. Bloch, "Robust and Adaptive OMR System Including Fuzzy Modeling, Fusion of Musical Rules, and Possible Error Detection," *EURASIP Journal on Advances in Signal Processing*, vol. 2007, no. 1, p. 081541, 2006.

[14] A. Ríos-Vila, J. Calvo-Zaragoza, and D. Rizo, "Evaluating simultaneous recognition and encoding for optical music recognition," in *Proceedings of the 7th International Conference on Digital Libraries for Musicology*. ACM, 2020, pp. 10–17.

[15] A. Pacha and H. Eidenberger, "Towards a universal music symbol classifier," in *Proceedings of the 12th International Workshop on Graphics Recognition*. IEEE, 2017, pp. 35–36.

[16] F. J. Castellanos, A. Gallego, and J. Calvo-Zaragoza, "Automatic scale estimation for music score images," *Expert Syst. Appl.*, vol. 158, p. 113590, 2020.

[17] A. Pacha, K. Choi, B. Coüasnon, Y. Ricquebourg, R. Zanibbi, and H. Eidenberger, "Handwritten music object detection: Open issues and baseline results," in *Proceedings of the 13th IAPR International Workshop on Document Analysis Systems*. IEEE Computer Society, 2018, pp. 163–168.

[18] L. Tuggener, Y. P. Satyawan, A. Pacha, J. Schmidhuber, and T. Stadelmann, "The DeepScoresV2 Dataset and Benchmark for Music Object Detection," in *Proceedings of the 25th International Conference on Pattern Recognition*. IEEE, 2020, pp. 9188–9195.

[19] C. Garrido-Munoz, A. Ríos-Vila, and J. Calvo-Zaragoza, "Retrieval of music-notation primitives via image-to-sequence approaches," in *Proceedings of the 10th Iberian Conference on Pattern Recognition*, ser. Lecture Notes in Computer Science, vol. 13256. Springer, 2022, pp. 482–492.

[20] J. Calvo-Zaragoza, A. H. Toselli, and E. Vidal, "Hybrid hidden markov models and artificial neural networks for handwritten music recognition in mensural notation," *Pattern Anal. Appl.*, vol. 22, no. 4, pp. 1573–1584, 2019.

[21] C. Garrido-Munoz, A. Ríos-Vila, and J. Calvo-Zaragoza, "A holistic approach for image-to-graph: application to optical music recognition," *Int. J. Document Anal. Recognit.*, vol. 25, no. 4, pp. 293–303, 2022.

[22] A. Pacha, J. Calvo-Zaragoza, and J. Hajic Jr., "Learning notation graph construction for full-pipeline optical music recognition," in *Proceedings of the 20th International Society for Music Information Retrieval Conference*, 2019, pp. 75–82.

[23] J. Hajic Jr. and P. Pecina, "The MUSCIMA++ dataset for handwritten optical music recognition," in *Proceedings of the 14th IAPR International Conference on Document Analysis and Recognition*. IEEE, 2017, pp. 39–46.

[24] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations*, 2015.

WHITE BOX SEARCH OVER AUDIO SYNTHESIZER PARAMETERS

Yuting Yang[1] **Zeyu Jin**[2] **Connelly Barnes**[2] **Adam Finkelstein** [1]
[1] Princeton University [2] Adobe Research
[1]{yutingy, af}@princeton.edu, [2]{zejin, cobarnes}@adobe.com

ABSTRACT

Synthesizer parameter inference searches for a set of patch connections and parameters to generate audio that best matches a given target sound. Such optimization tasks benefit from access to accurate gradients. However, typical audio synths incorporate components with discontinuities – such as sawtooth or square waveforms, or a categorical search over discrete parameters like a choice among such waveforms – that thwart conventional automatic differentiation (AD). AD libraries in frameworks like TensorFlow and PyTorch typically ignore discontinuities, providing incorrect gradients at such locations. Thus, SOTA parameter inference methods avoid differentiating the synth directly, and resort to workarounds such as genetic search or neural proxies. Instead, we adapt and extend recent computer graphics methods for differentiable rendering to directly differentiate the synth as a white box program, and thereby optimize its parameters using gradient descent. We evaluate our framework using a generic FM synth with ADSR, noise, and IIR filters, adapting its parameters to match a variety of target audio clips. Our method outperforms baselines in both quantitative and qualitative evaluations.

1. INTRODUCTION

Synthesizers provide musicians and sound designers with flexibility for exploring sound with various audio characteristics. However, the versatility of synths also poses challenges in terms of control, because manually searching over numerous parameters to seek a particular type of sound requires expertise, time, and effort. Synth parameter inference addresses these challenges by automating this search process to find parameters that best match a given target sound. Given a synth f with parameters $\boldsymbol{\theta}$ and a target T, the search seeks the optimal parameters $\boldsymbol{\theta}^*$ to minimize some loss L between the synth output and the target.

$$\boldsymbol{\theta}^* = \text{argmin}_{\boldsymbol{\theta}}\ L(f(\boldsymbol{\theta}), T) \tag{1}$$

If the synth f can be expressed as a white box program, a straightforward solution to Equation 1 would differentiate L wrt the parameters $\boldsymbol{\theta}$, and then minimize L by gradient descent. However, in practice, typical synthesizers f contain discontinuous oscillators, like square or sawtooth waveforms, and discrete categorical parameters, such as choosing different waveforms and modules, that thwart traditional automatic differentiation (AD).

Researchers have developed several workarounds to avoid directly differentiating f. For example, genetic algorithms [1, 2] approximately solve Equation 1 at the expense of greater computation and potential artifacts from failures near local minima. Alternatively, Equation 1 may be approximated as black box models using deep learning: either the synthesizer can be approximated via a differentiable neural proxy [3,4], or the entire argmin mapping can be approximated by a parameter prediction network [5, 6]. Similarly, the parameter space can be mapped to a VAE latent space for direct control [7]. However, the flexibility of deep learning approaches is constrained, as data collection and training are typically limited to specific synthesizers with fixed parameter choices, making it impractical to directly apply trained models to arbitrary synthesizers.

Graphics researchers developed recent methods to approximate the gradient for discontinuous white box image generation processes [8–10]. These generally integrate over the discontinuous function f, and approximate the gradient for the integral $\hat{f}$. This paper builds on Aδ [10], which replaces the traditional calculus rules for AD to directly enable backpropagation on arbitrary discontinuous programs. Our method relies on the key observation that the discontinuous function will eventually be band-limited and sampled at some rate (e.g. 48kHz). Each sample represents an integration over a time interval that *may contain* a discontinuity. However, the band-limited function $\hat{f}$ is continuous so differentiation rules can be developed for it.

Our optimization framework differentiates a pre-filtered white box synth, and solves Equation 1 via gradient descent. We adapt and extend the math in Aδ [10] to differentiate discontinuous and discrete synth components, and also introduce heuristic methods for better convergence. We evaluate on a FM synthesizer and our approach finds parameters that better match the target than baselines qualitatively and quantitatively. Moreover, our framework allows musicians to incorporate domain expertise to flexibly modify and fine-tune synth modules. Because our white box approach does not incur training overhead, our framework can be flexibly applied to arbitrary synth programs.

2. RELATED WORK

Researchers have explored a variety of techniques to automatically search for optimal synthesizer parameters with-

© Y. Yang, Z. Jin, C. Barnes, A. Finkelstein. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Y. Yang, Z. Jin, C. Barnes, A. Finkelstein, "White Box Search over Audio Synthesizer Parameters", in *Proc. of the 24rd Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

out having to explicitly differentiate the synthesizer. Genetic algorithm (GA) approaches [1, 2] mutate and cross variants to search over the entire program space for arbitrary synthesizers, but suffer from excessive computation and difficulty in accurately converging to local minima without the guidance of the gradient. On the other hand, deep learning models can be used to directly predict the synthesizer parameters [5, 6, 11]. However, they heavily rely on the annotated datasets of synthesizer presets, therefore cannot be flexibly generalized to *any* synthesizer. Similarly, each trained model can only be used for one particular synthesizer patching, greatly limiting the flexibility of the method. Unlike learning methods, our approach does not rely on a dataset, and can flexibly differentiate *any* white-box program, supporting finetuning and parameter transfer between synthesizer patches. Our gradient-based process also converges more robustly than GA.

Alternatively, synthesizers can be defined by differentiable functions, therefore allowing optimal parameters to be learned through gradient descent. For example, neural audio synthesis methods use black-box neural networks to generate audio samples [12, 13]. The neural proxies can be combined with continuous synthesizer components as well, such as DDSP methods that incorporate digital signal processing modules [4, 14], and DWTS methods with learnable wavetables [15]. However, because these methods use continuous proxies, they usually do not match the exact parameterization of complicated discontinuous synthesizers, therefore cannot be flexibly used to control conventional synthesizers. Moreover, the neural modules introduce nontrivial inference overhead and are less efficient than synthesizers. Unlike the differentiable neural proxies, our method directly differentiates a white box program that can be any desired synthesizer. Therefore, it optimizes semantically meaningful parameters.

We leverage recent ideas from differentiable rendering in computer graphics. Researchers developed compiler frameworks to systematically differentiate arbitrary discontinuous programs [8, 10], and application-specific solutions to efficiently differentiate specific types of discontinuities in the rendering pipeline [9, 16]. Our method differentiates synthesizer discontinuities by combining these two approaches: we adapt the gradient rules from Aδ [10] for use with discontinuous audio waveforms, and introduce a specialized gradient rule for discrete categorical choices.

3. METHOD

This section describes our optimization pipeline for synth parameter inference. Section 3.1 introduces our approach to differentiating a synth. Section 3.2 considers loss function options. Finally Section 3.3 discusses how to explore the multi-modality and avoid local minima.

3.1 Approximating the Gradient

We introduce a customized gradient, which includes differentiating at discontinuities, avoiding plateaus with zero gradient, and efficiently differentiating IIR filters. We first

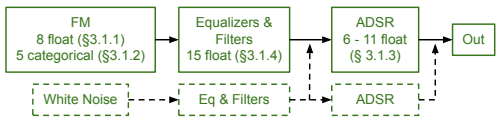


Figure 1. Summary for our FM synthesizer and how they are differentiated. Dashed boxes and arrows are optional components whose connection is decided per target.

introduce Aδ's [10] gradient rule for differentiating discontinuities and discuss its usage in audio synthesizers in Section 3.1.1, followed by our novel synthesizer-specific gradient rules in Sections 3.1.2 - 3.1.4.

3.1.1 Differentiating Discontinuous Waveforms

We view discontinuities as compositions of the Heaviside step function H, which evaluates to 0 on the one side of a discontinuity, and 1 on the other side. The discontinuity can be differentiated using the gradient rules from Aδ [10]. The key idea is to approximate the gradient as if the discontinuous function is first convolved with a 1D box filter $\phi(t)$ along the time dimension t. As an example, if H is controlled by a continuous function c, we can differentiate the convolution of $H(c(t,\theta))$ with $\phi(t)$ by applying the Dirac delta's scaling property at the discontinuity t_d.

$$\frac{\partial}{\partial\theta}\int H(c(t',\theta))\phi(t-t')dt' = \int \delta(c)\frac{dc}{d\theta}\phi(t-t')dt'$$
$$= \int \frac{\delta(t'-t_d)\frac{dc}{d\theta}}{|\frac{dc}{dt}|}\phi(t-t')dt' = \phi(t-t_d)\frac{\frac{dc}{d\theta}}{|\frac{dc}{dt}|}\,|_{t_d}$$

This can be approximated with two samples corresponding to two ends of the box kernel ϕ, denoted as t^+ and t^-. The box kernel $\phi(t-t_d)$ either evaluates to 0 or $\frac{1}{t^+-t^-}$, depending on whether $H(c(t,\theta))$ evaluates to the same or different values at t^+ and t^-. Because c is continuous, $dc/d\theta$ can be computed with AD, and its evaluation on either t^+ or t^- approximates that of t_d because Lipschitz continuous functions are locally bounded. Finally, dc/dt is approximated by finite difference: $\frac{c(t^+,\theta)-c(t^-,\theta)}{t^+-t^-}$. Because the audio signal already samples at a regular interval along the time dimension (e.g. 48kHz), we conveniently set the support of the box kernel to straddle the current sample and its neighbor. While the mathematical correctness in [10] is derived assuming a single discontinuity in the neighborhood, empirically the approximated gradient also works well for signals with sparse multi-discontinuities, such as when both the carrier and FM modulation waves are square. However, if the sampling rate is too low and causes aliasing, the Aδ rule is unable to correctly approximate the gradient as if the signal was antialiased.

Discontinuous waves such as square and sawtooth can be constructed as periodic compositions of H. These discontinuities are differentiated using the gradient rules introduced in Aδ [10] that are analogous to the equation above, where θ might e.g., be the frequency of a square wave. The gradient for the synthesizer parameters are obtained by differentiating the loss term (Section 3.2) using Aδ gradient rules, which reduce to traditional AD for continuous parameters, combined with our

customized gradients (Sections 3.1.2 - 3.1.4). This approach is more accurate than differentiating a discontinuity naively smoothed with arbitrary linear or sigmoid transitions, especially when discontinuities are composited – for example, the composition of discontinuous modulation and carrier signals in an FM synthesizer.

3.1.2 Differentiating Discrete Categorical Choices

Section 3.1 discusses a simple scenario where the discontinuity can be sampled along the time dimension. However, the challenge remains for the discrete categorical choices, because for fixed parameterization, the corresponding discontinuity H evaluates to a constant for any time t, therefore the discontinuity cannot be easily sampled.

This section proposes a stochastic approach to differentiate the discrete parameters. We define a categorical node g as taking input from a discrete parameter x with potential choices A, B, ..., and outputs to a floating point value:

$$g(x;\boldsymbol{\theta}) = \begin{cases} g_A(\boldsymbol{\theta}) & \text{if } x == A \\ g_B(\boldsymbol{\theta}) & \text{if } x == B, \\ \dots \end{cases} \tag{2}$$

g_A, g_B are floating point functions associated with choices A, B respectively, such as sine or square wave equations.

Our stochastic approach views the discrete parameter x as a discrete random variable $\mathcal{X}$ with different samples X at different time steps. Therefore $g(\mathcal{X};\boldsymbol{\theta})$ is a random variable as well. Throughout this section, we will use lowercase (e.g. x) for the synth parameters that need to be optimized, calligraphic (e.g. $\mathcal{X}$) for its corresponding random variables, and regular uppercase (e.g. X) for sampled values from the random variable. Note when $\mathcal{X}$ has close to zero variance, it consistently samples the same choice for every time step, therefore X can be viewed as a constant identical to x. We further model $g(\mathcal{X};\boldsymbol{\theta})$ similarly to an argmax operator, where each potential choice A, B, ... is associated with a "score" random variable, and the output of g corresponds to the choice with the highest "score". Specifically, the "score" for choice A is modeled as $\mathcal{Y}_A = \mu_A + \sigma_A \cdot \mathcal{U}$, where μ_A, σ_A are the mean and standard deviation, and $\mathcal{U}$ is a uniform random variable with zero mean and unit variance. For any two neighboring samples with disagreeing categorical choices A and B, we view the inconsistency as a discontinuous branching conditioned on whether the sampled "score" Y for choice A is greater than B or not: $g = \text{select}(Y_A > Y_B, g_A, g_B)$. By forming the discontinuity this way, the gradient wrt $\mu_{A/B}, \sigma_{A/B}$ can be easily computed with the Aδ gradient rules on the time domain. At convergence, the variance to every "score" variable should be reduced to a small value such that the categorical choice is sampled consistently.

The stochastic gradient rule works best when there is a high correlation between the functions associated with each choice g_A, g_B, etc. Intuitively, this allows g_A, g_B, ... to form a smaller convex hull for the sampled output $g(X;\boldsymbol{\theta})$, therefore reducing the variance of the gradient estimation. Therefore when differentiating categorical waveform choices, we align the phase of the wave functions such that their correlation is maximized.

3.1.3 Avoiding Zero Gradient in Plateaus

Many synthesizer parameters have constraints on their values, such as the period for ADSR stages should be nonnegative, and the filters' cutoff frequencies should be within a range to avoid singularities. A typical strategy for optimizing these constrained parameters in an unconstrained problem is to clamp the parameters: taking the min and max against their upper and lower bounds. However, clamping introduces another challenge for optimization: once the parameter clamped, the gradient wrt the parameter becomes zero across an entire "out of bounds" plateau in the loss function. For example, $\frac{\partial \max(\theta,0)}{\partial \theta} = 0$ whenever $\theta < 0$.

We propose a heuristic workaround that avoids constrained parameters getting stuck at out-of-bound values, via a customized gradient for the min (or max) operator f:

$$\begin{aligned} f &= \min(\theta, C) \\ \frac{\partial L}{\partial \theta} &= \text{select}(\theta < C, \frac{dL}{df}, \max(\frac{dL}{df}, 0)) \end{aligned} \tag{3}$$

Here the min operator compares with constant C, and we assume the gradient wrt f is already computed as dL/df. Note only the blue term in Equation 3 is different from traditional AD. The gradient for the max operator is similar to Equation 3, but $<$ and max are replaced by $>$ and min respectively. Note this is only a heuristic workaround for reverse-mode AD, and can not be used for forward-mode because it computes dL/df before differentiating f. Intuitively, our customized gradient will push the out-of-bound θ back to its valid range whenever the gradient wrt f wishes to bring the clamped value back to valid. We only apply this workaround when constraining parameter values against a constant, and generic min/max comparisons between two non-constants are still differentiated by AD.

3.1.4 Efficient IIR Filter Back-propagation

Infinite impulse response (IIR) filters are widely used in synths to flexibly control the timbre. However, differentiating the IIR filter introduces performance challenges because each output value at a certain time step recurrently depends on every input/output value in previous steps, and naively unrolling the gradient in the time domain is computationally expensive. We, therefore, avoid the complex dependency in the time domain by applying the filter in the frequency domain similar to [3]. During optimization, we only differentiate the multiplication between the unfiltered spectrogram and the frequency response of the filters. Because most popular filters (e.g. Biquad, Butterworth) used in synthesizers already have closed-form solutions for their frequency responses, requiring a frequency domain proxy does not restrict the expressiveness of this approach.

3.2 Loss Function

Unlike supervised deep-learning methods that could rely on losses in the parameter space at the cost of collecting the preset dataset, our optimization pipeline can only rely

on spectral and time domain losses. However, finding the ideal loss that is consistent with human perception is challenging for several reasons. Firstly, standard losses such as L2 on the (log mel) spectrogram only work well when distances between two signals are smaller than just noticeable difference (JND); but this is rarely the case during our optimization, as we start with random initial guesses, and the synthesizer may never even approach JND to an out-of-domain target. Furthermore, although deep perceptual metrics have been developed for speech signals (e.g., [17]), they do not generalize well to music synths.

We propose a heuristic combination of several different losses to approximate the perceptual similarity. The intuition is that the gradient to the majority of the losses should agree with human perception even if a few of them are noisy. In addition to standard losses, we also include the 1D Wasserstein distance [18] along the frequency dimension because of its wide applicability in matching distributions. Our final optimization loss is a weighted combination of the Wasserstein distance, L2, log mel L2, and a deep feature distance from the wav2clip model [19]. The weights are chosen such that each component has a relatively equal contribution. For the losses that work on a spectrogram (L2, log mel L2, and Wasserstein), we use three different window sizes (512, 1024, 2048) with 75% overlap between windows. The deep feature loss also uses the same window and hop sizes, but for efficiency, we stochastically evaluate the model using one of the window sizes per iteration. Additionally, because the deep feature model takes time domain signal as input, we need to apply inverse STFT to the spectrogram because of the frequency domain IIR approximation described in Section 3.1.4.

3.3 Identifying Perceptually Similar Results

Gradient-based optimizations may converge to a variety of local minima with different perceptual similarities to the target. Our framework runs multiple random restarts to avoid getting stuck at local minima. However, we are not aware of a quantitative metric that reliably characterizes the perceptual similarity for synthesizers [20, 21]. While we use our weighted loss in Section 3.2 to provide a gradient for the optimization, its absolute value does not precisely correspond to perceptual similarity: perceptually dissimilar results sometimes have lower loss than similar results. Thus, manual selection is needed to choose the best results. We also implement early termination to avoid wasting compute at local minima, and also a mechanism to identify good quality results after convergence.

Our early termination strategy is a generalization to the intuition that good initializations have a higher probability of good convergence. We generalize the heuristic to arbitrary iterations within the optimization, and terminate the ones with bad results at the end of a sequence of predetermined iterations. Additionally, because the weighted loss in Section 3.2 cannot reliably characterize perceptual similarity, we rely on the Pareto ranking [22] on multiple losses to identify bad results. We terminate optimizations whose Pareto rank on every non-deep-learning loss in Section 3.2 is higher than ceil(0.5max_rank), where max_rank is the maximum Pareto rank for the current population. Our implementation checks for early termination every 100 iterations, starting at iter 200, and we run every optimization until full convergence and simulate the early termination.

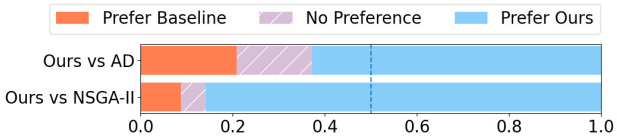


Figure 2. MOS listening test preference distribution.

We also note that when the optimization result is already close to the target at convergence, its loss metrics calculated from a larger window size better resemble perceptual similarity. Specifically, large L2 errors are usually bad. We therefore further omit any converged result whose L2 loss on the spectrogram with window size 2048 is 2x higher than the lowest among all results, and finally rank the remaining results based on the weighted sum of Wasserstein, L2, and log mel L2 on the same spectrogram.

4. VALIDATION

This section validates our proposed framework by optimizing the parameters of an FM synthesizer to match various audio signals for musical instruments and special sound effects. All the targets are downloaded from the web and are therefore out of domain. We first describe our FM synthesizer in Section 4.1 and evaluation setup in Section 4.2, then compare our method with two baselines through a user study (Section 4.3). Section 4.4 also shows the optimization convergence. Finally, Section 4.5 demonstrates the flexibility of our framework with a case study that modifies the synthesizer modules for better quality result.

4.1 Synthesizer Model

We choose an FM synthesizer as in Figure 1 following the recommendation from a synthesizer expert, because it is simple yet expressive enough to approximate most of our target signals. It has one carrier signal modulated by the weighted sum of four different signals. Each signal is parameterized with a categorical choice from the four base waveforms: sin, square, triangle, and sawtooth. Each modulation signal is also parameterized by ratio and index, which controls the frequency and the magnitude of the modulation. The FM signal will further be filtered by three Biquad equalizers (low/high shelf, peak) parameterized by their cutoff frequency, resonance and gain, and a pair of Butterworth low/high pass filters parameterized by their cutoff, bandwidth, and attenuation. After that, the filtered signal is multiplied by an ADSR parameterized with the duration of each stage, overall volume and that of sustain, the starting time of the attack, and optionally the exponential decay of the release as well as the scale, frequency, and phase to an optional AM envelope applied to attack, decay, and sustain. Finally, filtered white noise can be optionally added either by sharing the original ADSR or with a differ-

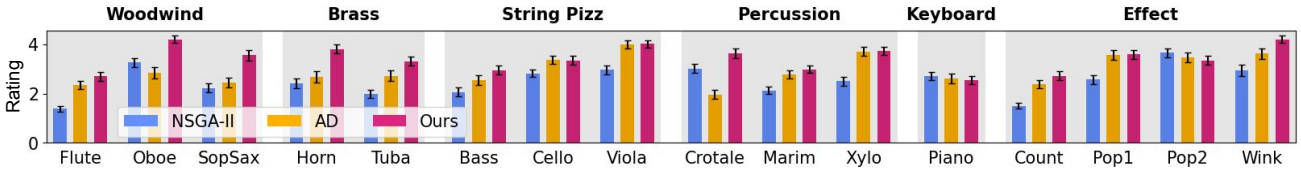


Figure 3. MOS listening test ratings (higher is better) for each of 16 target clips, grouped in 6 categories. Error bars correspond to 2SEM (standard error of mean). To save space we shorten names: Marim(ba), Xylo(phone), and Count(down).

ent ADSR. Optional configurations are included based on audio characteristics. For example, sustained sounds such as woodwind and brass uses the AM envelope for ADSR, and shorter sound such as percussion includes a filtered white noise with separate ADSR. The overall model includes 40 (e.g. oboe) - 70 (e.g. crotale) parameters.

We implement the FM synthesizer in PyTorch to leverage its AD framework. The gradient discussed in Section 3.3 is implemented as the cutomized backward pass, and AD is used for the rest of the computation (e.g. ADSR, STFT). Note this could also be generated by a compiler for arbitrary synthesizers similar to Aδ [10].

4.2 Evaluation Setup

We compare with two baselines: traditional AD and zeroth order optimization with genetic algorithm NSGA-II. AD baseline uses the same optimization framework described in Section 3, except that the gradient described in Sections 3.1.1 - 3.1.3 is replaced by traditional AD. The zeroth order baseline does not require any gradient, and instead uses the genetic algorithm NSGA-II [23] to search over the parameter space. Because NSGA-II is multi-objective, it directly finds Pareto optimal solutions to the various loss functions in Section 3.2 without having to compute their weighted sum as in gradient-based optimization.

We use 16 different target sounds, including 12 musical instruments and 4 special sound effects listed in Figure 3. For ours and AD, we run the experiment with 100 random restarts for a maximum of 2000 iterations per restart. Note that because of the early termination described in Section 3.3, the actual number of iterations per restart varies. We additionally supply the NSGA-II with a reasonable sample range to the parameters, and run the algorithm with 100 population size and 2000 generations.

4.3 MOS Listening Test

As mentioned in Section 3.2, we have no perceptually-accurate loss for comparing synth output to target audio. Therefore, we rely on a Mean Opinion Score (MOS) test to qualitatively compare results for our method and baselines. For each method and target sound, we use the top 4 results based on the Pareto ranking from Section 3.3 for testing, resulting in 12 samples across 3 methods: ours, AD, and NSGA-II. These clips may be heard in our supplemental material.

Workers on Amazon Mechanical Turk (AMT) rate how similar each result is to the target on a scale of 1 (bad) - 5 (identical). They are "master" workers, English-speakers in the US, and are paid $20 per hour. Each worker is asked to rate all 12 samples for two different targets. We further embed four validation tests to filter out careless ratings: two that are intentionally corrupted from the two targets to be worse than any of the 12 samples to be rated, and two that are identical to targets randomly chosen from all 16 targets. Therefore each worker rates $2 \times 12 + 4 = 28$ samples for each assigned task called HIT (Human Intelligence Task). In the end we collected 240 valid HITs where each audio sample gets 30 ratings from 30 different workers.

We compute a preference score for each worker and each instrument: we calculate a mean rating for each method over the 4 rated samples. If Method 1 has a higher score than Method 2, we say that Method 1 is preferred by this worker. Figure 2 shows the preferences among pairs of methods aggregated across all workers. Our method outperforms both baselines by a larger margin, but AD is preferred more than NSGA-II. We compute the p-value for the hypothesis: *our average rating per user per instrument is higher than that of the baseline.* The p-value for the AD baseline is 2e-8, and for the NSGA-II baseline is 3e-61.

We additionally report in Figure 3 the rating for each target. Ours performs best when the FM synth is a good emulation of the underlying instrument , such as for woodwind or brass. AD has similar ratings to ours more frequently than NSGA-II, which is consistent with Figure 2. Note in all cases when baselines have similar or higher ratings than ours, the rating difference is always within the error bar, indicating the preference is not statistically significant. We characterize the cases where ours and baselines have similar ratings into two scenarios. The first one is when the target is less challenging, and can be easily reconstructed by various local minimums, such as Pop1 and Pop2. The second scenario is when the FM synth cannot nicely emulate the instrument, such as for Piano. Therefore none of the methods can converge close enough to the target, resulting in similarly low ratings.

4.4 Optimization Convergence

This section discusses the optimization convergence to demonstrate how frequently each method converges in the optimization. Figure 4 demonstrates two representative results: Horn for ours outperforms baselines and Xylophone for ours performs similarly to baseline AD. In both plots, all 100 populations for NSGA-II converge similarly because bad results are removed at the end of each generation. Unlike genetic algorithms, the 100 optimizations for both ours and AD have diverging performances because gradient-descent only explores the local parameter space and may be stuck at a local minimum. The early termina-

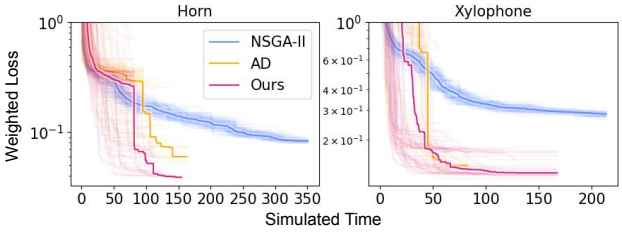


Figure 4. Comparing the convergence of ours and baselines for the 100 random restarts of two tasks. The x-axis reports simulated time: the number of function evaluations scaled with the actual runtime for each method. The y-axis reports the weighted loss for the optimization. For ours and AD, each transparent line corresponds to a restart. For NSGA-II, each transparent line plots the loss for the kth population at each generation (k $\in [1, 100]$). The median within all runs at a given time is shown as the solid line.

tion described in Section 3.3 conservatively removes some of the local minimums, but more importantly reduces the number of evaluations toward the end of the optimization because fewer restarts are still active. Typically, the convergence plot is consistent with the listening test result in Figure 3, but with the exception of Oboe, where NSGA-II converges to the lowest error, but its listening test performs worse than ours. But this is simply due to the choice of weights that combine multiple losses into one scalar: NSGA-II converges to lower Wasserstein and higher L2 and log mel L2, thus it is *not* Pareto superior to ours.

4.5 Case Study: Modify Synthesizer Modules

This section uses the Xylophone target as a case study to demonstrate that our white box method can be flexibly combined with user expertise to modify the synthesizer components to improve the quality of generated audio.

Similar to other targets, Xylophone is initially approximated by the synth model described in Section 4.1. It uses filtered white noise with independent ADSR to model the strike at the start of the sound. However, the optimization result is not ideal, specifically, the beginning of the audio sounds very different from the target. This can be verified by Figure 5, which compares the spectrogram for the first 0.07s of the sound between the target (a) and the optimization(b): the optimization has a longer attack stage.

We ask a synthesizer expert to identify the potential cause of the inconsistency: instead of using filtered white noise, the beginning of the audio may be better approximated by an impulse with IIR filters. We, therefore, use the following impulse component to replace the original filtered white noise. We first manually calibrate the starting time of the Xylophone within the target audio, and set the impulse at that location. Similar to the white noise, the impulse is also filtered by three Biquad equalizers (low/high shelf and peak) and a pair of low/high-pass Biquad filters. Because the impulse is not static, we have to optimize the IIR parameters in the time domain rather than the frequency domain as in Section 3.1.4. Therefore we avoid using any Butterworth filters mentioned in Section 4.1 for a faster backward pass. Because the original optimization

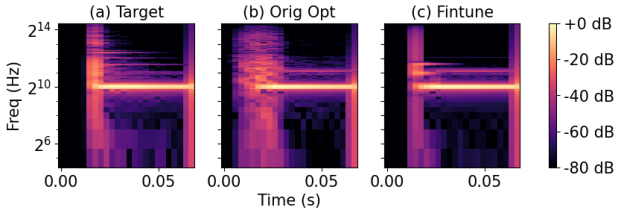


Figure 5. Visualizing the spectrogram for the Xylophone target (a), original optimization (b) using filtered white noise described in Section 4.1, and finetune result (c) using an impulse module described in Section 4.5. The spectrogram is computed with window size 512 and hop size 128.

nicely approximates the target except at the beginning, we only compute the loss for the first 2048 samples, and keep all the FM-related parameters fixed to only optimize the newly added IIR parameters, the scale of the impulse, and the original ADSR parameters that are initialized with their previously optimized values. To better characterize the filtered impulse signal, we use smaller spectrogram window sizes: 128, 256, and 512 with 75% overlap. Figure 5(c) shows the spectrogram of the finetune result that indeed better matches the attack stage of the target. Perceptually it also sounds better: please refer to supplemental material.

Note the finetuning process described in this section cannot be supported by deep learning methods without recollecting a new dataset and re-training the model for any change in the synthesizer design. Because our method directly optimizes the white-box programs, we can flexibly add the synthesizer components and reuse any parameters from previous optimizations.

5. CONCLUSION AND FUTURE WORK

This paper proposes to find synthesizer parameter settings that best match a given target sound by directly differentiating the white-box synthesizer program. We adapt and extend recent methods from differentiable rendering to differentiate the discontinuous and discrete components of the synthesizer, and design an optimization pipeline to solve the problem through gradient descent. We validate our method through user studies on Mechanical Turk, where our result is preferred over baselines by a large margin. We further demonstrate the benefit of differentiating white-box programs through a case study, where we can flexibly modify and finetune synthesizer components.

This work suggests several directions for future research. Our framework only searches for synthesizer parameters, and leaves patch connections fixed. Nevertheless, the gradient rules described in Section 3.1 provide a potential solution. It could be easily extended to optimize binary connection decisions, therefore the general patch connection could be optimized if viewed as compositions of binary choices. Additionally, because no perceptually accurate loss exists for music, our framework relies on a combination of various loss terms (Section 3.2) together with a Pareto rank based early termination strategy to improve convergence. Future work on perceptual similarity could simplify and improve our process.

6. REFERENCES

[1] M. J. Yee-King, L. Fedden, and M. d'Inverno, "Automatic programming of vst sound synthesizers using deep networks and other techniques," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 2, pp. 150–159, 2018.

[2] K. Tatar, M. Macret, and P. Pasquier, "Automatic synthesizer preset generation with presetgen," *Journal of New Music Research*, vol. 45, no. 2, pp. 124–144, 2016.

[3] N. Masuda and D. Saito, "Synthesizer sound matching with differentiable dsp." in *ISMIR*, 2021, pp. 428–434.

[4] F. Caspe, A. McPherson, and M. Sandler, "Ddx7: Differentiable fm synthesis of musical instrument sounds," *arXiv preprint arXiv:2208.06169*, 2022.

[5] G. Le Vaillant, T. Dutoit, and S. Dekeyser, "Improving synthesizer programming from variational autoencoders latent space," in *Proceedings of the 24th International Conference on Digital Audio Effects (DAFx20in21)*, Sep. 2021.

[6] O. Barkan, D. Tsiris, O. Katz, and N. Koenigstein, "Inversynth: Deep estimation of synthesizer parameter configurations from audio signals," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 12, pp. 2385–2396, 2019.

[7] P. Esling, N. Masuda, A. Bardet, R. Despres *et al.*, "Universal audio synthesizer control with normalizing flows," *arXiv preprint arXiv:1907.00971*, 2019.

[8] S. Bangaru, J. Michel, K. Mu, G. Bernstein, T.-M. Li, and J. Ragan-Kelley, "Systematically differentiating parametric discontinuities," *ACM Trans. Graph.*, vol. 40, no. 107, pp. 107:1–107:17, 2021.

[9] S. Bangaru, T.-M. Li, and F. Durand, "Unbiased warped-area sampling for differentiable rendering," *ACM Trans. Graph.*, vol. 39, no. 6, pp. 245:1–245:18, 2020.

[10] Y. Yang, C. Barnes, A. Adams, and A. Finkelstein, "Aδ: Autodiff for discontinuous programs - applied to shaders," in *SIGGRAPH, to appear*, Aug. 2022.

[11] J. Shier, G. Tzanetakis, and K. McNally, "Spiegelib: An automatic synthesizer programming library," in *Audio Engineering Society Convention 148*. Audio Engineering Society, 2020.

[12] 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.

[13] 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.

[14] J. Engel, L. Hantrakul, C. Gu, and A. Roberts, "Ddsp: Differentiable digital signal processing," *arXiv preprint arXiv:2001.04643*, 2020.

[15] S. Shan, L. Hantrakul, J. Chen, M. Avent, and D. Trevelyan, "Differentiable wavetable synthesis," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 4598–4602.

[16] T.-M. Li, M. Lukáč, G. Michaël, and J. Ragan-Kelley, "Differentiable vector graphics rasterization for editing and learning," *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, vol. 39, no. 6, pp. 193:1–193:15, 2020.

[17] P. Manocha, A. Finkelstein, R. Zhang, N. J. Bryan, G. J. Mysore, and Z. Jin, "A differentiable perceptual audio metric learned from just noticeable differences," in *Interspeech*, Oct. 2020.

[18] Wikipedia contributors, "Wasserstein metric — Wikipedia, the free encyclopedia," 2023, [Online; accessed 15-April-2023]. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Wasserstein_metric&oldid=1147354544

[19] H.-H. Wu, P. Seetharaman, K. Kumar, and J. P. Bello, "Wav2clip: Learning robust audio representations from clip," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.

[20] F. Pachet and J.-J. Aucouturier, "Improving timbre similarity: How high is the sky," *Journal of negative results in speech and audio sciences*, vol. 1, no. 1, pp. 1–13, 2004.

[21] K. Siedenburg and D. Müllensiefen, "Modeling timbre similarity of short music clips," *Frontiers in psychology*, vol. 8, p. 639, 2017.

[22] P. Sitthi-Amorn, N. Modly, W. Weimer, and J. Lawrence, "Genetic programming for shader simplification," *ACM Trans. Graph.*, vol. 30, no. 6, p. 1–12, dec 2011. [Online]. Available: https://doi.org/10.1145/2070781.2024186

[23] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002.

DECODING DRUMS, INSTRUMENTALS, VOCALS, AND MIXED SOURCES IN MUSIC USING HUMAN BRAIN ACTIVITY WITH FMRI

Vincent K.M. Cheung [1] **Lana Okuma** [2] **Kazuhisa Shibata** [2]
Kosetsu Tsukuda [3] **Masataka Goto** [3] **Shinichi Furuya** [1]
[1] Sony Computer Science Laboratories, Tokyo, Japan
[2] RIKEN Center for Brain Science, Japan
[3] National Institute of Advanced Industrial Science and Technology (AIST), Japan

ABSTRACT

Brain decoding allows the read-out of stimulus and mental content from neural activity, and has been utilised in various neural-driven classification tasks related to the music information retrieval community. However, even the relatively simple task of instrument classification has only been demonstrated for single- or few-note stimuli when decoding from neural data recorded using functional magnetic resonance imaging (fMRI). Here, we show that drums, instrumentals, vocals, and mixed sources of naturalistic musical stimuli can be decoded from single-trial spatial patterns of auditory cortex activation as recorded using fMRI. Comparing classification based on convolutional neural networks (CNN), random forests (RF), and support vector machines (SVM) further revealed similar neural encoding of vocals and mixed sources, despite vocals being most easily identifiable. These results highlight the prominence of vocal information during music perception, and illustrate the potential of using neural representations towards evaluating music source separation performance and informing future algorithm design.

1. INTRODUCTION

The goal of brain decoding is to infer mental states and perceptual information from neural activity [1, 2]. Common neuroimaging techniques such as functional magnetic resonance imaging (fMRI) and electroencephalography (EEG) allow data acquisition in a non-invasive manner, which has resulted in rapid developments in brain-computer interfaces (BCI) [3, 4].

Although fMRI- and EEG-based models both make use of neural activity for decoding, the form of information retrieved is substantially different. That is because fMRI offers (sub-)millimetre spatial resolution at the cost of low temporal resolution, whilst EEG provides a millisecond-level temporal resolution at the expense of poor spatial resolution [2, 5]. Consequently, fMRI-based decoders typically rely on spatial representations of neural activation as features, whilst EEG-based decoders exploit the temporal dynamics of neural activity.

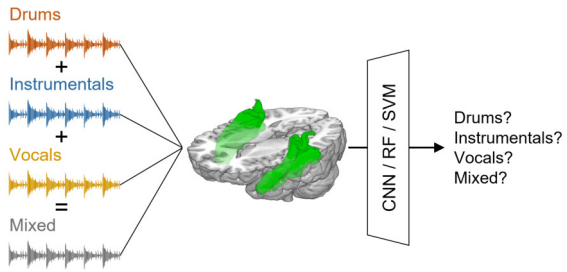


Figure 1. We compared the decoding performance of convolutional neural networks (CNN), random forests (RF), and support vector machines (SVM) in classifying drums, instrumentals, vocals, and mixed naturalistic musical sources based on human auditory cortex activation (highlighted in green) as recorded using fMRI.

In the context of music information retrieval (MIR), both fMRI- and EEG-based decoders have been employed for a variety of classification/estimation tasks, such as genre [6–9], pitch [6, 10–12], rhythm [13, 14], musical emotion classification [15–21], song identification [22–25], music composition [26], beat and note onset detection [27,28], and acoustic feature extraction [29], as well as reconstruction from heard and imagined melodies [30–35].

However, a problem that has remained under-studied is the decoding of different instruments within a song based on brain activity. This is despite its intimate relation to the standard MIR task of music source separation, which seeks to decompose a musical sound mixture into a linear sum of instrumental sources [36, 37]. Although music and speech source separation share the same goals, the key difference is that sound sources from multiple musical instruments are more correlated in music than in speech [36].

The most relevant literature on neural-driven music source separation is the work by Cantisani et al. [38, 39]. Their initial work showed that EEG can be used to decode listeners' attention deployment to a particular instrument

© V.K.M. Cheung, L. Okuma, K. Shibata, K. Tsukuda, M. Goto, and S. Furuya. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** V.K.M. Cheung, L. Okuma, K. Shibata, K. Tsukuda, M. Goto, and S. Furuya, "Decoding drums, instrumentals, vocals, and mixed sources in music using human brain activity with fMRI", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.*, Milan, Italy, 2023.

from naturalistic polyphonic music mixtures [38]. The approach was to first record listeners' EEG as they were presented with solo instrumental sources. A temporal response function was then trained to reconstruct the solo instrumental sources from EEG. This response function was later applied to the EEG signal when subjects listened to the polyphonic mixtures. The attended instrument was identified as the one that showed the highest correlation with the reconstructed source. In their subsequent work [39], they showed that the reconstructed sources from their EEG attention decoding model can be used as contrastive priors to inform a non-negative matrix factorisation-based source separation model.

On the other hand, relevant work based on fMRI data seems to be lacking. While existing studies have identified the role of the auditory cortex in processing timbre [40, 41] via correlational approaches, even the relatively simple task of decoding musical instrument category has only been restricted to single- or few-note stimuli [19, 42].

In this paper we address this gap by showing that distinct musical sources, namely drums, instrumentals, and vocals from naturalistic musical stimuli, as well as their mixtures, can be decoded from spatial representations of neural activation recorded using fMRI on the single-trial level. We report that decoding performance was the highest when detecting the presence of vocal information in the auditory stimulus, and we explain our model decisions in terms of patterns of neural activations. Importantly, unlike most existing decoding studies which have relied on a single classification algorithm, we additionally compared performance across three decoders, convolutional neural networks (CNN), random forests (RF), and support vector machines (SVM), to enhance the generalisability of our findings. In the last section of this paper, we also discuss how brain activity could be used in the future to evaluate music source separation and inform algorithm design.

2. METHODS

2.1 Experimental stimuli

Experimental stimuli consisted of 15-second musical audio excerpts derived from the beginning of the chorus section of 24 unreleased pop and rock songs within an in-house music dataset created by professional musicians.

Four versions of each song—*drums*, *instrumentals*, *vocals*, and *mixed*—were compiled, resulting in a total of 96 stimuli. The versions were produced by first separating the original song into `bass`, `drums`, `other`, and `vocals` using a state-of-the-art music source separation model, Demucs-v4 [43]. Due to the frequency response of MRI-compatible noise-isolating earphones (Sensimetrics S15), `bass` and `other` were linearly combined to form an *instrumentals* version. A 100-ms fade-out was then applied to the *drums*, *instrumentals*, and *vocals* versions, followed by loudness normalisation to the EBR U 128 standard. Finally, the normalised *drums*, *instrumentals*, and *vocals* versions of each song were linearly combined to form the *mixed* version, which was also normalised for loudness. We chose to use the *mixed* version rather than the original song to ensure that decoding was not biased by differences in loudness from the underlying versions. We made sure that each song actually included drums, vocals, and other instruments before source separation, and we checked our resulting stimuli after source separation to ensure that they were free from audible artefacts and separation errors, and that they did not contain silences at the start and end that would shorten the stimuli.

2.2 Data acquisition

Data were collected from 24 healthy, normal-hearing adults aged between 19-34 with their written informed consent. The 96 music stimuli were presented over eight runs whilst functional gradient echo planar images (TR/TA/TE = 2/2/0.025 s, voxel size = 3×3×3 mm^3, 33 slices, flip angle = 77°, 188 volumes per run) were acquired using a Siemens Prisma 3T MRI scanner. Each run lasted approximately 6 minutes, and was separated by a short break of around one minute. Stimulus presentation was counter-balanced across runs, with the constraint that each run contained three samples of the four versions, all stimuli came from different songs, and that each song (regardless of version) appeared only once every other run. Stimulus presentation within a run was randomised. To maintain attention, subjects were also asked to rate their preference on a 1-9 scale within a 4-second time window using a button box after each stimulus presentation. Our study was approved by the Ethics Committee at RIKEN.

2.3 fMRI data preprocessing

Functional MRI data for each subject were preprocessed using fMRIprep [44]. Functional images were first corrected for slice-timing differences, motion artefacts, and susceptibility distortions, then co-registered to subjects' anatomical image, and then normalised to standard MNI-space using the ICBM 152 Nonlinear Asymmetrical template. Next, for each subject, we fitted a general linear model in each voxel to estimate the blood oxygen level-dependent (BOLD) response on the single-trial level using SPM [45] following a 'least-squared all' approach [46]: each stimulus was modelled as a separate regressor in the design matrix, and a parametric modulator that varied by subjects' stimulus rating was also added to control for differences in preference. Another regressor was included to account for variance during the rating period. These regressors were modelled as boxcar functions and convolved with the canonical haemodynamic response function (HRF). Six motion, one cardiac, and one respiratory regressors were further added to the design matrix to control for motion- and physiology-induced artefacts. Model parameters were estimated using restricted maximum likelihood, and the resulting parameter estimates at each voxel provided a spatial representation (i.e., *beta maps*) of neural activations for each stimulus separately, which we used for subsequent decoding.

As we were interested in stimulus differences in the neural-perceptual level, we considered voxels in the hu-

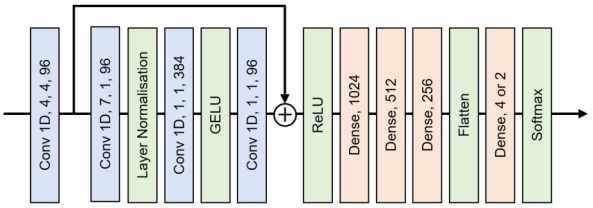


Figure 2. Architecture of our CNN-based decoder.

man auditory cortex (see Figure 1) as decoding features. These were obtained by applying a mask to the bilateral `early-auditory` and `auditory-associative` regions in the HCP-MM1 brain atlas [47,48], and then flattened into a 1D-vector using nilearn [49].

2.4 Decoding analyses

We performed two decoding analyses. The first was a four-way classification task, whose goal was to classify which of the four versions a stimulus belonged to based on subjects' brain activation as summarised by its beta map. The second was a binary recognition task, whose goal was to detect the presence of drums, instrumentals, or vocals in the stimulus from brain activation. As an example, for drum-recognition, *drums* and *mixed* versions would be assigned a positive label, whilst *instrumentals* and *vocals* versions would be assigned a negative label.

We also examined whether decoding performance depended on neural information encoded in the left, right, or both auditory cortices. This was motivated by neuroscientific findings suggesting a right-lateralised hemispheric dominance to musical stimuli [50], and that the left auditory cortex may be more sensitive to rapid temporal features in an auditory stimulus whilst the right may be more sensitive towards spectral features [51].

To enhance the generalisability of our findings, we performed leave-one-subject-out cross-validation, where each decoder was trained on data from 23 subjects and tested on 1 remaining subject. Note that brain decoding between subjects is generally harder than decoding within subjects, because the decoder must additionally overcome individual differences in structural and functional organisation of the brain when predicting on an unseen subject [52].

2.5 Implementation

We trained three types of classifiers—convolutional neural networks (CNN), random forests (RF), and support vector machines (SVM)—for our two decoding tasks. While classical approaches such as SVMs and RFs remain popular [53], CNNs have also been recently used to decode visual objects [54], vocal emotions [55], and musical pitch [10] from fMRI data. We implemented CNN decoders on TensorFlow2, whilst RF and SVM were implemented on scikit-learn [1].

Training data were first put through a variance threshold to remove features that gave identical outputs (e.g., at the boundary of the brain), and then scaled using a robust scaler, before decoders were fitted.

Our CNN decoders (see Figure 2) were inspired by ConvNeXt [56], which is a family of purely convolutional neural networks that recently achieved state-of-the-art performance in image classification. Input features first passed through a 1D-convolution layer (96 units, kernel size = 4), and a ConvNeXt-like residual block. This block comprised a 1D-convolution layer (96 units, kernel size = 7), followed by layer normalisation, 1D-convolution (384 units, kernel size = 1), GELU activation, another 1D-convolution (96 units, kernel size = 1), and a residual connection layer followed by ReLU activation. Outputs of the residual block then passed through three dense layers (1024, 512, and 256 units, respectively), a flattening layer, and finally a dense layer with softmax output. All convolution layers had a stride length of 1 (except for the first, which had a length of 4) and `same`-padding. Each model was trained to minimise categorical entropy loss for 200 epochs, and early-stopped if validation performance did not improve after 25 epochs (with best weights restored). Data from two random subjects ($\sim$10%) in the training set were held-out for validation, and we selected a batch size of 512, and an AdamW optimiser [57] with learning rate = 0.001 and weight decay = 0.0001 for training.

RF decoders were trained with bootstrapping using 100 trees in the forest, at least 1 sample per leaf, and 2 samples per split. Quality of split was assessed with Gini impurity.

SVM decoders were trained using regularisation parameter of 1 with squared-L2 penalty and a linear kernel for a maximum of 10,000 iterations.

3. RESULTS AND DISCUSSION

3.1 Four-way classification

Table 1 and Figure 3 show the leave-one-subject-out cross-validation performance of CNN, RF, and SVM decoders in classifying whether a stimulus belonged to *drums*, *instrumentals*, *vocals*, or *mixed* versions of a song, given auditory cortex activation. To test the statistical significance of our results (see Table 2), we fitted linear mixed models with the interaction between classifier and hemisphere (and lower order terms) as fixed effects and a maximal random effects structure with subject as a grouping factor.

All classifiers showed significantly above-chance performance (all $p < 2.2\times10^{-16}$, see Supplementary Information [1]) in decoding accuracy and Area Under the Receiver Operating Characteristic Curve (ROC AUC), suggesting that despite the slow temporal resolution of fMRI, correlated sources of the same song can be decoded and classified from spatial representations of brain activation.

For all classifiers, accuracy and ROC AUC were highest when decoding from the bilateral auditory cortex, followed by the right and left hemispheres. Resolving significant interaction effects between classifier and hemisphere for accuracy and ROC AUC furthermore revealed that accuracy was significantly worse when decoding from the left compared to the right and bilateral auditory cortices for CNN,

[1] For data, code, and Supplementary Information, please refer to `https://github.com/vkmcheung/neuromusic-decoding/`

	CNN		RF		SVM	
	acc	**auc**	**acc**	**auc**	**acc**	**auc**
Four-way classification						
l AC	.506	.799	.523	.802	.524	.810
r AC	.588	.858	.536	.817	.563	.843
l+r AC	**.604**	.863	.554	.824	.597	**.863**
l+r PV	.301	.560	.319	.554	.253	.510
l+r SM	.304	.547	.332	.550	.276	.554
Drums recognition						
l AC	.595	.638	.603	.637	.550	.559
r AC	.622	.677	.586	.638	.559	.591
l+r AC	**.630**	**.683**	.599	.655	.553	.601
l+r PV	.507	.505	.528	.533	.526	.531
l+r SM	.517	.544	.530	.545	.490	.500
Instrumentals recognition						
l AC	.656	.726	.638	.688	.615	.679
r AC	.642	.723	.666	.727	.627	.687
l+r AC	**.680**	**.762**	.657	.712	.652	.703
l+r PV	.577	.593	.585	.611	.495	.509
l+r SM	.558	.576	.580	.600	.517	.553
Vocals recognition						
l AC	.794	.891	.799	.913	.746	.847
r AC	.816	.926	.841	.937	.836	.936
l+r AC	.839	.946	.829	.936	**.843**	**.950**
l+r PV	.527	.527	.525	.541	.495	.502
l+r SM	.516	.544	.563	.581	.516	.552

Table 1. Mean brain decoding performance with leave-one-subject-out cross-validation. *acc* = accuracy; *auc* = ROC AUC; *l/r/l+r* = left/right/bilateral; *AC* = auditory, *PV* = primary visual, *SM* = somatosensory-motor cortices.

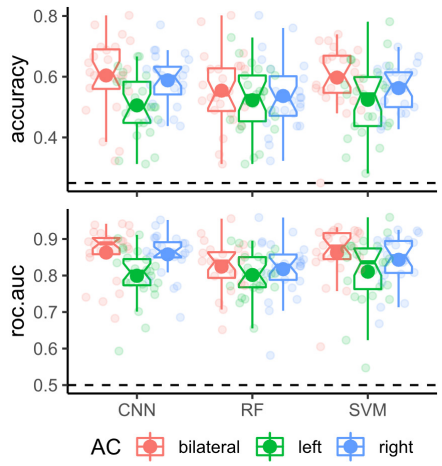


Figure 3. Box plots showing four-way classification performance when decoding from voxels in the left and/or right auditory cortex using CNN, RF, and SVM. Light circles indicate test performance on each held-out subject. Filled circles indicate mean. Dashed lines indicate chance.

and compared to the right for SVM. Likewise, ROC AUC was significantly lower when decoding from the left compared to the bilateral auditory cortex for all classifiers, and compared to the right for CNN and SVM. Although these results corroborate previous findings (e.g., [50]) that support a dominant role of the right auditory cortex in processing musical stimuli, they nevertheless show that both auditory cortices were engaged and provided useful information for decoding.

Four-way classification	χ^2	*df*	*p*
Accuracy			
hemisphere	37.3	2	8.08×10^{-9} ***
classifier	6.81	2	.0331 *
hemisphere:classifier	14.6	4	.00551 **
ROC AUC			
hemisphere	55.3	2	9.58×10^{-13} ***
classifier	11.8	2	.00278 **
hemisphere:classifier	14.3	4	.00625 **
Recognition task	χ^2	*df*	*p*
Accuracy			
hemisphere	3.30	2	.192
classifier	14.5	2	.000720 ***
task	59.9	2	9.78×10^{-14} ***
hemisphere:classifier	3.60	4	.463
task:classifier	9.76	4	.0447 *
hemisphere:task	2.90	4	.574
hemisphere:classifier:task	11.3	8	.184
ROC AUC			
hemisphere	4.89	2	.0866
classifier	13.6	2	.00114 **
task	94.2	2	$< 2.2\times10^{-16}$ ***
hemisphere:classifier	2.07	4	.722
task:classifier	10.9	4	.0275 *
hemisphere:task	2.68	4	.612
hemisphere:classifier:task	9.54	8	.299

Table 2. ANOVA table evaluating the statistical significance of hemisphere and classifier in four-way classification and recognition task performance. * = p < .05; ** = p < .01; *** = p < .001.

Confusion matrices in Figure 4(A) provide further insight on decoding performance. We notice that across the three decoders trained on both hemispheres, recall was the highest for *drums* and the lowest for *mixed*. The high recall for *drums* could be because they were the most temporally regular and had limited pitch possibilities. Furthermore, *mixed* and *vocals*, as well as *drums* and *instrumentals*, were often misclassified as the other. These suggest a similar neural representation between *mixed* and *vocals*, as well as *drums* and *instrumentals*. Whether this pairing is contingent on the stimulus set, the part of a song used (here, our stimulus excerpts were taken from the beginning of the chorus section), and the experimental design remains to be verified in future studies.

3.2 Neural representations

To explain the impact of each voxel towards classification, we turned to SHapley Additive exPlanations (SHAP) [58]. SHAP decomposes a model prediction into the additive contribution of each feature from the mean using game theory. Figure 4(B) shows the mean-averaged contribution of voxels in the bilateral auditory cortex towards classifying a stimulus as belonging to the four versions in Subject 4 using a CNN. We notice that the pattern of contributions were quite similar for *mixed* and *vocals*, which could explain the misclassification of the two labels observed above. Furthermore, there were substantial contributions from both

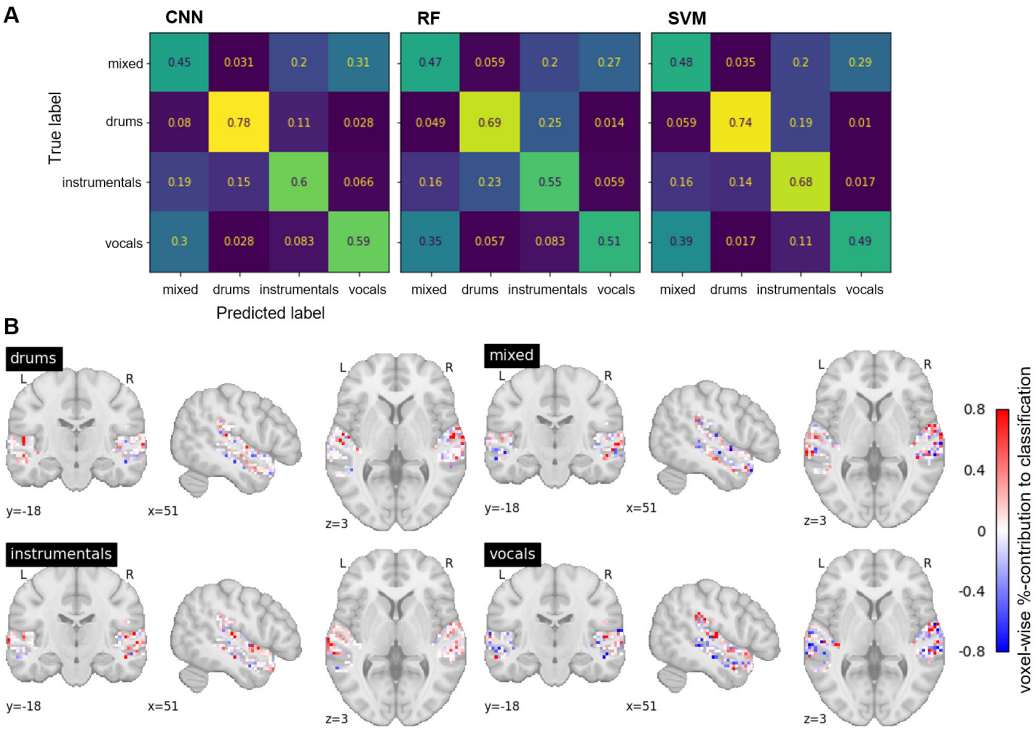


Figure 4. (A) Confusion matrix (normalised along rows) for each decoder when trained on the bilateral auditory cortices, pooled across all subjects. Notice that *drums* recall was highest, and a consistent misclassification between *mixed* and *vocals*, as well as *drums* and *instrumentals*. (B) Mean additive contribution of each voxel in the bilateral auditory cortex towards classifying a given label for one subject using a CNN decoder derived using SHAP [58].

auditory cortices, again indicating a bilateral engagement during music processing.

3.3 Recognition task

We next tested whether decoding performance in recognising the presence of drums, vocals, or instrumentals varied from the left and/or right auditory cortex. Results from leave-one-subject-out cross-validation are summarised in Figure 5 and Tables 1 and 2.

Resolving significant main effect of tasks for accuracy and ROC AUC revealed substantially higher decoding performance across CNN, RF, and SVM in recognising vocals compared to drums and instrumentals (see Supplementary Information [1]). That the presence of vocal information was most robustly encoded from neural activation patterns is very interesting, as it suggests that listeners show an enhanced sensitivity towards perceiving human voice in music. This finding is in line with the view that singing vocals play a prominent and powerful role in communicating and expressing meaning and emotion during music listening [59, 60]. We speculate that the presence of vocal information might have additionally engaged neural populations involved in language processing, which consequently increased its dissimilarity amongst other labels.

Significant main effects of classifier for accuracy and ROC AUC also indicated superior performance of CNN and RF over SVM when averaged across recognition tasks. However, significant task-by-classifier interactions for both measures suggest that performance varied according to recognition task. Resolving the interaction revealed significantly lower accuracy and ROC AUC for SVM compared to CNN and RF in drums recognition. Significantly higher ROC AUC was also observed for CNN compared to SVM in recognising instrumentals. Nevertheless, we were not able to detect any meaningful differences in laterality across tasks or classifiers.

3.4 Feature-encoding specificity

Thus far, we relied on neural activations in the auditory cortex as input features for our decoding models. To assess the specificity of information encoding, we repeated the above analyses in two other sensory processing brain regions, namely the bilateral `primary-visual`, and the `somatosensory-and-motor` regions as derived from the HCP-MM1 brain atlas [47, 48]. As before, we assessed the statistical significance of decoding performance using linear mixed models. However, rather than comparing the effects of hemisphere within the auditory cortex, we now compare performance across the bilateral auditory, primary visual, as well as somatosensory-motor cortices.

In the four-way classification task, we observe in Table 1 and the Supplementary Information [1] that decoding from the bilateral auditory cortex resulted in significantly higher accuracy and ROC AUC compared to the two other sensory cortices across all classifiers (all $p < 2.2 \times 10^{-16}$).

Interestingly, decoding accuracy and ROC AUC were also significantly above chance when CNNs and RFs were trained using features from the visual and somatosensory-motor regions (with no significant differences between these two regions). Furthermore, resolving signifi-

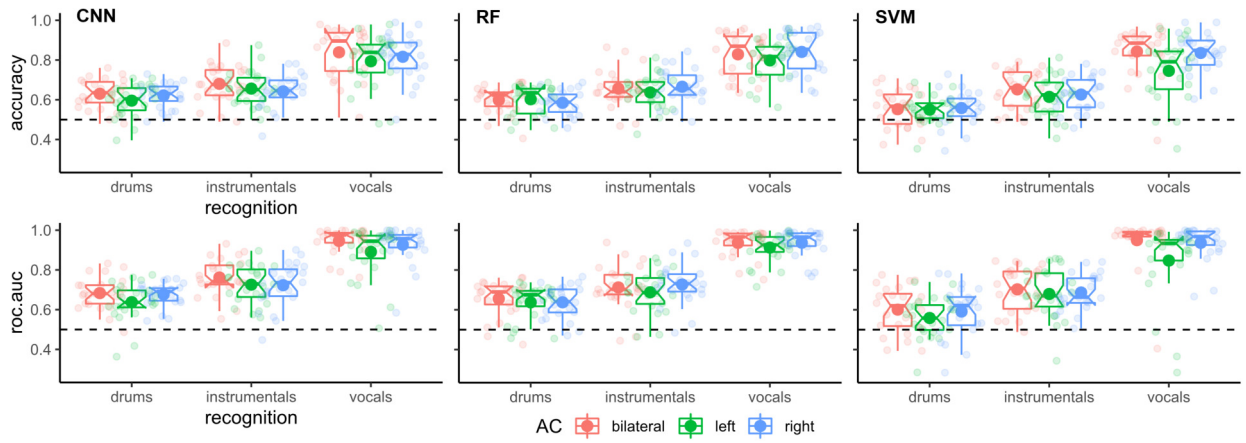


Figure 5. Box plots comparing performance in recognising the presence of *drums*, *instrumentals*, or *vocals* in a musical stimulus using left and/or right auditory cortex activation as decoding features. Significantly higher decoding performance was detected in the *vocals* recognition. Dashed lines indicate chance performance.

cant cortex-by-classifier interactions showed significantly lower accuracy and ROC AUC when decoding from the primary visual cortex using SVM compared to CNN and RF, and from the auditory cortex using RF compared to CNN, as well as lower accuracy when decoding from the somatosensory-motor cortex using SVM compared to RF.

A similar picture could be seen in recognition performance. Significant main effects of cortex for recognition accuracy and ROC AUC indicate superior performance when decoding from the auditory compared to visual or somatosensory-motor regions. Resolving significant cortex-by-task interactions further revealed that the significantly higher performance in recognising vocals compared to drums or instrumentals was specific to the auditory cortex. By contrast, accuracy and ROC AUC for instrumentals were significantly higher than drums in the auditory and somatosensory-motor areas, as well as in the primary visual cortex (ROC AUC only).

Engagement of the primary visual cortex during music has been suggested to be related to mental imagery [61,62], which is thought to be an important way through which music evokes emotions [63]. Likewise, the somatosensory cortex has been said to encode the emotional percept or feeling states associated with music [15], whilst auditory-motor interactions during music perception is thought to be related to the integration and updating of hierarchical predictions of the musical beat [64, 65]. Combined with the substantially higher performance observed when decoding from the auditory cortex, these suggest that while musical sources could be decoded from visual and somatosensory-motor regions, the information encoded is unlikely to be related to the auditory content itself. Rather, such representations might encode affective or metrical information from associated cognitive processes that arise when perceiving the four different musical sources.

4. CONCLUSION AND FUTURE PERSPECTIVES

In this paper, we demonstrated that drums, instrumentals, vocals, and mixed sources of naturalistic music can be decoded from human auditory cortex fMRI data on the single-trial, between-subject level. While decoding performance was the highest for CNN, performance across all classifiers—CNN, RF, and SVM—were above chance and suggested similar neural representations for vocals and mixed sources. An especially high performance in vocals recognition across all classifiers further pointed towards an enhanced perceptual sensitivity towards vocal information during music listening. Taken together, our results show that despite the low temporal resolution of fMRI, the high spatial resolution it offers could still provide relevant information for decoding in neural-driven MIR tasks.

Although our specificity analyses highlighted the auditory cortex in encoding stimulus-relevant information compared to other sensory areas, the perception of different musical sources is a hierarchical process that engages higher-order brain regions in the prefrontal cortex via dorsal and ventral pathways [66, 67]. Future work could examine differences in representations along these two pathways to shed light on neural mechanisms involved in auditory-object processing.

In the context of music source separation, one future possibility is to use neural data for evaluation. While current subjective evaluation of music source separation algorithms typically rely on explicit ratings such as MUSHRA or mean opinion scores, ratings are known to be prone to response biases [68–70] and might consequently fail to adequately reflect subjects' perception. This could be overcome by directly evaluating performance on the neural-perceptual level. Future work could, for example, compare the neural representations of source-separated stimuli from different algorithms or hyperparameters. Separation quality could be determined by identifying the algorithm that maximises dissimilarity in neural activation across the different sources. Another possibility is to assess sensitivity to each instrument by examining neural activation in response to different mixing proportions. This would provide perceptual priors that could be used to constrain the parameter space in future music source separation algorithms. While these prospects may seem too challenging at this time, we envision that our work will help pave the way in that direction.

5. ACKNOWLEDGMENTS

This work was supported in part by JST CREST Grant Number JPMJCR20D4 and JSPS KAKENHI Grant Number JP21H04917, Japan.

6. REFERENCES

[1] N. Kriegeskorte and P. K. Douglas, "Interpreting encoding and decoding models," *Current Opinion in Neurobiology*, vol. 55, pp. 167–179, 2019.

[2] 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*, ser. ISMIR 2015, 2015, pp. 538–544.

[3] M. Zhuang, Q. Wu, F. Wan, and Y. Hu, "State-of-the-art non-invasive brain–computer interface for neural rehabilitation: A review," *Journal of Neurorestoratology*, vol. 8, no. 1, pp. 12–25, 2020.

[4] R. Sitaram, A. Caria, R. Veit, T. Gaber, G. Rota, A. Kuebler, and N. Birbaumer, "fMRI brain-computer interface: a tool for neuroscientific research and treatment," *Computational Intelligence and Neuroscience*, vol. 2007, 2007.

[5] B. He and Z. Liu, "Multimodal functional neuroimaging: Integrating functional MRI and EEG/MEG," *IEEE Reviews in Biomedical Engineering*, vol. 1, pp. 23–40, 2008.

[6] M. A. Casey, "Music of the 7ts: Predicting and decoding multivoxel fMRI responses with acoustic, schematic, and categorical music features," *Frontiers in Psychology*, vol. 8, p. 1179, 2017.

[7] T. Nakai, N. Koide-Majima, and S. Nishimoto, "Encoding and decoding of music-genre representations in the human brain," in *Proceedings of the 2018 IEEE International Conference on Systems, Man, and Cybernetics*, ser. SMC 2018, 2018, pp. 584–589.

[8] T. Nakai, N. Koide-Majima, and S. Nishimoto, "Correspondence of categorical and feature-based representations of music in the human brain," *Brain and Behavior*, vol. 11, no. 1, p. e01936, 2021.

[9] J. S. Rahman, T. Gedeon, S. Caldwell, and R. Jones, "Brain melody informatics: Analysing effects of music on brainwave patterns," in *Proceedings of the 2020 International Joint Conference on Neural Networks*, ser. ICJNN 2020, 2020, pp. 1–8.

[10] V. K. Cheung, Y.-P. Peng, J.-H. Lin, and L. Su, "Decoding musical pitch from human brain activity with automatic voxel-wise whole-brain fMRI feature selection," in *Proceedings of the 2023 IEEE International Conference on Acoustics, Speech and Signal Processing*, ser. ICASSP 2023, 2023.

[11] K. Tsekoura and A. Foka, "Classification of EEG signals produced by musical notes as stimuli," *Expert Systems with Applications*, vol. 159, p. 113507, 2020.

[12] V. De Angelis, F. De Martino, M. Moerel, R. Santoro, L. Hausfeld, and E. Formisano, "Cortical processing of pitch: Model-based encoding and decoding of auditory fMRI responses to real-life sounds," *NeuroImage*, vol. 180, pp. 291–300, 2018.

[13] P.-C. Chang, J.-R. Chang, P.-Y. Chen, L.-K. Cheng, J.-C. Hsieh, H.-Y. Yu, L.-F. Chen, and Y.-S. Chen, "Decoding neural representations of rhythmic sounds from magnetoencephalography," in *Proceedings of the 2021 IEEE International Conference on Acoustics, Speech and Signal Processing*, ser. ICASSP 2021, 2021, pp. 1280–1284.

[14] S. Stober, D. J. Cameron, and J. A. Grahn, "Classifying EEG recordings of rhythm perception," in *Proceedings of the 15th International Society for Music Information Retrieval Conference*, ser. ISMIR 2014, 2014, pp. 649–654.

[15] S. Koelsch, V. K. Cheung, S. Jentschke, and J.-D. Haynes, "Neocortical substrates of feelings evoked with music in the acc, insula, and somatosensory cortex," *Scientific Reports*, vol. 11, no. 1, p. 10119, 2021.

[16] M. E. Sachs, A. Habibi, A. Damasio, and J. T. Kaplan, "Decoding the neural signatures of emotions expressed through sound," *NeuroImage*, vol. 174, pp. 1–10, 2018.

[17] V. Putkinen, S. Nazari-Farsani, K. Seppälä, T. Karjalainen, L. Sun, H. K. Karlsson, M. Hudson, T. T. Heikkilä, J. Hirvonen, and L. Nummenmaa, "Decoding music-evoked emotions in the auditory and motor cortex," *Cerebral Cortex*, vol. 31, no. 5, pp. 2549–2560, 2021.

[18] I. Daly, D. Williams, F. Hwang, A. Kirke, E. R. Miranda, and S. J. Nasuto, "Electroencephalography reflects the activity of sub-cortical brain regions during approach-withdrawal behaviour while listening to music," *Scientific Reports*, vol. 9, no. 1, p. 9415, 2019.

[19] S. Paquette, S. Takerkart, S. Saget, I. Peretz, and P. Belin, "Cross-classification of musical and vocal emotions in the auditory cortex," *Annals of the New York Academy of Sciences*, vol. 1423, no. 1, pp. 329–337, 2018.

[20] X. Cui, Y. Wu, J. Wu, Z. You, J. Xiahou, and M. Ouyang, "A review: Music-emotion recognition and analysis based on EEG signals," *Frontiers in Neuroinformatics*, vol. 16, p. 997282, 2023.

[21] D. S. Naser and G. Saha, "Influence of music liking on EEG based emotion recognition," *Biomedical Signal Processing and Control*, vol. 64, p. 102251, 2021.

[22] S. Hoefle, A. Engel, R. Basilio, V. Alluri, P. Toiviainen, M. Cagy, and J. Moll, "Identifying musical pieces from fMRI data using encoding and decoding models," *Scientific Reports*, vol. 8, no. 1, pp. 1–13, 2018.

[23] D. Sonawane, K. P. Miyapuram, B. Rs, and D. J. Lomas, "Guessthemusic: Song identification from electroencephalography response," in *Proceedings of the 3rd ACM India Joint International Conference on Data Science & Management of Data (8th ACM IKDD CODS & 26th COMAD)*, ser. CODS-COMAD '21, 2021, pp. 154–162.

[24] P. Pandey, G. Sharma, K. P. Miyapuram, R. Subramanian, and D. Lomas, "Music identification using brain responses to initial snippets," in *Proceedings in the 2022 IEEE International Conference on Acoustics, Speech and Signal Processing*, ser. ICASSP 2022, 2022, pp. 1246–1250.

[25] R. S. Schaefer, J. Farquhar, Y. Blokland, M. Sadakata, and P. Desain, "Name that tune: decoding music from the listening brain," *NeuroImage*, vol. 56, no. 2, pp. 843–849, 2011.

[26] D. Wu, C. Li, Y. Yin, C. Zhou, and D. Yao, "Music composition from the brain signal: Representing the mental state by music," *Computational Intelligence and Neuroscience*, vol. 2010, 2010.

[27] S. Stober, T. Prätzlich, and M. Müller, "Brain beats: Tempo extraction from EEG data." in *Proceedings of the 17th International Society for Music Information Retrieval Conference*, ser. ISMIR 2016, 2016, pp. 276–282.

[28] I. Sturm, M. Treder, D. Miklody, H. Purwins, S. Dähne, B. Blankertz, and G. Curio, "Extracting the neural representation of tone onsets for separate voices of ensemble music using multivariate EEG analysis." *Psychomusicology: Music, Mind, and Brain*, vol. 25, no. 4, p. 366, 2015.

[29] N. Gang, B. Kaneshiro, J. Berger, and J. P. Dmochowski, "Decoding neurally relevant musical features using canonical correlation analysis," in *Proceedings of the 18th International Society for Music Information Retrieval Conference*, ser. ISMIR 2017, 2017, pp. 131–138.

[30] I. Daly, "Neural decoding of music from the EEG," *Scientific Reports*, vol. 13, no. 1, pp. 1–17, 2023.

[31] L. May, A. R. Halpern, S. D. Paulsen, and M. A. Casey, "Imagined musical scale relationships decoded from auditory cortex," *Journal of Cognitive Neuroscience*, vol. 34, no. 8, pp. 1326–1339, 2022.

[32] S. Ntalampiras and I. Potamitis, "A statistical inference framework for understanding music-related brain activity," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 2, pp. 275–284, 2019.

[33] G. M. Di Liberto, G. Marion, and S. A. Shamma, "Accurate decoding of imagined and heard melodies," *Frontiers in Neuroscience*, vol. 15, p. 673401, 2021.

[34] S. Stober, A. Sternin, A. M. Owen, and J. A. Grahn, "Towards music imagery information retrieval: Introducing the OpenMIIR dataset of EEG recordings from music perception and imagination." in *Proceedings of the 16th International Society for Music Information Retreival Conference*, ser. ISMIR 2015, 2015, pp. 763–769.

[35] 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 21th International Society for Music Information Retrieval Conference*, ser. ISMIR 2020, 2020, pp. 566–573.

[36] 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.

[37] E. Manilow, P. Seetharman, and J. Salamon, "Open source tools & data for music source separation," 2020. [Online]. Available: https://source-separation.github.io/tutorial

[38] G. Cantisani, G. Trégoat, S. Essid, and G. Richard, "MAD-EEG: an EEG dataset for decoding auditory attention to a target instrument in polyphonic music," in *Proceedings of the Speech, Music and Mind (SMM), Satellite Workshop of Interspeech 2019*, 2019.

[39] G. Cantisani, S. Essid, and G. Richard, "Neuro-steered music source separation with eeg-based auditory attention decoding and contrastive-nmf," in *Proceedings of the 2021 IEEE International Conference on Acoustics, Speech and Signal Processing*, ser. ICASSP 2021, 2021, pp. 36–40.

[40] E. J. Allen, M. Moerel, A. Lage-Castellanos, F. De Martino, E. Formisano, and A. J. Oxenham, "Encoding of natural timbre dimensions in human auditory cortex," *NeuroImage*, vol. 166, pp. 60–70, 2018.

[41] V. Alluri, P. Toiviainen, I. P. Jääskeläinen, E. Glerean, M. Sams, and E. Brattico, "Large-scale brain networks emerge from dynamic processing of musical timbre, key and rhythm," *NeuroImage*, vol. 59, no. 4, pp. 3677–3689, 2012.

[42] M. Ogg, D. Moraczewski, S. E. Kuchinsky, and L. R. Slevc, "Separable neural representations of sound sources: Speaker identity and musical timbre," *NeuroImage*, vol. 191, pp. 116–126, 2019.

[43] 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*, ser. ICASSP 2023, 2023.

[44] O. Esteban, C. J. Markiewicz, R. W. Blair, C. A. Moodie, A. I. Isik, A. Erramuzpe, J. D. Kent, M. Goncalves, E. DuPre, M. Snyder, H. Oya, S. S. Ghosh, J. Wright, J. Durnez, R. A. Poldrack, and K. J. Gorgolewski, "fmriprep: a robust preprocessing pipeline for functional MRI," *Nature Methods*, vol. 16, no. 1, pp. 111–116, 2019.

[45] W. D. Penny, K. J. Friston, J. T. Ashburner, S. J. Kiebel, and T. E. Nichols, *Statistical parametric mapping: the analysis of functional brain images*. Elsevier, 2011.

[46] J. A. Mumford, T. Davis, and R. A. Poldrack, "The impact of study design on pattern estimation for single-trial multivariate pattern analysis," *NeuroImage*, vol. 103, pp. 130–138, 2014.

[47] M. F. Glasser, T. S. Coalson, E. C. Robinson, C. D. Hacker, J. Harwell, E. Yacoub, K. Ugurbil, J. Andersson, C. F. Beckmann, M. Jenkinson, S. M. Smith, and D. C. Van Essen, "A multi-modal parcellation of human cerebral cortex," *Nature*, vol. 536, no. 7615, pp. 171–178, 2016.

[48] A. Horn, "HCP-MMP1.0 projected on MNI2009a GM (volumetric) in NIfTI format," 2016. [Online]. Available: https://figshare.com/articles/dataset/HCP-MMP1_0_projected_on_MNI2009a_GM_volumetric_in_NIfTI_format/3501911

[49] A. Abraham, F. Pedregosa, M. Eickenberg, P. Gervais, A. Mueller, J. Kossaifi, A. Gramfort, B. Thirion, and G. Varoquaux, "Machine learning for neuroimaging with scikit-learn," *Frontiers in Neuroinformatics*, p. 14, 2014.

[50] S. Koelsch, "Neural substrates of processing syntax and semantics in music," *Current Opinion in Neurobiology*, vol. 15, no. 2, pp. 207–212, 2005.

[51] R. J. Zatorre, P. Belin, and V. B. Penhune, "Structure and function of auditory cortex: music and speech," *Trends in Cognitive Sciences*, vol. 6, no. 1, pp. 37–46, 2002.

[52] Y. Zhang, H. Ruan, Z. Yuan, H. Du, X. Gao, and J. Lu, "A learnable spatial mapping for decoding the directional focus of auditory attention using eeg," in *Proceedings of the 2023 IEEE International Conference on Acoustics, Speech and Signal Processing*, ser. ICASSP 2023, 2023.

[53] A. Floren, B. Naylor, R. Miikkulainen, and D. Ress, "Accurately decoding visual information from fMRI data obtained in a realistic virtual environment," *Frontiers in Human Neuroscience*, vol. 9, p. 327, 2015.

[54] T. Horikawa and Y. Kamitani, "Generic decoding of seen and imagined objects using hierarchical visual features," *Nature Communications*, vol. 8, no. 1, p. 15037, 2017.

[55] Y.-T. Wu, H.-Y. Chen, Y.-H. Liao, L.-W. Kuo, and C.-C. Lee, "Modeling perceivers neural-responses using lobe-dependent convolutional neural network to improve speech emotion recognition." in *Proceedings of the 18th Annual Conference of the International Speech Communication Association*, ser. INTERSPEECH 2017, 2017, pp. 3261–3265.

[56] 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*, 2022, pp. 11 976–11 986.

[57] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proceedings of the 7th International Conference on Learning Representations*, ser. ICLR 2019, 2019.

[58] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17, 2017, p. 4768–4777.

[59] E. J. Humphrey, S. Reddy, P. Seetharaman, A. Kumar, R. M. Bittner, A. Demetriou, S. Gulati, A. Jansson, T. Jehan, B. Lehner, A. Krupse, and L. Yang, "An introduction to signal processing for singing-voice analysis: High notes in the effort to automate the understanding of vocals in music," *IEEE Signal Processing Magazine*, vol. 36, no. 1, pp. 82–94, 2018.

[60] C. Gupta, H. Li, and M. Goto, "Deep learning approaches in topics of singing information processing," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 2422–2451, 2022.

[61] W. Trost, T. Ethofer, M. Zentner, and P. Vuilleumier, "Mapping aesthetic musical emotions in the brain," *Cerebral Cortex*, vol. 22, no. 12, pp. 2769–2783, 2012.

[62] S. Koelsch and S. Skouras, "Functional centrality of amygdala, striatum and hypothalamus in a "small-world" network underlying joy: An fmri study with music," *Human Brain Mapping*, vol. 35, no. 7, pp. 3485–3498, 2014.

[63] P. N. Juslin, "From everyday emotions to aesthetic emotions: Towards a unified theory of musical emotions," *Physics of Life Reviews*, vol. 10, no. 3, pp. 235–266, 2013.

[64] 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, 2007.

[65] C. L. Gordon, P. R. Cobb, and R. Balasubramaniam, "Recruitment of the motor system during music listening: An ale meta-analysis of fmri data," *PloS ONE*, vol. 13, no. 11, p. e0207213, 2018.

[66] J. K. Bizley and Y. E. Cohen, "The what, where and how of auditory-object perception," *Nature Reviews Neuroscience*, vol. 14, no. 10, pp. 693–707, 2013.

[67] V. K. M. Cheung and S. Sakamoto, "Separating uncertainty from surprise in auditory processing with neurocomputational models: Implications for music perception," *Journal of Neuroscience*, vol. 42, no. 29, pp. 5657–5659, 2022.

[68] S. K. Zieliński, P. Hardisty, C. Hummersone, and F. Rumsey, "Potential biases in MUSHRA listening tests," in *Proceedings of the 123rd Audio Engineering Society Convention*, ser. AES 123, 2007.

[69] A. Furnham, "Response bias, social desirability and dissimulation," *Personality and Individual Differences*, vol. 7, no. 3, pp. 385–400, 1986.

[70] G. Kalton and H. Schuman, "The effect of the question on survey responses: A review," *Journal of the Royal Statistical Society: Series A (General)*, vol. 145, no. 1, pp. 42–57, 1982.

DUAL ATTENTION-BASED MULTI-SCALE FEATURE FUSION APPROACH FOR DYNAMIC MUSIC EMOTION RECOGNITION

Liyue Zhang[1] **Xinyu Yang**[2] **Yichi Zhang**[2]
Jing Luo[2]
[1]The School of Software, Xi'an Jiaotong University , China
[2]The School of Computer Science and Technology, Xi'an Jiaotong University , China
{3121358019, datasonezyc, luojingl}@stu.xjtu.edu.cn, yxyphd@mail.xjtu.edu.cn

ABSTRACT

Music Emotion Recognition (MER) refers to automatically extracting emotional information from music and predicting its perceived emotions, and it has social and psychological applications. This paper proposes a Dual Attention-based Multi-scale Feature Fusion (DAMFF) method and a newly developed dataset named MER1101 for Dynamic Music Emotion Recognition (DMER). Specifically, multi-scale features are first extracted from the log Mel-spectrogram by multiple parallel convolutional blocks. Then, a Dual Attention Feature Fusion (DAFF) module is utilized to achieve multi-scale context fusion and capture emotion-critical features in both spatial and channel dimensions. Finally, a BiLSTM-based sequence learning model is employed for dynamic music emotion prediction. To enrich existing music emotion datasets, we developed a high-quality dataset, MER1101, which has a balanced emotional distribution, covering over 10 genres, at least four languages, and more than a thousand song snippets. We demonstrate the effectiveness of our proposed DAMFF approach on both the developed MER1101 dataset, as well as on the established DEAM2015 dataset. Compared with other models, our model achieves a higher Consistency Correlation Coefficient (CCC), and has strong predictive power in arousal with comparable results in valence.

1. INTRODUCTION

With the rising demand for music consumption and the explosive growth of music content, Music Emotion Recognition (MER) demonstrates its critical position in music understanding and applications. It has been widely used in personalized music recommendation [1], music therapy [2], music education [3], music generation [4], etc.

To portray human emotions, two main types of models were differentiated in the past [5]: discrete emotion model [6, 7] and dimensional emotion model [8–11]. The discrete emotion model describes human emotion as categorical adjectives, such as happiness, anger, sadness, joy, etc. However, limited words cannot adequately describe human emotions, different emotions are better described on a continuous scale than as a set of discrete values. In Russell's two-dimensional valence-arousal (V-A) emotional model [12], emotions are described as points on the plane that is spanned by the arousal and valence axes. This turns the problem of emotion prediction into a two-dimensional regression issue based on Russell's emotion model. This paper is focused on the study of Dynamic Music Emotion Recognition (DMER), which predicts the emotion of music using continuous V-A values at a short interval.

Among the existing studies, Long Short-Term Memory (LSTM) has received extensive attention in the DMER due to its superiority in sequence modeling [8, 13–15]. Convolutional Neural Network (CNN) is used to extract features in many fields. Researchers have recently focused on improving emotion recognition accuracy using a combination of CNN and Recurrent Neural Network (RNN) [9, 16–18]. However, LSTM-based models still use handcrafted features as input, and some widely used handcrafted feature operations will lose high-level features. The CNN-RNN-based model mainly uses a fixed-scale CNN. Due to its fixed receptive field, the learned CNN features are limited, and the emotional crucial features of different fields of view are not extracted. Moreover, various problems exist in existing music emotion datasets, which also hinder the progress of DMER.

This paper proposes a novel Dual Attention-based Multi-scale Feature Fusion (DAMFF) model and develops the music emotion dataset MER1101 for DMER. On the one hand, our model first utilizes multi-scale convolution to extract features at different temporal-frequency spans from the log Mel-spectrogram. Then, we propose a Dual Attention Feature Fusion (DAFF) module for fusing multi-scale context features from spatial and channel dimensions to enhance the expressive ability of CNN. Finally, the BiLSTM model processes these features and predicts V-A emotional labels. On the other hand, we develop a high-quality dataset named MER1101. Compared with the existing publicly available datasets in the MER domain, MER1101 contains 1101 music snippets from 16 genres with richer languages, more extensive size, and more balanced emotion label distribution. We evaluate our method using the MER1101 dataset and DEAM2015 [19] dataset.

© L. Zhang, X. Yang, Y. Zhang, J. Luo. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).
Attribution: L. Zhang, X. Yang, Y. Zhang, J. Luo, "Dual Attention-based Multi-scale Feature Fusion Approach for Dynamic Music Emotion Recognition", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

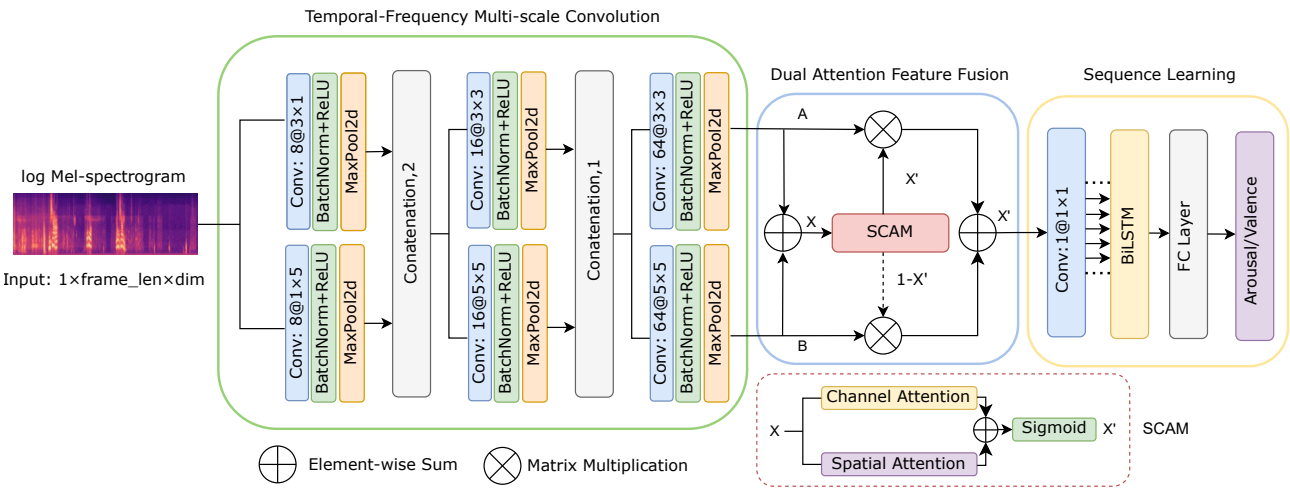


Figure 1. DAMFF model architecture. The input is 2D spectrogram. The architecture combines temporal-frequency multi-scale feature extraction, dual attention feature fusion, and sequence learning to achieve dynamic emotion prediction for music.

On the MER1101 dataset, we achieve a Consistency Correlation Coefficient (CCC) of 0.4223 for arousal and 0.1115 for valence. On the DEAM2015 dataset, we achieve a CCC of 0.4203 for arousal and 0.0151 for valence. Experimental results show our method outperforming a number of baseline and SOTA models in DMER, by means of an improved CCC metric.

2. RELATED WORK

Researchers have made many efforts in the DMER in the past few years. In the early days, RNN made a breakthrough in this field due to their advantages in sequence processing. In the "Emotion in Music" task at MediaEval from 2013 to 2015, LSTM-based methods achieved state-of-the-art performance [20]. Li *et al.* [8] pointed out that in music composition, performance, and annotation, the emotion in music is related to the previous and future contexts. Therefore, they chose Bidirectional LSTM (BiLSTM) as the regression model and proposed a multi-scale fusion method based on an Extreme Learning Machine (ELM) to improve the performance of the BiLSTM model. But the LSTM-based models mentioned above use suboptimal hand-crafted features as input, making it difficult to improve emotion recognition.

Later, researchers began to employ CNN for high-level invariant features extracted from raw music data [21–23]. Pons *et al.* [24] discussed how convolution filters with different shapes are suitable for specific musical concepts and experimentally proved that the size of CNN filters can be interpreted in both the temporal and frequency dimensions of the spectrogram. Researchers have combined CNN and RNN to improve the accuracy of emotion recognition, Malik *et al.* [16] proposed a two-dimensional V-A space continuous emotion prediction method composed of stacked convolution and recurrent neural network. Compared to using BiLSTM [15] only, this method achieved better results with fewer parameters; Dong *et al.* [9] replaced the connection between the input layer and the hidden layer of the RNN with a CNN to adaptively learn the sequential-information-included affect-salient features from the spectrogram; Zhang *et al.* [25] extracted MFCCs and Cochleagrams from raw music data as input features, and adopted an audio feature fusion method based on the combination of CNN and BiLSTM to predict the emotional V-A values in music. However, CNN-RNN-based models still have problems with limited convolutional receptive fields. For MER, due to the limited size of the convolution kernel, the convolution is mainly biased towards learning local information, which is insufficient for learning the correlation between the spatial and channel axes.

Various attention mechanisms are devised to solve the above problem in speech emotion recognition [21, 26, 27]. Guo *et al.* [26] proposed a representation learning method with spectral-temporal channel (STC) attention, which was integrated with CNN to improve representation learning ability; Zhang *et al.* [21] applied multi-scale region attention in deep convolutional neural networks to focus on emotional features at different granularities; Zhang *et al.* [27] implemented an attention layer on the arousal, valence, and dominance tasks and completed multi-task predictions to capture the contribution of different parts of each task. Nonetheless, the attention mechanism is currently not widely applied in the field of DMER.

In this paper, we propose a novel attention module, the Spatial Channel Attention Module (SCAM), which considers spatial and channel dimensions to capture the relative importance of features and integrates multi-scale convolutions for enhanced representation learning. We aim to build an attention mechanism that extracts salient information from multiple dimensions and can fuse contextual information.

J. S. Gómez-Cañón et al. [28] summarized existing MER datasets. But they have some problems, for example, some datasets have insufficient number of music, and some datasets have no dimension labels. After our com-

prehensive comparison, the three datasets CH818 [29], P-MEmo [30] and DEAM [19] are relatively suitable for the DMER task. However, all three datasets have some disadvantages. The songs in the CH818 dataset only contain Chinese pop songs and are not public, while PMEmo only Western pop songs; The annotators and annotating times of the training set and evaluation set in the DEAM2015 dataset are different, resulting in a discrepancy in performance [19]. To enrich existing musical emotion datasets, we develop a high-quality dataset, MER1101. MER1101 contains 1101 music snippets, which is better than most datasets in the MER domain in terms of genre, language, number of music, and has more balanced distributed emotion annotations.

3. METHODOLOGY

The proposed DMER processing method consists of three phases. Firstly, we build a Temporal-Frequency Multi-scale Convolution network using three different shapes of convolutional filters. Secondly, we propose a Dual Attention Feature Fusion network to focus more on the channel and spatial with important information and fuse multi-scale convolutional features in different dimensions. And finally, we employ BiLTSM, building a map from emotion-crucial features to emotional space. The specifics are as follows.

3.1 Temporal-Frequency Multi-scale Convolution

CNN has been proven effective at tackling various visual tasks [31, 32]. In vision tasks, the filter dimension has spatial meaning, and the audio spectrogram filter dimension corresponds to temporal and frequency [24]. We design a temporal-frequency multi-scale convolution module with three types of filters to capture various musical features. From the musical point of view, the temporal filter (*1-by-n*) can learn temporal dependence in music; the frequency filter (*m-by-1*) can learn pitch and timbre, and the square filter (*m-by-n*) can learn different musical features according to the size of *m* and *n*. As shown in Figure 1, we extract features through three layers of parallel convolutional blocks in the Temporal-Frequency Multi-scale Convolution module.

Firstly, we take the 30-second log Mel-spectrogram as input and perform distinct convolution operations on each 0.5-second segment to keep the individual properties at each moment. Secondly, the first layer introduces 3×1 and 1×5 filters to capture features along the temporal and frequency axes, and their outputs are concatenated along the time dimension. Finally, the concatenated results of the first layer are put into consecutive parallel convolutional layers with kernel sizes 3×3 and 5×5. The output of the second layer is concatenated along the channel dimension, while the output of the third layer is fed into a dual attention feature fusion module for feature fusion. After each convolutional layer, batch normalization [33], the ReLU function [34] and max pooling are applied.

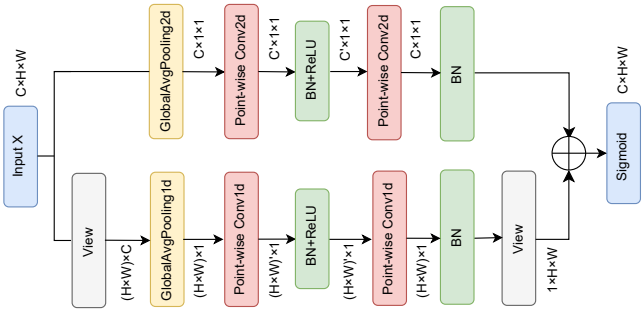


Figure 2. SCAM model architecture.

3.2 Dual Attention Feature Fusion

To further enhance the representation ability of CNN and capture the important information, we design a Dual Attention Feature Fusion (DAFF) module to focus more on the channel and spatial with important information for fusing multi-scale convolutional features. As shown in Figure 1, the DAFF module includes the Spatial Channel Attention Module (SCAM). By element-wise summing the outputs of 3×3 and 5×5 convolutions, we get a feature map $X \in R^{C\times H\times W}$ as input to SCAM, which is then fed into the spatial and channel attention modules, respectively. In Sections 3.2.1 and 3.2.2, we describe the proposed SCAM in detail.

3.2.1 Channel Attention Module

We convert a single channel into 64 channels through the Temporal-Frequency Multi-scale Convolution, strengthening the temporal correlation between distinct channels. In this case, we use the channel attention mechanism, which focuses on *what* the essential features are. While traditional attention mechanisms only focus on temporal structures, channel attention can learn the importance of different channels to deactivate features that do not contribute much to emotion. Figure 2 shows the channel attention module, similar to the Squeeze-and-Excitation block [35]. The module is mainly divided into two parts: squeeze and excitation operations. Specifically, given an input feature $X \in R^{C\times H\times W}$, we first use Global Average Pooling independently for each channel to aggregate spatial information and generate a channel attention map $C \in R^{C\times 1\times 1}$. Next, we perform an excitation operation using two point-wise convolutions to enable cross-channel interaction. Finally, the channel attention map $C \in R^{C\times 1\times 1}$ is obtained. In short, the channel attention map is calculated as follows:

$$C = \beta(Conv2d_2(\delta(\beta(Conv2d_1(Pool2d(X)))))) \quad (1)$$

where δ and β denote the ReLU function and batch normalization, respectively, and $Pool2d$ and $Conv2d$ represent the global average pooling2d and point-wise convolution2d, respectively.

3.2.2 Spatial Attention Module

We propose the spatial attention model, which exploits the spatial relationship between features to generate a spatial

attention map. Spatial attention focuses on *where* the important features are and supplements channel attention.

The spatial attention module obtains the spatial attention map in four phases. First, the input feature through view operation is converted into a spatial feature map $S' \in R^{(H\times W)\times C}$. Second, a global pooling operation is applied along the channel axis to compress the channels to obtain spatial-level features. Third, we use two point-wise convolutions to execute excitation operations and get feature weights at distinct positions. Finally, the resulting spatial attention map is translated into $S \in R^{1\times H\times W}$. In short, the spatial attention map is calculated as follows:

$$S = \beta(Conv1d_2(\delta(\beta(Conv1d_1(Pool1d(X)))))) \quad (2)$$

where δ and β denote the ReLU function and batch normalization, respectively, and $Pool1d$ and $Conv1d$ represent the global average pooling1d and point-wise convolution1d, respectively.

After that, we perform an element-wise sum operation on the output of the dual attention and through the sigmoid function to obtain a new attention weight map $X' \in R^{H\times W\times C}$.

$$X' = Sigmoid(S \oplus C) \quad (3)$$

3.2.3 Feature Fusion Strategy

In order to effectively aggregate multi-scale context information, we introduce the fusion strategy in [36], as shown by Dual Attention Feature Fusion in Figure 1. The output of SCAM is represented as X', $1-X'$ by the solid line and dotted line, respectively. Based on the SCAM, the multi-scale feature fusion can be expressed as:

$$Z = X' \otimes A + (1 - X') \otimes B \quad (4)$$

where A and B represent the outputs of 3×3 and 5×5 convolutions respectively, $Z \in R^{C\times H\times W}$ is the fused feature.

3.3 Sequence Learning

Through the DAFF module, we get emotion-crucial features from multi-scale convolutional features. After reducing dimension, the features of the entire 30s of music snippet are input into the Bidirectional LSTM (BiLSTM) for long-term sequence learning. Finally, the emotional features are mapped to the emotional space with the help of a fully connected layer.

4. EXPERIMENTS

4.1 Dataset

We conduct our experiments on the DEAM2015 [19] dataset and our newly developed dataset MER1101. The details of each dataset are given below.

DEAM: This dataset was developed in the "Emotion in Music" (EiM) task [37] of the MediaEval benchmark. We utilized the DEAM2015 dataset, with the training set consisting of 431 30-second samples and the evaluation set consisting of 58 full-length songs. This dataset is the most commonly used benchmark in dynamic music emotion recognition, but Cronbach's α of the evaluation set is 0.29±0.94 for valence, which is relatively low [19]. Furthermore, due to the different spatio-temporal environments and annotators of the emotion annotation process of the training set and the evaluation set [19], the performance derived from the training and evaluation set shows a non-negligible discrepancy, especially in the valence dimension.

MER1101 [1] **:** Similar with DEAM, MER1101 is also based on Russell's valence-arousal emotion model. It contains 1101 music snippets gathered from the internet, with each ranging in duration from 16.5 seconds to 125.5 seconds. The dataset has both discrete and dimensional labels. Every song in the dataset has been annotated by three music experts and ten college students. The annotators listened to the song once and annotated the emotional adjectives of the song. After they were familiar with the song, they listened to it twice and annotated the V-A values. Annotators were only paid the full fee after their work had been reviewed. Student-labeled Cronbach's α arousal is 0.6295 ± 0.3574, 0.5624 ± 0.3766 for the valence. Expert-labeled Cronbach's α arousal is 0.3556 ± 0.3442, 0.2420 ± 0.3148 for the valence.

Compared with other music datasets, the MER1101 dataset has the following four advantages: 1) The dataset contains more genres (16 genres), including pop, DJ dance, chinoiserie, electronic, hip-hop, etc.; 2) It contains richer language, meeting the ratio of nearly 5:3:1:1 for Chinese, English, Japanese and Korean, and other languages; 3) The samples in our dataset distribute more balanced in the emotional quadrants and there are no more than three songs by the same artist in each V-A quadrant; 4) The size of our dataset is relatively larger than the current music datasets.

Our dataset can be used for a variety of music tasks, such as music genre classification, music generation with emotion, music emotion recognition, *etc*.

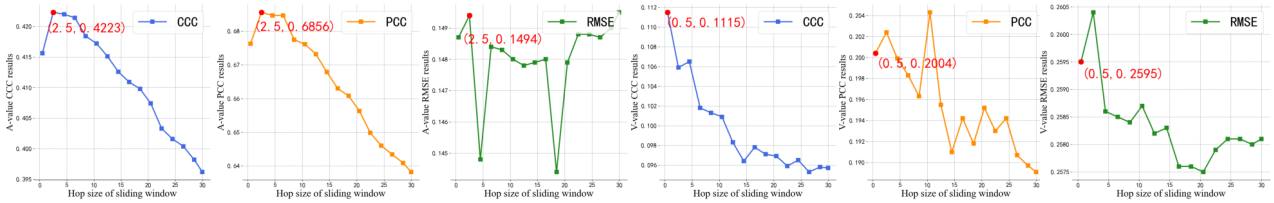
4.2 Evaluation Metrics

We use the Concordance Correlation Coefficient (CCC), Pearson Correlation Coefficient (PCC), and Root Mean Square Error (RMSE) as evaluation metrics. Each metric is computed by the ground-truth and predicted V-A values for each song and averaged across songs. The CCC combines the characteristics of PCC and RMSE to evaluate not only the trend of emotional changes but also the disparity between predictions and ground-truth. As a result, we consider CCC to be the most important evaluation metric.

4.3 Experimental Setup

Since DEAM2015 predefines the training and evaluation set configuration, we only describe the dataset division for MER1101 here. Firstly, we choose 925 songs lasting more than 30 seconds from the MER1101 dataset and randomly split them into a training set (80% of the data) and an evaluation set (20% of the data). Then, we split each song in

[1] See https://ismir-2023.github.io/MER1101/ for details.



(a) A-value CCC (b) A-value PCC (c) A-value RMSE (d) V-value CCC (e) V-value PCC (f) V-value RMSE

Figure 3. The CCC, PCC, and RMSE of arousal and valence with different hop sizes on the MER1101 dataset.

the training set into 30-second segments and kept complete songs for the evaluation set. The final training set contains 1526 30-second music snippets, and the evaluation set contains 185 complete songs. The DEAM dataset uses the official training and evaluation sets. To obtain a more accurate comparison and minimize accidental errors, we use 5-fold cross-validation on both datasets.

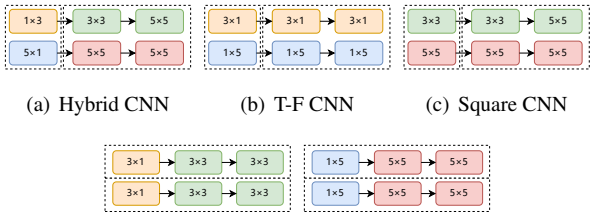
The log Mel-spectrogram is extracted using librosa [38], where the Mel band is 128, the sampling rate is 44100Hz, and the window size and hop size are 60 ms and 10 ms, respectively. The size of the convolution kernel is shown in Figure 1. We utilize the Adam optimizer for training, with learning rate of 0.0003, training epoch of 100, and batch size of 32. To prevent overfitting, we adopt the early stopping strategy. In addition, we use CCC and RMSE as loss functions for arousal and valence, respectively.

4.4 Experimental Results

4.4.1 Hop Size Selection of Sliding Window

For the MER1101 dataset, we train the model with music snippets of fixed duration, while the durations of the music snippets are variable during the test. Thus, we could not directly predict the emotion of the whole music. We propose a sliding window-based testing scheme to address this issue and ensure the continuity of the predicted V-A curves. During testing, we utilize the window size of T seconds and the hop size of t seconds. Each T second of audio in the window is input to the model, and the corresponding T seconds V-A curves are predicted. The first window takes the prediction result of T seconds, and each subsequent window only takes the result of the last t seconds.

We investigate the impact of hop size on the results of music emotion recognition on the MER1101 dataset. We set the window size to 30s, the same as the training set sample duration. Figure 3 shows the experimental results, CCC and PCC change significantly and show a downward trend with increasing hop size, and the change in RMSE is not obvious. We observe that with the increase of the hop size, the emotion prediction effect decreased significantly, demonstrating that the shorter hop size performs better. During listening to music, the user's emotion at a certain moment is an accumulation of previous music content. Therefore, providing the model with as much context as possible benefits emotion recognition. A shorter hop size can provide more context for the model to predict the current musical mood. In the experiments on the MER1101 dataset below, we adopt hop sizes of 2.5s and 0.5s for arousal and valence, respectively.



(a) Hybrid CNN (b) T-F CNN (c) Square CNN

(d) T-S CNN (e) F-S CNN

Figure 4. Five CNN architectures.

Model	Arousal			Valence		
	CCC ↑	PCC* ↑	RMSE* ↓	CCC ↑	PCC ↑	RMSE ↓
Hybrid CNN	**0.4223**	0.6856	0.1494	**0.1115**	**0.2004**	0.2595
T-F CNN	0.4120	0.6787	0.1478	0.0846	0.1363	0.2684
Square CNN	0.4130	0.6894	0.1439	0.0732	0.1343	0.2703
T-S CNN	0.4090	0.6881	0.1458	0.1085	0.1959	**0.2542**
F-S CNN	0.4150	0.6804	0.1562	0.1046	0.1640	0.2800

* The result of the significance test (Student's t test) show that there is no significant difference between the results of this metric.

Table 1. Experimental results of different CNN architectures on the MER1101 dataset.

4.4.2 Impact of CNN filters

In this section, we compare the influence of different CNN architectures on the experimental results of the MER1101 dataset. In this paper, we adapt three types of convolution: temporal filters (*1-by-n*), frequency filters (*m-by-1*), and squared filters (*m-by-n*). Convolution filters of different shapes have different musical concepts. We combined them into five architectures. In Figure 4(a), the CNN architecture used here is a "Hybrid CNN" architecture. Figure 4(b) uses the temporal filters and frequency filters, and we call it the "T-F CNN" architecture. Figure 4(c) only uses a square filter, so we call it "Square CNN" architecture. Figure 4(d) and Figure 4(e) are referred to as "T-S CNN" and "F-S CNN", respectively. The experimental results are shown in Table 1, which show that the "Hybrid CNN" architecture has better expressiveness on the DMER

Model	MER1101 dataset						DEAM2015 dataset					
	Arousal			Valence			Arousal			Valence		
	CCC ↑	PCC ↑	RMSE ↓	CCC ↑	PCC↑	RMSE ↓	CCC ↑	PCC ↑	RMSE ↓	CCC ↑	PCC↑	RMSE ↓
CRNN [16]	0.2798	0.5177	0.1625	0.0573	0.1033	0.2721	0.3488	0.5885	**0.2197**	0.0053	-0.0292	0.3542
BCRSN [9]	0.1741	0.3770	0.3063	0.0660	-0.0647	0.4143	0.3168	0.5148	0.2397	0.0125	-0.0171	**0.2914**
DNN [17]	0.0529	0.0903	0.2372	0.0118	0.0017	0.2734	0.2757	0.4282	0.2483	0.0075	0.0031	0.3353
MCRNN [18]	0.0564	0.0918	0.2401	0.0155	0.0028	0.2752	0.2700	0.4396	0.2428	0.0137	0.0126	0.3135
DAMFF	**0.4223**	**0.6856**	**0.1494**	**0.1115**	**0.2004**	**0.2595**	**0.4203**	**0.6866**	0.2401	**0.0151**	**0.0366**	0.3403

Table 2. Compared with the existing results.

task. It is shown that extracting them simultaneously is beneficial to obtain music emotion information from different perspectives, and the PCC and RMSE changes of Arousal are not significant.

4.4.3 Comparison with the Existing Models

We compare the DAMFF to other DMER methods [9, 16–18] published in recent years. They differ from us in that [18] takes DEAM2014 [39] as the dataset, which consists of 744 songs. [16–18] take RMSE as the evaluation metrics, and [9] translates numerical-type V-A values to binary representation and independently predict emotion for each 0.5s.

In this paper, we reproduce the models mentioned above on the DEAM2015 and MER1101 datasets. All models' performance is evaluated with the same experimental configurations, i.e., the same dataset, evaluation metrics, and metric calculation method. Table 2 shows the results of the experiments. On the MER1101 dataset, our model is superior to the others in all three metrics. On the DEAM2015 dataset, our model shows powerful recognition ability for arousal, but the valence slightly outperforms the previous models, which may stem from the less consistent valence annotations [15]. We believe predicted valence values on the DEAM2015 dataset are relatively incapable of evaluating DMER since the predicted CCC value number in valence driven from all models is near zero. Experiments show that our model can perform well in emotion recognition on different datasets, especially in the arousal dimension. Overall, valence values are more impoverished in both datasets than arousal values, indicating that predicting valence is more challenging. This is also consistent with the conclusions of most works.

Model	Arousal			Valence		
	CCC ↑	PCC* ↑	RMSE* ↓	CCC ↑	PCC ↑	RMSE ↓
DAMFF	**0.4223**	0.6856	0.1494	**0.1115**	**0.2004**	**0.2595**
w/o Fusion Strategy	0.4097	0.6869	0.1563	0.1074	0.1722	0.2707
w/o Channel Attention	0.4061	0.6740	0.1494	0.1071	0.1904	0.2650
w/o Spatial Attention	0.4177	0.6819	0.1518	0.1009	0.1874	0.2720
w/o DAFF	0.3982	0.6813	0.1562	0.0977	0.1693	0.2670

* The result of the significance test (Student's t test) show that there is no significant difference between the results of this metric.

Table 3. Ablation experiments of arousal and valence on the MER1101 dataset.

4.4.4 Ablation Study

To investigate the role of various modules, we constructed four ablation modules. Among them, "w/o Fusion Strategy" directly inputs the result of the SCAM module into BiLSTM, which explores the role of fusion strategy. In addition, the influence of dual attention is studied using "w/o Channel Attention", "w/o Spatial Attention", and "w/o DAFF". Table 3 shows the experimental results on the MER1011 datasets. The results show that: 1) the non-linear fusion strategy of the attention mechanism better aggregates the multi-scale context and performs better; 2) the attention mechanism increases the weights of emotional features, which is helpful for emotion recognition. At the same time, dual attention is better than single attention, indicating that spatial and channel attention mechanisms learn and emphasize *what* and *where* affect-salient features, effectively improving CNN features. In summary, we conclude that fusing multi-scale convolutional features from spatial and channel dimensions is more conducive to capturing key emotional features, which is more evident on the CCC metric.

5. CONCLUSION

This paper proposes a novel Dual Attention-based Multi-scale Feature Fusion (DAMFF) network, which extracts multi-scale convolutional features from spectrograms and exploits the dual-attention mechanism to capture important channel and spatial information. The network adopts the fusion mechanism that aggregates multi-scale context information, effectively improving CNN features' expressive ability. The music emotion dataset MER1101 we developed contains 1101 music audio with 16 genres, 5 languages and a balanced distribution of emotion labels. Experimental results show that our model outperforms the comparison methods on the CCC metric on both MER1101 and DEAM2015 datasets. Furthermore, our model has substantial prediction capabilities in terms of arousal and comparable results in terms of valence.

The prediction of the valence dimension is still challenging in DMER. In the future, we will focus on developing more effective techniques, such as pre-training audio features for improving the recognition performance of valence.

6. REFERENCES

[1] S. M. Florence and M. Uma, "Emotional detection and music recommendation system based on user facial expression," in *IOP Conference Series: Materials Science and Engineering*, vol. 912, no. 6. IOP Publishing, 2020, p. 062007.

[2] G. A. Dingle, P. J. Kelly, L. M. Flynn, and F. A. Baker, "The influence of music on emotions and cravings in clients in addiction treatment: A study of two clinical samples," *The Arts in Psychotherapy*, vol. 45, pp. 18–25, 2015.

[3] T. Xia, Z. Li *et al.*, "Behavioral training of high-functioning autistic children by music education of occupational therapy," *Occupational Therapy International*, vol. 2022, 2022.

[4] 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.

[5] X. Yang, Y. Dong, and J. Li, "Review of data features-based music emotion recognition methods," *Multimedia systems*, vol. 24, pp. 365–389, 2018.

[6] K. Trohidis, G. Tsoumakas, G. Kalliris, I. P. Vlahavas *et al.*, "Multi-label classification of music into emotions." in *International Conference on Music Information Retrieval (ISMIR)*, vol. 8, 2008, pp. 325–330.

[7] J.-H. Su, T.-P. Hong, Y.-H. Hsieh, and S.-M. Li, "Effective music emotion recognition by segment-based progressive learning," in *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2020, pp. 3072–3076.

[8] X. Li, H. Xianyu, J. Tian, W. Chen *et al.*, "A deep bidirectional long short-term memory based multi-scale approach for music dynamic emotion prediction," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 544–548.

[9] Y. Dong, X. Yang, X. Zhao, and J. Li, "Bidirectional convolutional recurrent sparse network (BCRSN): an efficient model for music emotion recognition," *IEEE Transactions on Multimedia*, vol. 21, no. 12, pp. 3150–3163, 2019.

[10] S. Chaki, P. Doshi, S. Bhattacharya, and P. Patnaik, "Explaining perceived emotion predictions in music: An attentive approach." in *International Conference on Music Information Retrieval (ISMIR)*, 2020, pp. 150–156.

[11] Z. Huang, S. Ji, Z. Hu, C. Cai, J. Luo, and X. Yang, "ADFF: Attention Based Deep Feature Fusion Approach for Music Emotion Recognition," in *Interspeech 2022, 23rd Annual Conference of the International Speech Communication Association, Incheon, Korea, 18-22 September 2022*, 2022, pp. 4152–4156.

[12] J. A. Russell, "A circumplex model of affect." *Journal of personality and social psychology*, vol. 39, no. 6, p. 1161, 1980.

[13] F. Weninger, F. Eyben, and B. Schuller, "On-line continuous-time music mood regression with deep recurrent neural networks," in *2014 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2014, pp. 5412–5416.

[14] Y. Ma, X. Li, M. Xu, J. Jia, and L. Cai, "Multi-scale context based attention for dynamic music emotion prediction," in *Proceedings of the 25th ACM international conference on Multimedia*, 2017, pp. 1443–1450.

[15] X. Li, J. Tian, M. Xu, Y. Ning, and L. Cai, "Dblstm-based multi-scale fusion for dynamic emotion prediction in music," in *2016 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2016, pp. 1–6.

[16] M. Malik, S. Adavanne, K. Drossos, T. Virtanen, D. Ticha, and R. Jarina, "Stacked convolutional and recurrent neural networks for music emotion recognition," in *Proceedings. 14th Sound Music Comput. Conf.*, 2017, pp. 208–213.

[17] R. Orjesek, R. Jarina, M. Chmulik, and M. Kuba, "Dnn based music emotion recognition from raw audio signal," in *2019 29th International Conference Radioelektronika (RADIOELEKTRONIKA)*. IEEE, 2019, pp. 1–4.

[18] N. He and S. Ferguson, "Multi-view neural networks for raw audio-based music emotion recognition," in *2020 IEEE International Symposium on Multimedia (ISM)*. IEEE, 2020, pp. 168–172.

[19] A. Aljanaki, Y.-H. Yang, and M. Soleymani, "Emotion in music task at mediaeval 2015," in *MediaEval*, 2015.

[20] Aljanaki, Anna and Yang, Yi-Hsuan and Soleymani, Mohammad, "Developing a benchmark for emotional analysis of music," *PloS one*, vol. 12, no. 3, p. e0173392, 2017.

[21] M. Xu, F. Zhang, X. Cui, and W. Zhang, "Speech emotion recognition with multiscale area attention and data augmentation," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6319–6323.

[22] J. Liu, Z. Liu, L. Wang, L. Guo, and J. Dang, "Speech emotion recognition with local-global aware deep representation learning," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7174–7178.

[23] W. Zhu and X. Li, "Speech emotion recognition with global-aware fusion on multi-scale feature representation," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 6437–6441.

[24] J. Pons, T. Lidy, and X. Serra, "Experimenting with musically motivated convolutional neural networks," in *2016 14th international workshop on content-based multimedia indexing (CBMI)*. IEEE, 2016, pp. 1–6.

[25] C. Zhang, J. Yu, and Z. Chen, "Music emotion recognition based on combination of multiple features and neural network," in *2021 IEEE 4th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, vol. 4. IEEE, 2021, pp. 1461–1465.

[26] L. Guo, L. Wang, C. Xu, J. Dang, E. S. Chng, and H. Li, "Representation learning with spectro-temporal-channel attention for speech emotion recognition," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6304–6308.

[27] Z. Zhang, B. Wu, and B. Schuller, "Attention-augmented end-to-end multi-task learning for emotion prediction from speech," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6705–6709.

[28] 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.

[29] X. Hu and Y.-H. Yang, "The mood of chinese pop music: Representation and recognition," *Journal of the Association for Information Science and Technology*, vol. 68, no. 8, pp. 1899–1910, 2017.

[30] K. Zhang, H. Zhang, S. Li, C. Yang, and L. Sun, "The pmemo dataset for music emotion recognition," in *Proceedings of the 2018 acm on international conference on multimedia retrieval*, 2018, pp. 135–142.

[31] G. Li and Y. Yu, "Visual saliency based on multiscale deep features," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 5455–5463.

[32] M. Zhao, G. Cao, X. Huang, and L. Yang, "Hybrid transformer-cnn for real image denoising," *IEEE Signal Processing Letters*, vol. 29, pp. 1252–1256, 2022.

[33] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*. pmlr, 2015, pp. 448–456.

[34] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

[35] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[36] 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.

[37] "Mediaeval benchmarking initiative for multimedia evaluation," http://www.multimediaeval.org/.

[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 *Proceedings of the 14th python in science conference*, vol. 8. Citeseer, 2015, pp. 18–25.

[39] 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.

AUTOMATIC PIANO TRANSCRIPTION WITH HIERARCHICAL FREQUENCY-TIME TRANSFORMER

Keisuke Toyama[1] **Taketo Akama**[2] **Yukara Ikemiya**[1]
Yuhta Takida[1] **Wei-Hsiang Liao**[1] **Yuki Mitsufuji**[1]
[1] Sony Group Corporation, Japan
[2] Sony Computer Science Laboratories, Japan
keisuke.toyama@sony.com

ABSTRACT

Taking long-term spectral and temporal dependencies into account is essential for automatic piano transcription. This is especially helpful when determining the precise onset and offset for each note in the polyphonic piano content. In this case, we may rely on the capability of self-attention mechanism in Transformers to capture these long-term dependencies in the frequency and time axes. In this work, we propose *hFT-Transformer*, which is an automatic music transcription method that uses a two-level hierarchical frequency-time Transformer architecture. The first hierarchy includes a convolutional block in the time axis, a Transformer encoder in the frequency axis, and a Transformer decoder that converts the dimension in the frequency axis. The output is then fed into the second hierarchy which consists of another Transformer encoder in the time axis. We evaluated our method with the widely used MAPS and MAESTRO v3.0.0 datasets, and it demonstrated state-of-the-art performance on all the F1-scores of the metrics among *Frame*, *Note*, *Note with Offset*, and *Note with Offset and Velocity* estimations.

1. INTRODUCTION

Automatic music transcription (AMT) is to convert music signals into symbolic representations such as piano rolls, Musical Instrument Digital Interface (MIDI), and musical scores [1]. AMT is important for music information retrieval (MIR), its result is useful for symbolic music composition, chord progression recognition, score alignment, etc. Following the conventional methods [1–15], we estimate the frame-level metric and note-level metrics as follows: (1) *Frame*: the activation of quantized pitches in each time-processing frame, (2) *Note*: the onset time of each note, (3) *Note with Offset*: the onset and offset time of each note, and (4) *Note with Offset and Velocity*: the onset, offset time, and the loudness of each note.

© K. Toyama, T. Akama, Y. Ikemiya, Y. Takida, W. H. Liao, and Y. Mitsufuji. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** K. Toyama, T. Akama, Y. Ikemiya, Y. Takida, W. H. Liao, and Y. Mitsufuji, "Automatic Piano Transcription with Hierarchical Frequency-Time Transformer", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

For automatic piano transcription, it is important to analyze several harmonic structures that spread in a wide range of frequencies, since piano excerpts are usually polyphonic. Convolutional neural network (CNN)-based methods have been used to aggregate harmonic structures as acoustic features. Most conventional methods apply multi-layer convolutional blocks to extend the receptive field in the frequency axis. However, the blocks often include pooling or striding to downsample the features in the frequency axis. Such a downsampling process may reduce the frequency resolution [6]. It is worth mentioning, many of these methods use 2-D convolutions, which means the convolution is simultaneously applied in the frequency and time axes. The convolution in the time axis works as a pre-emphasis filter to model the temporal changes of the input signals.

Up to now, recurrent neural networks (RNNs), such as gated recurrent unit (GRU) [16] and long short-term memory (LSTM) [17], are popular for analyzing the temporal sequences of acoustic features. However, recently some of the works start to use Transformer [18], which is a powerful tool for analyzing sequences, in AMT tasks. Ou et al. [2] applied a Transformer encoder along the time axis and suggested that using Transformer improves velocity estimation. Hawthorne et al. [3] used a Transformer encoder-decoder as a sequence-to-sequence model for estimating a sequence of note events from another sequence of input audio spectrograms. Their method outperformed other methods using GRUs or LSTMs. Lu et al. [19] proposed a method called SpecTNT to apply Transformer encoders in both frequency and time axes and reached state-of-the-art performance for various MIR tasks such as music tagging, vocal melody extraction, and chord recognition. This suggests that such a combination of encoders helps in characterizing the broad-scale dependency in the frequency and time axes. However, SpecTNT aggregates spectral features into one token, and the process in its temporal Transformer encoder is not independent in the frequency axis. This inspires us to incorporate Transformer encoders in the frequency and time axes and make the spectral information available for the temporal Transformer encoder.

In addition, we usually divide the input signal into chunks since the entire sequence is often too long to be

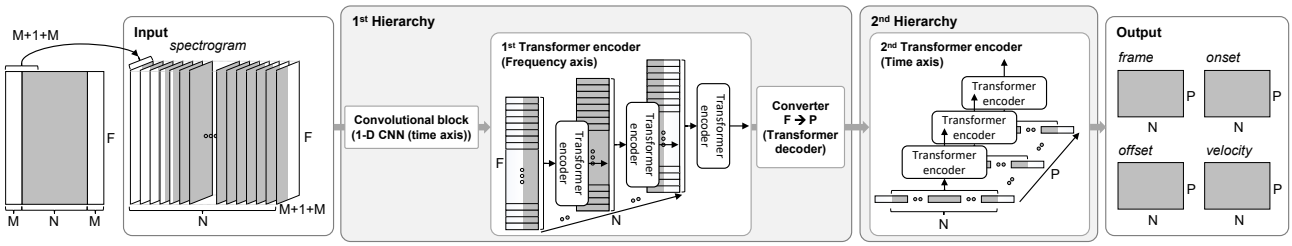


Figure 1. hFT-Transformer (N: number of frames in each processing chunk, M: length of margin, F: number of frequency bins, P: number of pitches)

dealt at once. However, this raises a problem that the estimated onset and offset accuracy fluctuates depending on the relative position in the processing chunk. In our observation, the accuracy tends to be worse at both ends of the processing chunk. This motivates us to incorporate extra techniques during the inference time to boost the performance.

In summary, we propose *hFT-Transformer*, an automatic piano transcription method that uses a two-level hierarchical frequency-time Transformer architecture. Its workflow is shown in Figure 1. The first hierarchy consists of a one-dimensional (1-D) convolutional block in the time axis, a Transformer encoder in the frequency axis, and a Transformer decoder in the frequency axis. The second hierarchy consists of another Transformer encoder in the time axis. In particular, the Transformer decoder at the end of the first hierarchy converts the dimension in the frequency axis from the number of frequency bins to the number of pitches (88 for piano). Regarding the issue of the location dependent accuracy fluctuation in the processing chunks, we propose a technique which halves the stride length at inference time. It uses only the result of the central part of processing chunks, which will improve overall accuracy. Finally, in Section 4, we show that our method outperforms other piano transcription methods in terms of F1 scores for all the four metrics.

A `PyTorch` implementation of our method is available here [1] .

2. RELATED WORK

Neural networks, such as CNNs, RNNs, generative adversarial networks (GANs) [20], and Transformers have been dominant for AMT. Since Sigtia et al. [4] proposed the first method to use a CNN to tackle AMT, CNNs have been widely used for the methods of analyzing the spectral dependency of the input spectrogram [2, 6–10, 12–15]. However, it is difficult for CNNs to directly capture the harmonic structure of the input sound in a wide range of frequencies, as convolutions are used to capture features in a local area. Wei et al. [5] proposed a method of using harmonic constant-Q transform (CQT) for capturing the harmonic structure of piano sounds. They first applied a 3-Dimensional CQT, then applied multiple dilated convolutions with different dilation rates to the output of CQT. Because the dilation rates are designed to capture the harmonics, the performance of *Frame* and *Note* accuracy reached state-of-the-art. However, the dilation rates are designed specifically for piano. Thus, the method is not easy to adapt to other instruments.

For analysis of time dependency, Kong et al. [6] proposed a method that uses GRUs. Howthorner et al. [7], Kwon et al. [8], Cheuk et al. [9], and Wei et al. [5] proposed methods that use bi-directional LSTMs for analysis. Ou et al. [2] used a Transformer encoder to replace the GRUs in Kong et al.'s method [6], and showed the effectiveness of the Transformer. Usually, the note onset and offset are estimated in each frequency and time-processing frame grid, then paired as a note for note-level transcription by post-processing algorithms such as [6]. However, compared to heuristically designed algorithms, end-to-end data-driven methods are often preferred. For example, Keltz et al. [10] applied a seven-state hidden Markov model (HMM) for the sequence of attack, decay, sustain, and release to achieve note-level transcription. Kwon et al. [8] proposed a method of characterizing the output of LSTM as a five-state statement (onset, offset, re-onset, activate, and inactivate). Hawthorne et al. [3] proposed a method of estimating a sequence of note events, such as note pitch, velocity, and time, from another sequence of input audio spectrograms using a Transformer encoder-decoder. This method performs well in multiple instruments with the same model [11]. Yan et al. [12] proposed a note-wise transcription method for estimating the interval between onset and offset. This method shows state-of-the-art performance in estimating *Note with Offset* and *Note with Offset and Velocity*. However, the performance in estimating *Frame* and *Note* is worse than that of Wei et al.'s method [5].

3. METHOD

3.1 Configuration

Our proposed method aims to transcribe N frames of the input spectrogram into N frames of the output piano rolls (*frame*, *onset*, *offset*, and *velocity*) as shown in Figure 1, where N is the number of frames in each processing chunk. Each input frame is composed of a log-mel spectrogram having size (F, $M + 1 + M$), where F is the number of frequency bins, and M is the size of the forward margin and that of the backward margin. To obtain

[1] `https://github.com/sony/hFT-Transformer`

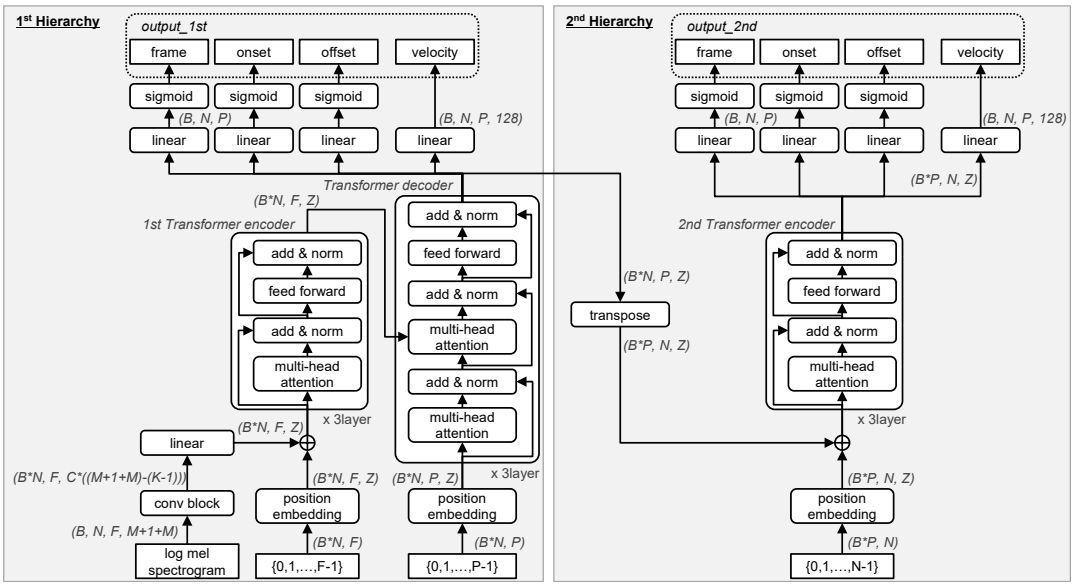


Figure 2. Model architecture of hFT-Transformer

the log-mel spectrogram, we first downmix the input waveform into one channel and resample them to 16 kHz. Then, the resampled waveform is transformed into a mel spectrogram with `transforms.MelSpectrogram` class in the `Torchaudio` library [21]. For the transformation, we use *hann* window, setting the window size as 2048, fast-Fourier-transform size as 2048, F as 256, padding mode as *constant*, and hop-size as 16 ms. The magnitude of the mel spectrogram is then compressed with a log function.

3.2 Model Architecture and Loss Functions

The model architecture of our proposed method is shown in Figure 2. We first apply a convolutional block to the input log-mel spectrogram, the size of which is (B, N, F, $M+1+M$) where B is the batch size. In the convolutional block, we apply a 1-D convolution in the $M + 1 + M$ dimension. After this process, the data are embedded with a linear module.

The embedded vector is then processed with the first Transformer encoder in the frequency axis. The self-attention is processed to analyze the dependency between spectral features. The positional information is designated as [0, 1, ..., $F - 1$]. These positional values are then embedded with a trainable embedding. These are processed in the frequency axis only, thus completely independent to the time axis (N dimension).

Next, we convert the frequency dimension from F to the number of pitches (P). A Transformer decoder with cross-attention is used as the converter. The Transformer decoder calculates the cross-attention between the output vectors of the first Transformer encoder and another trainable positional embedding made from [0, 1, ..., $P-1$]. The decoded vectors are then converted to the outputs of the first hierarchy with a linear module and a sigmoid function (hereafter, we call these outputs *output_1st*).

Regarding the loss calculation for the outputs, *frame*, *onset*, and *offset* are calculated with binary cross-entropy, and *velocity* is calculated with 128-category cross-entropy. The losses can be summarized as the following equations:

$$L_{\mathrm{bce}}^{<\mathrm{m}>} = \sum_{n=0}^{N-1} \sum_{p=0}^{P-1} l_{\mathrm{bce}}(y_{n,p}^{<\mathrm{m}>}, \hat{y}_{n,p}^{<\mathrm{m}>}), \tag{1}$$

$$L_{\mathrm{cce}}^{\mathrm{velocity}} = \sum_{n=0}^{N-1} \sum_{p=0}^{P-1} l_{\mathrm{cce}}(y_{n,p}^{\mathrm{velocity}}, \hat{y}_{n,p}^{\mathrm{velocity}}), \tag{2}$$

$$L = L_{\mathrm{bce}}^{\mathrm{frame}} + L_{\mathrm{bce}}^{\mathrm{onset}} + L_{\mathrm{bce}}^{\mathrm{offset}} + L_{\mathrm{cce}}^{\mathrm{velocity}}, \tag{3}$$

where $<\mathrm{m}>$ is the placeholder for each output (*frame*, *onset*, and *offset*), l_{bce} and l_{cce} denote the loss function for binary cross-entropy and categorical cross-entropy, respectively, and y and $\hat{y}$ denote the ground truth and predicted values of each output (*frame*, *onset*, *offset*, and *velocity*), respectively. Although it is intuitive to apply the mean squared error (MSE) for *velocity*, we found that using the categorical cross-entropy yields much better performance than the MSE from a preliminary experiment.

Finally, the output of the converter is processed with another Transformer encoder in the time axis. The self-attention is used to analyze the temporal dependency of features in each time-processing frame. A third positional embedding made from [0, 1, ..., $N-1$] is used here. Then, similar to the first hierarchy, the outputs of the second hierarchy are obtained through a linear module and a sigmoid function. We call these outputs of the second hierarchy as *output_2nd* hereafter. The losses for the *output_2nd* are evaluated in the same way as those for *output_1st*. These losses are summed with the coefficients α_{1st} and α_{2nd} as follows:

$$L_{\mathrm{all}} = \alpha_{\mathrm{1st}} L_{\mathrm{1st}} + \alpha_{\mathrm{2nd}} L_{\mathrm{2nd}}. \tag{4}$$

Although both outputs are used for computing losses during training, only *output_2nd* is used in inference. As Chen et al. [22] suggested that the performance of their method of calculating multiple losses outperformed the method

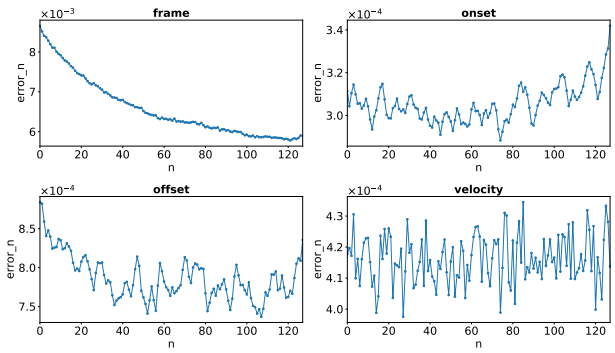


Figure 3. Estimation error (Eqn (5)) on location in each time-processing frame

that uses single loss only, it hints us that utilizing both *output_1st* and *output_2nd* in training has the potential to achieve better performance.

3.3 Inference Stride

As mentioned in Section 1, chunk-based processing is required because the input length is limited due to system limitations, such as memory size and acceptable processing delay. We found that the estimation error tends to increase at certain part within each processing chunk. This can be demonstrated by evaluating the error for each instance of time n within the chunks:

$$error_n^{<\mathrm{m}>} = \frac{1}{IP}\sum_{i=0}^{I-1}\sum_{p=0}^{P-1}(y_{i,n,p}^{<\mathrm{m}>} - \hat{y}_{i,n,p}^{<\mathrm{m}>})^2, \quad (5)$$

where $<\mathrm{m}>$ is the placeholder for each output (*frame*, *onset*, *offset*, and *velocity*), and I is the number of processing chunks over the test set. The result using our proposed model trained using the MAESTRO training set (described in Section 4) is shown in Figure 3. Here, the error $error_n^{<\mathrm{m}>}$ is calculated using the MAESTRO test set. In the figure, we observe a monotonic decrease for *frame* and a similar but much weaker trend for *onset* and *offset*. However, for *velocity*, no such trend can be observed. This hints us to use only the middle portion of a processing chunk as the output to reduce the error rate. We call this as the half-stride strategy, since a 50% overlap is required for processing chunks, as shown in Figure 4 (B).

4. EXPERIMENTS

4.1 Datasets

We use two well-known piano datasets for the evaluation. The MAPS dataset [23] consists of CD-quality recordings and corresponding annotations of isolated notes, chords, and complete piano pieces. We use the full musical pieces and the train/validation/test split as stated in [4, 7]. The number of recordings and the total duration in hours in each split are 139/71/60 and 8.3/4.4/5.5, respectively. The MAESTRO v3.0.0 dataset [13] includes about 200 hours of paired audio and MIDI recordings from ten years of the International Piano-e-Competition. We used the

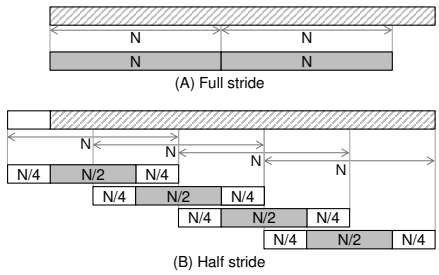


Figure 4. Inference stride: (A) full stride, (B) half stride

train/validation/test split configuration as provided. In each split, the number of recordings and total duration in hours are 962/137/177 and 159.2/19.4/20.0, respectively. For both datasets, the MIDI data have been collected by Yamaha Disklaviers concert-quality acoustic grand pianos integrated with a high-precision MIDI capture and playback system.

4.2 Model Configuration

Regarding our model architecture depicted in Figure 2, we set N as 128, M as 32, F as 256, P as 88, the CNN channels (C) as 4, size of the CNN kernel (K) as 5, and embedding vector size (Z) as 256. For the Transformers, we set the feed-forward network vector size as 512, the number of heads as 4, and the number of layers as 3. For training, we used the following settings: a batch size of 8, learning rate of 0.0001 with Adam optimizer [24], dropout rate of 0.1, and clip norm of 1.0. `ReduceLROnPlateu` in `PyTorch` is used for learning rate scheduling with default parameters. We set $\alpha_{1\mathrm{st}}$ and $\alpha_{2\mathrm{nd}}$ as 1.0, which were derived from a preliminary experiment (see Section 4.6).

We trained our models for 50 epochs on MAPS dataset and 20 epochs for MAESTRO dataset using one NVIDIA A100 GPU. It took roughly 140 minutes and 43.5 hours to train one epoch with our model for MAPS and MAESTRO, respectively. The best model is determined by choosing the one with the highest F1 score in the validation stage.

In order to obtain high-resolution ground truth for *onset* and *offset*, we followed the method in Kong et al. [6]. We set J, the hyper-parameter to control the sharpness of the targets, to 3. Also, the label of *velocity* is set only when an *onset* is present. We set the threshold as 0.5, which means if the *onset* is smaller than 0.5, the *velocity* is set as 0.

4.3 Inference

At inference time, we use *output_2nd* as the final output. We set the threshold for *frame* as 0.5. For note-wise events (*onset*, *offset*, and *velocity*), the outputs in each pitch-frame grid are converted to a set containing note-wise onset, offset, and velocity following Kong et al.'s *Algorithm 1* [6] in five steps shown below:

Step 1. onset detection: find a local maximum in *onset* with a value at least 0.5. Then calculate the precise onset time using the values of the adjacent three frames [6].

Step 2. velocity: If an onset is detected in Step 1, extract the *velocity* value at the frame. If the value is zero, then

Method	half stride	Params	Frame			Note			Note w/ Offset			Note w/ Offset&Velocity		
			P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)
Onsets&Frames [7]		26M	88.53	70.89	78.30	84.24	80.67	82.29	51.32	49.31	50.22	35.52	30.80	35.59
ADSR [10]		0.3M	**90.73**	67.85	77.16	**90.15**	74.78	81.38	61.93	51.66	56.08	-	-	-
hFT-Transformer		5.5M	83.36	82.00	82.67	86.63	83.75	85.07	67.18	65.06	66.03	48.75	47.21	47.92
hFT-Transformer	✓	5.5M	83.68	**82.11**	**82.89**	86.72	**83.81**	**85.14**	**67.51**	**65.36**	**66.34**	**49.05**	**47.48**	**48.20**

Table 1. Evaluation results on MAPS test dataset (P: precision, R: recall, **bold**: best score, underline: second best score)

Method	half stride	Params	Frame			Note			Note w/ Offset			Note w/ Offset&Velocity		
			P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)
Seq2Seq [3]		54M	-	-	-	-	-	96.01	-	-	83.94	-	-	82.75
HPT-T [2]		-	-	-	90.09	97.88	**96.72**	96.77	84.13	82.31	83.20	82.85	81.07	81.90
Semi-CRFs [12]		9M	**93.79**	88.36	90.75	98.69	93.96	96.11	90.79	86.46	88.42	89.78	85.51	87.44
HPPNet-sp [5]		1.2M	92.79	93.59	93.15	98.45	95.95	97.18	84.88	82.76	83.80	83.29	81.24	82.24
hFT-Transformer		5.5M	92.62	93.43	93.02	99.62	95.41	97.43	92.32	88.48	90.32	91.21	87.44	89.25
hFT-Transformer	✓	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**

Table 2. Evaluation results on MAESTRO v3.0.0 test dataset

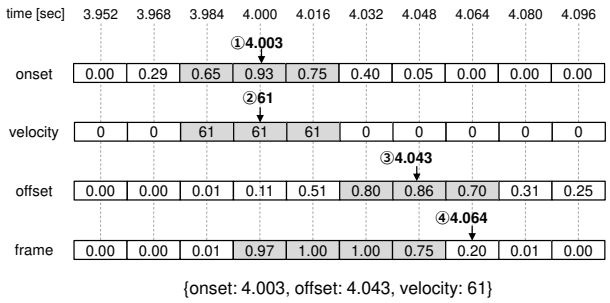


Figure 5. Example of conversion from grid-wise values to note-wise values

discard both onset and velocity at this frame.

Step 3. offset detection with *offset*: find a local maximum in *offset* with a value at least 0.5. Then calculate the precise offset time using the values of the adjacent three frames [6].

Step 4. offset detection with *frame*: choose the frame that is nearest to the detected onset which has a *frame* value below 0.5.

Step 5. offset decision: choose the smaller value between the results of Step 3 and 4.

An example is shown in Figure 5. The *onset* is 4.003, and the *velocity* is 61. For *offset*, the direct estimation from *offset* is 4.043, and that estimated via *frame* is 4.064. Thus, we choose 4.043 as *offset*. Finally, we obtain a note with {*onset*: 4.003, *offset*: 4.043, *velocity*: 61} in the output.

4.4 Metrics

We evaluate the performance of our proposed method with frame-level metrics (*Frame*) and note-level metrics (*Note*, *Note with Offset*, and *Note with Offset & Velocity*) with the standard precision, recall, and F1 scores. We calculated these scores using `mir_eval` library [25] with its default settings. The scores were calculated per recording, and the mean of these per-recording scores was presented as the final metric for a given collection of pieces, as explained in Hawthorne et al. [7].

4.5 Results

Tables 1 and 2 show the scores on the test sets of MAPS and MAESTRO datasets. The numbers of parameters in these Tables are referred from [5, 10]. For the MAPS dataset, our proposed method outperformed the other methods in F1 score for all metrics. For the MAESTRO dataset, our proposed method outperformed the other methods in F1 score for *Note*, *Note with Offset*, and *Note with Offset & Velocity*. Furthermore, our method with the half-stride strategy which is mentioned in 3.3 outperformed other methods in all metrics. In contrast, the two state-of-the-art methods for MAESTRO, which are Semi-CRFs [12] and HPPNet-sp [5], performed well only on a subset of the metrics.

The results suggest that the proposed two-level hierarchical frequency-time Transformer structure is promising for AMT.

4.6 Ablation Study

To investigate the effectiveness of each module in our proposed method, we trained various combinations of those modules using the MAPS training set and evaluated them using the MAPS validation set. The variations are shown in Table 3. In this study, we call our proposed method *1-F-D-T*, which means it consists of the *1*-D convolution block, the first Transformer encoder in the *F*requency axis, the Transformer *D*ecoder, and the second Transformer encoder in the *T*ime axis. Table 4 shows evaluation results for each variation.

Second Transformer encoder in time axis. To verify the effectiveness of the second Transformer encoder, we compared the 1-F-D-T and the model without the second Transformer encoder (1-F-D-N). For the 1-F-D-N model, we use *output_1st* in both training and inference stages as the final output. The result indicates that the second Transformer encoder improved *Note with Offset* performance, in

Model	1st-Hierarchy			2nd-Hierarchy		Output
	Convolutional block	1st Transformer encoder	Converter	2nd Transformer encoder		
1-F-D-T†	1-D (time axis)	Frequency axis	Transformer Decoder	Time axis		output_2nd
1-F-D-N	1-D (time axis)	Frequency axis	Transformer Decoder	n/a		output_1st
2-F-D-T	2-D	Frequency axis	Transformer Decoder	Time axis		output_2nd
1-F-L-T	1-D (time axis)	Frequency axis	Linear	Time axis		output_2nd

Table 3. Model variations for ablation study (†: the proposed method, hFT-Transformer)

Model	Params	Frame			Note			Note w/ Offset			Note w/ Offset&Velocity		
		P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)
1-F-D-T†	5.5M	93.61	**88.71**	**91.09**	98.81	**94.81**	**96.72**	**86.18**	**82.81**	**84.42**	**77.47**	**74.55**	**75.95**
1-F-D-N	3.9M	92.85	87.49	90.09	99.01	93.24	95.95	82.67	78.06	80.23	73.89	69.90	71.78
2-F-D-T	6.1M	75.49	61.08	67.52	97.03	19.68	31.10	64.07	13.28	20.88	42.11	8.57	13.50
1-F-L-T	3.4M	**93.71**	88.42	90.99	**99.11**	92.90	95.79	85.77	80.56	82.98	71.66	67.32	69.34

Table 4. Evaluation results of ablation study on MAPS validation dataset

which the F1 score is 84.42 for 1-F-D-T and 80.23 for 1-F-D-N. This shows the effectiveness of the second Transformer encoder as it provides an extra pass to model the temporal dependency of acoustic features, which is presumably helpful in offset estimation.

Compelxity of convolutional block. To investigate how the complexity of the convolutional block affects the AMT performance, we compared the 1-F-D-T model and the model that replaces the 1-D convolutional block with a 2-D convolutional block (2-F-D-T). Surprisingly, the result shows that the performance of the 2-F-D-T model is significantly worse than that of the 1-F-D-T model. This is probably because the two modules working on the spectral dependency do not cohere with each other. The 2-D convolutional block may over aggregate the spectral information thus resulting into an effectively lower frequency resolution. Then, the Transformer encoder can only evaluate the spectral dependency over an over-simplified feature space, causing the performance degradation.

Converter. We used a Transformer decoder to convert the dimension in the frequency axis from F to P. In contrast, almost all of the existing methods used a linear module to achieve this. We compared the performance of the 1-F-D-T model to a model with the Transfomer decoder replaced with a linear converter (1-F-L-T). The result indicates that the 1-F-D-T model outperformed the 1-F-L-T model in F1 score for all four metrics. Especially, the difference in *Note with Offset and Velocity* is large (75.95 for the 1-F-D-T model and 69.34 for the 1-F-L-T model in F1 score). This suggests that using a Transformer decoder as converter is an effective way of improving the performance, although the side effect is the increase of model size.

We also investigated how the coefficients for the loss functions, $\alpha_{1\mathrm{st}}$ and $\alpha_{2\mathrm{nd}}$ in Eqn (4), affect the performance. We investigated six pairs of coefficients of loss functions ($\alpha_{1\mathrm{st}}$, $\alpha_{2\mathrm{nd}}$) in Eqn (4), i.e., (1.8, 0.2), (1.4, 0.6), (1.0, 1.0), (0.6, 1.4), (0.2, 1.8), and (0.0, 2.0), for the 1-F-D-T model. Figure 6 shows the F1 scores of *frame*, *onset*, *offset*, and *velocity* evaluated on the MAPS validation set in each epoch. These results indicate that the (1.0, 1.0) pair

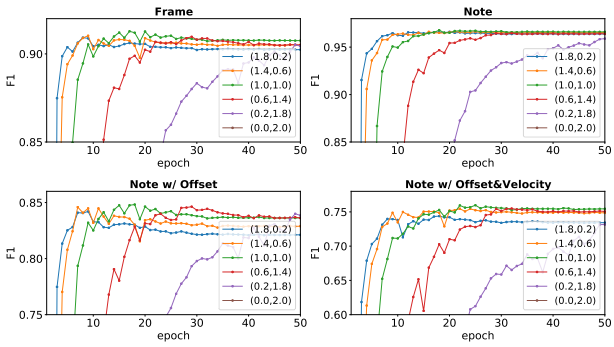


Figure 6. Performance of 1-F-D-T model trained with six-pairs of coefficients of loss functions

yields the best score. It also shows that the training converges faster when $\alpha_{1\mathrm{st}}$ is larger than $\alpha_{2\mathrm{nd}}$. Importantly, if we omit the *output_1st*, which is the case when training with the pair (0.0, 2.0), the training loss did not decrease much. Therefore, the F1 score stays around 0% and thus cannot be seen in Figure 6. This suggests that it is crucial to use both losses, *output_1st* and *output_2nd* in our proposed method.

5. CONCLUSION

In this work, we proposed *hFT-Transformer*, an automatic piano transcription method that uses a two-level hierarchical frequency-time Transformer architecture. The first hierarchy consists of a 1-D convolutional block in the time axis, a Transformer encoder and a Transformer decoder in the frequency axis, and the second hierarchy consists of a Transformer encoder in the time axis. The experiment result based on two well-known piano datasets, MAPS and MAESTRO, revealed that our two-level hierarchical architecture works effectively and outperformed other state-of-the-art methods in F1 score for frame-level and note-level transcription metrics. For future work, we would like to extend our method to other instruments and multi-instrument settings.

6. ACKNOWLEDGMENTS

We would like to thank Giorgio Fabbro and Stefan Uhlich for their valuable comments while preparing this manuscript. We are grateful to Kin Wai Cheuk for his dedicated support in preparing our github repository.

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, 2019.

[2] L. Ou, Z. Guo, E. Benetos, J. Han, and Y. Hang, "Exploring transformer's potential on automatic piano transcription," in *Proc. of the 47th Int. Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 776–780.

[3] C. Hawthorne, I. Simon, R. Swavely, E. Manilow, and J. Engel, "Sequence-to-sequence piano transcription with transformers," in *Proc. of the 22th Int. Society for Music Information Retrieval Conf.*, 2021, pp. 246–253.

[4] 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.

[5] W. Wei, P. Li, Y. Yu, and W. Li, "Hppnet: Modeling the harmonic structure and pitch invariance in piano transcription," in *Proc. of the 23rd Int. Society for Music Information Retrieval Conf.*, 2022, pp. 709–716.

[6] Q. Kong, B. Li, X. Song, Y. Wan, and Y. Wang, "High-resolution piano transcription with pedals by regressing onsets and offsets times," *IEEE/ACM Transactions on Audio Speech and Language Processing*, vol. 29, pp. 3707–3717, 2021.

[7] 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. of the 19th Int. Society for Music Information Retrieval Conf.*, 2018, pp. 50–57.

[8] T. Kwon, D. Jeong, and J. Nam, "Polyphonic piano transcription using autoregressive multi-state note model," in *Proc. of the 21st Int. Society for Music Information Retrieval Conf.*, 2020, pp. 454–460.

[9] 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 *Proc. of the 25th International Conference on Pattern Recognition (ICPR)*, 2020, pp. 9091–9098.

[10] R. Kelz, S. Böck, and G. Widmer, "Deep polyphonic adsr piano note transcription," in *Proc. of the 44th Int. Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 246–250.

[11] J. Gardner, I. Simon, E. Manilow, and C. H. J. Engel, "Mt3: Multi-task multitrack music transcription," in *Proc. of the Int. Conference on Learning Representations (ICLR)*, 2022.

[12] Y. Yan, F. Cwitkowitz, and Z. Duan, "Skipping the frame-level: Event-based piano transcription with neural semi-crfs," in *Proc. of the 36th Int. Conference on Neural Information Processing Systems (NeurIPS)*, 2021.

[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," in *Proc. of the 7th International Conference on Learning Representations (ICLR)*, 2019.

[14] R. Kelz, M. Dorfer, F. Korzeniowski, S. Böck, A. Arzt, and G. Widmer, "On the potential of simple framewise approaches to piano transcription," in *Proc. of the 17th Int. Society for Music Information Retrieval Conf.*, 2016, pp. 475–481.

[15] J. W. Kim and J. P. Bello, "Adversarial learning for improved onsets and frames music transcription," in *Proc. of the 20th Int. Society for Music Information Retrieval Conf.*, 2019, pp. 670–677.

[16] C. Kyunghyun, van Merrienboer Bart, G. Caglar, B. Dzmitry, B. Fethi, S. Holger, and B. Yoshua, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1724–1734.

[17] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, pp. 1735–1780, 1997.

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. olosukhin, "Attention is all you need," in *Proc. of the 31st Conference on Neural Information Processing Systems (NeurIPS)*, 2017, pp. 6000–6010.

[19] W.-T. Lu, J.-C. Wang, M. Won, K. Choi, and X. Song, "Spectnt: A time-frequency transformer for music audio," in *Proc. of the 22nd Int. Society for Music Information Retrieval Conf.*, 2021, pp. 396–403.

[20] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, vol. 27, 2014.

[21] Y.-Y. Yang, M. Hira, Z. Ni, A. Chourdia, A. Astafurov, C. Chen, C.-F. Yeh, C. Puhrsch, 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-Bélair, and Y. Shi, "Torchaudio: Building blocks for audio and speech processing," *arXiv preprint arXiv:2110.15018*, 2021.

[22] Y.-H. Chen, W.-Y. Hsiao, T.-K. Hsieh, J.-S. R. Jang, and Y.-H. Yang, "Towards automatic transcription of polyphonic electric guitar music: A new dataset and a multi-loss transformer model," in *Proc. of the 47th Int. Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 786–790.

[23] V. Emiya, R. Badeau, and B. David, "Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle," *IEEE ACM Transactions on Audio Speech and Language Processing*, vol. 18, no. 6, pp. 1643–1654, 2009.

[24] D. P. Kingm and J. L. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2015.

[25] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nietro, D. Liang, and D. Ellis, "mir_eval: A transparent implementation of common mir metrics," in *Proc. of the 15th Int. Society for Music Information Retrieval Conf.*, 2014, pp. 367–372.

HIGH-RESOLUTION VIOLIN TRANSCRIPTION USING WEAK LABELS

Nazif Can Tamer[♭] **Yigitcan Özer**[♯] **Meinard Müller**[♯] **Xavier Serra**[♭]
[♭] Music Technology Group, Universitat Pompeu Fabra, Barcelona, Spain
[♯] International Audio Laboratories Erlangen, Germany
nazifcan.tamer@upf.edu, yigitcan.oezer@audiolabs-erlangen.de,
meinard.mueller@audiolabs-erlangen.de, xavier.serra@upf.edu

ABSTRACT

A descriptive transcription of a violin performance requires detecting not only the notes but also the fine-grained pitch variations, such as vibrato. Most existing deep learning methods for music transcription do not capture these variations and often need frame-level annotations, which are scarce for the violin. In this paper, we propose a novel method for high-resolution violin transcription that can leverage piece-level weak labels for training. Our conformer-based model works on the raw audio waveform and transcribes violin notes and their corresponding pitch deviations with 5.8 ms frame resolution and 10-cent frequency resolution. We demonstrate that our method (1) outperforms generic systems in the proxy tasks of violin transcription and pitch estimation, and (2) can automatically generate new training labels by aligning its feature representations with unseen scores. We share our model along with 34 hours of score-aligned solo violin performance dataset, notably including the 24 Paganini Caprices.

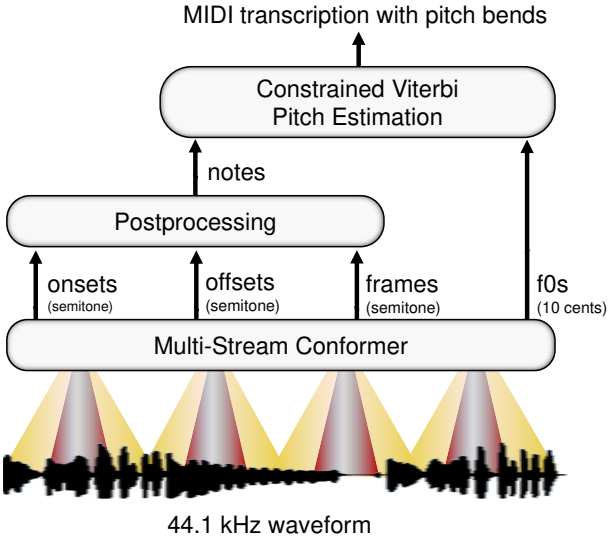


Figure 1: Our method transcribes violin recordings sampled with 44.1 kHz waveform into MIDI with a 5.8 ms time- and 10-cent frequency-resolution pitch bends.

1. INTRODUCTION

Automatic music transcription (AMT) is a core task in Music Information Retrieval that aims to convert a musical performance into some form of symbolic notation. While general-purpose AMT systems have recently seen substantial progress with deep learning [1–5], instrument-specific systems usually perform better, e.g., for piano [6–9], vocals [10, 11], guitar [12–14], and drums [15–17]. Despite the prominence of the violin in Western classical music and other traditions, a specialized high-precision violin transcription system that applies the recent advances in deep learning does not exist. In this paper, we aim to transcribe violin performances into a descriptive music notation [18]. As opposed to a prescriptive transcription, whose aim would be to produce an easily understandable score from which a musician can perform according to stylistic conventions of Western classical music writing, a descriptive transcription has an analytical purpose, aiming at notating high-precision pitch modulations along the notes.

Most typical AMT systems employ audio-to-MIDI transcription where each note event is represented with semitone resolution in the 12-tone equal temperament (12-TET). However, cognitive studies show that even the Western classical violinists heavily deviate from the 12-TET in favor of Pythagorean tuning and just intonation [19, 20]. Furthermore, the violin also plays a central role in many other traditions that do not employ the Western 12-TET [21]. Considering playing styles such as the vibrato and glissando that involve pitch modulations, a higher frequency resolution than the conventional 12-TET is required for violin transcription. An important step towards transcription outside the 12-TET was introduced by Bittner et al. [1] with an instrument-agnostic AMT system, which employs MIDI pitch bends to represent performances with 33-cent frequency resolution. However, adapting their approach to violin transcription remains to be a challenge since 33-cent frequency resolution is still too high compared to a violinist's intonation precision [20].

A further main challenge in violin transcription is the lack of frame-level annotated training data. To cope with the absence of frame-level annotations, Weiß and

© Nazif Can Tamer, Yigitcan Özer, Meinard Müller, Xavier Serra. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Nazif Can Tamer, Yigitcan Özer, Meinard Müller, Xavier Serra, "High-Resolution Violin Transcription using Weak Labels", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.*, Milan, Italy, 2023.

Peeters [22] employ sequence-level targets and a variant of the Connectionist Temporal Classification (CTC) loss for multipitch estimation. However, this strategy is sensitive to the segment duration (stable until segment lengths of 60 seconds) and, therefore, still requires some form of weak alignment. While some works explore data augmentation for frame-level supervised models through additional unlabeled [4] or pseudo-labeled [5] data, recent AMT methods are mostly trained using frame-level annotations [1, 3, 8, 9]. In some cases, obtaining such annotations is feasible through electronic music instruments, e.g., Disklavier. For example, the MAESTRO [23] dataset, with 200 hours of virtuoso piano performances and respective note labels captured with 3 ms frame resolution, enabled significant improvements for piano transcription.

In case electronic music instruments are unavailable, a common approach for obtaining automatic frame-level annotations is employing audio-to-score alignment (ASA), which found application in score following [24, 25]. ASA itself is not a technology developed for creating training datasets for AMT systems, and it has been reported that inaccurately aligned datasets may even worsen the result [2]. The intertwined nature of ASA and transcription can also be viewed from another aspect. For example, Kwon et al. [26] showed that frame and onset features of an AMT system work as robust feature representations for ASA. To our knowledge, the only deep-learning-based transcription system that integrates ASA into AMT is the recent work by Maman and Bermano [2], which utilizes ASA with chroma representations obtained from AMT frames.

As the main contribution of this paper, we propose a novel AMT system specifically tailored for descriptive violin transcription [1] regarding two crucial aspects: 1) We represent pitch deviations such as vibrato, glissando, or intonation choice by incorporating fine-grained pitch representations into the transcription. While borrowing our note postprocessing system and the MIDI pitch bend representations from Bittner et al. [1], we build a conformer-based model that works on the raw audio waveform and further improves the pitch bend estimation through note-constrained Viterbi pitch tracking. 2) We acquire frame-level annotations for violin transcription by considering simultaneous transcription and alignment in a joint framework, similar to the work by Maman and Bermano [2]. Following the findings from the music synchronization literature, we also incorporate activation-function-based features in the alignment [27, 28].

In order to benchmark our descriptive violin transcription method, we consider the proxy tasks of transcription and pitch estimation and compare our model with general-purpose baselines. As a side contribution, we also release a 34-hour dataset of solo violin recordings, with automatically aligned MIDIs and note-constrained multi-f0 tracks obtained using our descriptive violin transcription system.

The remainder of this paper is organized as follows: in Section 2, we introduce our MUlti-Stream Conformer (MUSC) model for AMT that processes an audio waveform into four musical representations. In Section 3, we describe our strategy for learning without frame-level annotations. In Section 4, we introduce how we simultaneously annotate a novel violin transcription dataset while training our model. In Section 5, we compare our descriptive violin transcription model against general-purpose transcription and pitch estimation baselines. Finally, we conclude in Section 6 with prospects on future work.

[1] https://github.com/MTG/violin-transcription/

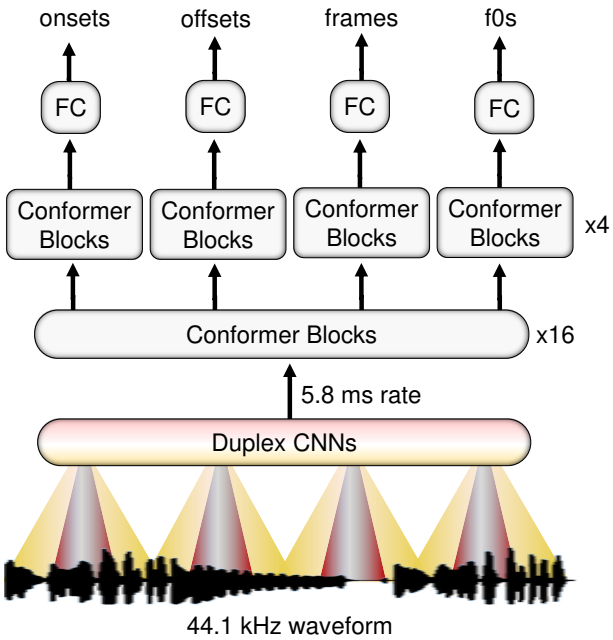


Figure 2: The Multi-Stream Conformer architecture converts raw audio sampled with 44.1 kHz into four feature representations with a frame rate of 5.8 ms.

2. MULTI-STREAM CONFORMER

We propose a MUlti-Stream Conformer (MUSC) that processes the raw audio waveform into four streams that estimate onset, offset, semitone-level pitch frames (denoted as frames as in the AMT literature), and high-resolution f0 frames as shown in Figure 2. The raw audio waveform sampled with 44.1 kHz is converted into 256-dimensional features with a hop size of 5.8 ms through duplex CNNs. Then, these features pass through the Conformer blocks to estimate the four representations. The resulting representations can be either used for MIDI transcription with pitch bends as in Figure 1, or for frame-level dataset annotation for training (see Section 3).

2.1 Duplex CNNs

We borrow the basic CNN structure from the first two layers of the CREPE [29] pitch estimator, except for zero padding. We remove the zero padding in the convolutional layers so that the duplex CNNs can access to the information at the borders of the window with varying receptive fields. With the raw audio in 44.1 kHz as the input, the duplex CNNs independently summarize the waveform into 128-dimensional frames with a hop length of 5.8 ms.

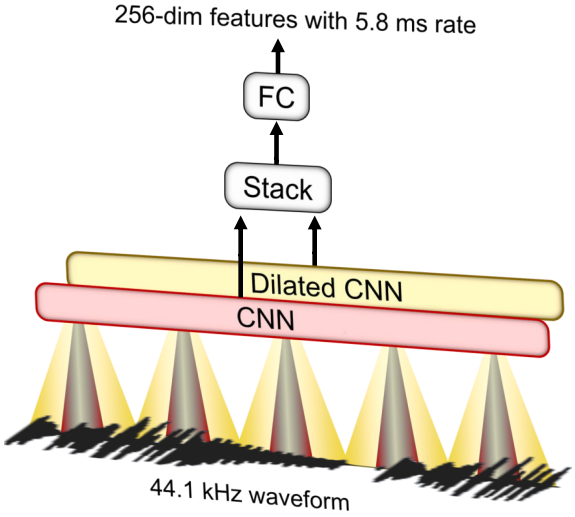


Figure 3: A closer look at the Duplex CNNs.

The standard CNN (shown in red in Figure 3) analyzes the frame with the CREPE configuration, resulting in a receptive field of 26 ms. The dilated CNN (depicted in yellow within Figure 3) incorporates double the number of dilations and strides per layer, ultimately leading to a receptive field of 118 ms. Thanks to the dilations and strides, the sampling rate for the dilated CNN is subsequently reduced to 22.05 kHz, and 11 kHz. Thus, it effectively analyzes a smoother version of the signal. The 128-dimensional outputs of the individual CNNs are then stacked into a 256-dimensional representation and pass through a simple fully-connected layer before the main Conformer stream.

2.2 Conformer Blocks

Due to the direct analogy between music transcription and speech recognition, we adopt the Conformer [30], a state-of-the-art automatic speech recognition (ASR) model, as the base block of MUSC. We directly employ conformer blocks from the Conformer encoder (M version) as described by Gulati et al. [30], i.e., with four attention heads, a depthwise convolution size of 32, and an encoder dimension of 256. For the main stream, we repeat the conformer blocks 16 times as in Conformer (M). Then, we employ separate conformer blocks for each of the onset, offset, frame, and f0 streams with four conformer blocks per representation. The total number of conformer blocks we utilize in the multi-stream conformer architecture is 32.

2.3 Feature Representations

Our method is based on transforming weak labels into frame-level features that are used both as training targets and alignment features. The feature representations encompass the violin pitch range from $F\sharp 3$ to $E8$, i.e., 58 bins for the onsets, offsets, and note frames, which work on semitone resolution, and 580 bins for the f0s, which work on 10-cent resolution. More precisely, we use a fixed sequence duration of three seconds and convert the audio waveform into 512×58 dimensional onset, offset, and (note) frames, and 512×580 dimensional f0 frames.

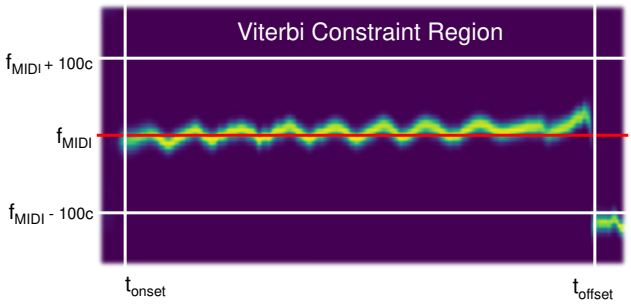


Figure 4: Constraint region for the Viterbi pitch tracking.

We train the model to predict strong onset, offset, and frame labels that are generated from iterative score alignments. We employ Gaussian label smoothing for onset, offset, and f0 features. For the onsets and offsets, we smooth the feature representations with a standard deviation of 4 ms. Following Kim et al. [29], we also blur the f0 features with a 12-cent standard deviation.

Note that the high-precision f0 features are not included in the score, hence cannot be inferred from the alignment. For the f0 features, we train the model to predict pseudo-labels generated by the TAPE model [31] in the first iteration. Then, we use our model's predictions as pseudo f0 labels. The polyphonic multipitch information are also encoded in the f0 representations. We employ constrained Viterbi pitch estimation (see Section 2.5) for generating pseudo-f0 labels for the polyphonic segments.

2.4 Note postprocessing

In the original Conformer paper [30], which is designed for ASR, the output of the encoder is proceeded by a decoder that uses an external language model to generate the word sequence. A natural adoption of this strategy to our scenario would require onsets, offsets, and frames to be fed into a language model that is specialized in the violin repertoire. However, employing a decoder is not viable since violin repertoire remains a low-resource language, and training decoders with such limited data is prone to overfitting. Instead, we experiment with postprocessing techniques from open-source AMT libraries and adopt the one [2] from Bittner et al. [1]. We leave improving the postprocessing stage as an open question for further studies.

2.5 Constrained Viterbi Pitch Estimation

Previous studies have shown that score information [32] and the continuity principle of pitch perception [33] can be used for refining the f0 estimation. We apply continuity constraints within note sections to detect the pitch bends with higher accuracy. First, we define the constraint region on the f0 matrix from the note onset, offset, and 200 cents around the note frequency as shown in Figure 4. We calculate the Viterbi path within the note boundaries by utilizing the constraint region as observation probabilities and f0 transition probability matrix $\mathbf{S} \in \mathbb{R}^{21\times 21}$ covering the

[2] https://github.com/spotify/basic-pitch/blob/main/basic_pitch/note_creation.py

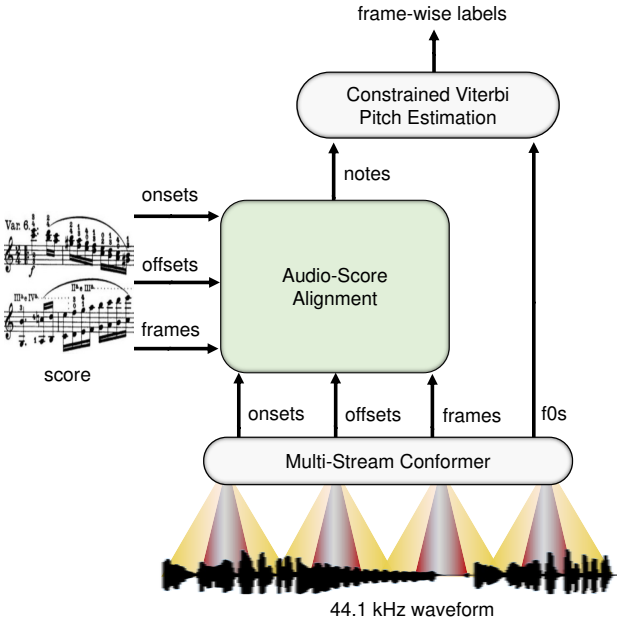


Figure 5: The proposed high-resolution violin transcription model only requires piece-level labels for learning as it can generate frame-wise labels using its own onset, offset, and frame feature representations.

200 cents around the note frequency. For each consecutive time instant, **S** allows smooth transitions with a Gaussian standard deviation of 25 cents, i.e., 2.5 f0 states:

$$s_{ij} = \frac{\exp\left(-\frac{1}{2}\left(\frac{j-i}{(25/10)}\right)^2\right)}{(25/10)\sqrt{2\pi}},$$

for $i, j \in [1 : 21]$, where s_{ij} denotes the state transition probabilities in the 10-cent resolution f0 matrix.

Since Viterbi algorithm has a complexity of $O(n^2)$, applying the pitch tracking within the constrained region also improves the runtime speed compared to Viterbi without note constraints. Moreover, applying Viterbi within note constraints allow detecting multiple f0s.

After per-note Viterbi paths are calculated, the frame-wise pitch predictions are obtained through the regional weighted averaging method from Kim et al. [29] to deter the f0 estimates through further interpolations.

3. LEARNING FROM WEAK LABELS

Our proposed method enables learning from weak labels, which involve pairs of violin recordings and their publicly-available scores. The learning procedure consists of four phases. First, we create initial audio-score alignments using music synchronization techniques. Second, we use the aligned audio-score pair for the first round of training. Third, we recompute the alignment using the estimated features. Fourth and finally, we finetune our model using the finer features learned by the model.

To create the initial audio-score alignments, we use dynamic time warping (DTW), which is a well-known technique for music synchronization [34–36]. Conventional methods for music synchronization typically use DTW and chroma features as the input representation [32, 37], whereas the integration of additional activation functions, e.g., onsets, beats, downbeats, has proven to enhance the synchronization accuracy [27, 28]. Since we deal with violin transcription in this paper, we follow the alignment method in [28], which deals with a similar scenario, i.e., audio-to-audio synchronization of string quartets. Inspired by their combined synchronization approach, we first incorporate beat, downbeat, and onset activation functions alongside chroma features to generate the initial audio-score alignments. The inclusion of activation functions results in a grid-like structure in the DTW cost matrix, which guides the alignment through activation cues that point to note onsets or other musical events. At the same time, chroma features account for the harmonic and melodic information.

Following the setting in [28], we use a sample rate of 22.05 kHz and a feature rate of 50 Hz to create the alignments. As this feature rate (20 ms) is coarser than the model's frame resolution (5.8 ms), we apply linear interpolation to create labels. Note that we cannot evaluate the synchronization accuracy of the training data since we do not have any annotations for these. Using these target labels obtained from the initial alignment, which can possibly be inaccurate, we train our model for one epoch in the first training phase.

Following the first training phase, we obtain the four learned representations, onset, offset, semitone-level frames, and high-resolution f0 frames for each audio-score pair. To acquire finer and more accurate labels, we run a novel synchronization stage. We recompute the alignment with the refined features, estimated semitone-level frame representations, and the activation with the stacked onset and offset features (see Section 2.3). Note that the feature rate we use in the alignment is the same as the MUSC features (hop size of 5.8 ms). Using the labels obtained from synchronization, we finetune our model using early stopping.

Our iterative training strategy resembles the approach by Maman and Bermano [2]. Their approach starts with training the transcription model with synthetic data and then creating the initial alignments with the features estimated by this model and involves three training iterations: first on synthetic data and two more iterations to finetune the model on the target dataset. In contrast, we start from a robust ASA and complete the training process in two iterations.

4. DATASET AND TRAINING

In this section, we describe our dataset that we use for the training and our training procedure. The weakly-labeled dataset consists of 120 scores and 34 hours of solo violin performances. We also provide automatic score alignments and frame-level pitch bends that are generated by our joint data curation and training process.

	#s	#p	#r	dur
Paganini, Op. 1	24	10	235	13:00
Wohlfahrt, Op. 45	60	6	506	11:36
Kayser, Op. 20	36	8	280	09:48
Total	**120**	**22**	**1021**	**34:24**

Table 1: Dataset statistics. #s: number of scores, #p: number of distinct players, #r: number of recordings, dur: total recording duration in hh:mm.

4.1 Dataset Statistics

Our dataset comprises public scores of 96 etudes which are included in the Violin Etudes dataset [38], i.e., Wohlfahrt Op. 45, and Kayser Op. 20. We also extend these scores with additional 24 etudes/caprices by Paganini Op. 1. In contrast to the Violin Etudes dataset, which only includes monophonic recordings, the recordings in our dataset include a mix of monophonic and polyphonic etudes. We collect multiple versions of these etudes from YouTube and automatically match and align them using the method described in Section 3. For the Paganini Op. 1 score, we noticed that performers do not always follow the repeat signs. To ensure better alignments, we automatically expand each repetition pattern individually and select the one that best matches the recording based on the alignment distance. As the most extreme case, we found four different repetition patterns for the Paganini Op. 1 No. 23, which we label as Op01-23, Op01-23-a, Op01-23-b, and Op01-23-c in the dataset, respectively.

The dataset we provide includes original YouTube links, annotated start and end timestamps, and aligned MIDI files containing multi-pitch bends. These resources can be utilized to generate expressive performances featuring vibrato. Moreover, for each etude and caprice, we provide at least five performances, which can be utilized for audio-to-audio synchronization and comparative studies. Table 1 summarizes the dataset statistics.

4.2 Training Details

Using Adam optimizer and a learning rate of $1\mathrm{e}{-3}$, we train the model to minimize the binary cross entropy (BCE) loss for the onset, offset, frame, and f0s:

$$\mathcal{L} = \mathcal{L}_{\text{onset}} + \mathcal{L}_{\text{offset}} + \mathcal{L}_{\text{frame}} + \frac{\mathcal{L}_{\text{f0}}}{10}.$$

In addition to Gaussian label smoothing as described in Section 2.3, we weight positive onset and offsets with 9 to balance the sparse matrices. Furthermore, we also observe that weighting the $\mathcal{L}_{\text{f0}}$ by $1/10$ helps in increasing the stability of the training.

Since our dataset includes several versions per piece, we do not employ further data augmentations. We train the model using a batch size of 16 and a fixed sequence duration of three seconds (512 frames). We employ $(80-20)$ train–validation splits and consider each sample with the etude no $\equiv 3 \pmod 5$ for the validation set.

After training for one epoch on the dataset obtained with initial alignments and pseudo f0 labels, we realign the dataset with the model's onset, offset, and frame features and apply constrained Viterbi tracking for the f0 labels. Using the new labels estimated by the model, we train the model further, applying early stopping.

5. EXPERIMENTS

While we aim at the task of descriptive violin transcription with high-resolution pitch bends, there is no previous work on which we can directly compare with. Therefore, we compare our model with general-purpose baselines for the closely-related proxy tasks of transcription and pitch estimation. We provide our experimental results on the violin tracks of two manually-annotated and corrected datasets, i.e., URMP [39] and Bach10 [40].

5.1 Test Datasets

The URMP dataset [39] is a multimodal dataset that includes 44 performances in various chamber ensemble settings. The dataset was annotated with the help of the Tony melody transcription software [41], which utilizes the pYIN [33] algorithm for the initial f0 estimates and applies a hidden Markov model for note quantization. The note onsets, offsets, and f0s are then manually corrected. For our evaluation, we use all the violin tracks from the URMP dataset. We note that one of our transcription baselines, the MT3 [3] model, was trained using this dataset. Since we employ our tests in the entirety of the violin tracks, the tests include the training samples of the MT3.

Our second test dataset, Bach10 [40], comprises 10 four-part chorales played by a violin, clarinet, tenor saxophone, and bassoon quartet. The ground-truth f0 annotations in the dataset were estimated first using the YIN [42] algorithm and then corrected manually. The dataset also includes note annotations derived from the beat times that are manually-annotated by musicians. However, the manual correction for offset times is not included in the dataset. For our evaluation, we use all the violin tracks from the Bach10 dataset. We note that the Bach10 dataset was included in the training of one of our baselines in pitch estimation, i.e., CREPE [29].

5.2 Evaluation Metrics

As a proxy to descriptive violin transcription, we evaluate our method's transcription and pitch estimation performance separately using the common `mir_eval` metrics, and compare with general-purpose baselines. For the transcription, we provide our results with Precision P, Recall R, F1-score F1, and F1-score without offset $\mathrm{F1}_{\text{no}}$ using the default thresholds. Namely, for P, R, and F1, a note is considered correct its pitch is within 50 cents, the onset is within 50 ms and the offset is within 20% of the note's duration. We also include an additional measure, $\mathrm{F1}_{\text{no}}$, where a note is considered correct if the onset is within 50 ms without considering the offset. For the pitch estimation experiments, we used the Raw Pitch Accuracy (RPA) metric with two thresholds: the standard RPA50 metric, which considers the estimate accurate if it is within 50

	URMP				Bach10			
	P	R	F1	$F1_{no}$	P	R	F1	$F1_{no}$
MUSC	**86.5**	83.1	**84.6**	**93.0**	**65.0**	**64.8**	**64.8**	**77.0**
MT3	79.1	**87.1**	82.2	88.9	54.2	51.5	52.7	62.0
BP	58.8	67.9	62.8	83.3	33.6	43.2	37.6	57.5

Table 2: Violin transcription results (%) comparing MUSC with two general-purpose AMT methods. Tests are conducted on all violin stems from the datasets. Bach10 represents the fair evaluation in a dataset unseen to all models. URMP was involved in the training dataset of the MT3, whereas it is unseen to both BP and MUSC.

	URMP				Bach10			
	P	R	F1	$F1_{no}$	P	R	F1	$F1_{no}$
Iter1	84.6	82.5	83.6	92.9	63.1	63.5	63.2	75.3
Iter2	**86.5**	**83.1**	**84.6**	**93.0**	**65.0**	**64.8**	**64.8**	**77.0**

Table 3: Violin transcription results (%) before (Iter1) and after (Iter2) fine-tuning the proposed MUSC model with the iterative alignment.

cents, and the RPA10 metric, which has a more strict 10-cent threshold.

5.3 Results

We compare MUSC with two recent general-purpose AMT baselines: Our first baseline is the Basic Pitch [1] (*BP*), which is a lightweight model for instrument-agnostic AMT. The postprocessing method of BP is optimized for $\mathrm{F1_{no}}$, and MUSC also shares the same postprocessing script with their default parameters. The second baseline we consider for transcription is the MT3 [3], which is a multi-instrument transcription model that predicts instrument labels alongside transcription. Since we only test on violin recordings, we combine their output without the instrument labels for fair evaluation.

Table 2 summarizes the results for the transcription experiments. At a first glance, the proposed violin-specific model MUSC outperforms MT3 and BP on both datasets, indicating that it is a more effective method for violin transcription. Even though the training set of MT3 included the test samples in the URMP dataset, MUSC yields the best F1-score value among the three AMT systems. Furthermore, the performance gap between MUSC and MT3 is greater for the Bach10, which was not included the training set of any method. The results indicate that the all the models yield rather poor scores on the Bach10 dataset when evaluated using the conventional P, R, and $\mathrm{F1}$ metrics. Since the offsets in the Bach10 dataset are not manually-corrected, the $\mathrm{F1_{no}}$ scores can be viewed as a better indicator of the transcription performance for this dataset.

We also compare our model's transcription performance before and after fine-tuning with alignments generated using its own feature representations. The Table 3 shows that some of the improvements in our model's transcription performance can be attributed to the iterative training strategy.

For the pitch estimation experiments, we compare MUSC with four well-known pitch estimators: the pre-trained CREPE model [29] from its official repository [3], pYIN [33], and YIN [42] from librosa [4], and SWIPE [43] from the libf0 library [5]. We use the same $F\sharp 3$ (min) to $E8$ (max) frequency range for a fair evaluation.

	URMP		Bach10	
	RPA50	RPA10	RPA50	RPA10
MUSC	98.3	89.0	98.3	86.9
vMUSC	**98.6**	**89.4**	98.4	87.0
CREPE	96.4	87.2	**98.6**	**88.1**
vCREPE	97.3	88.4	**98.6**	**88.1**
YIN	95.3	88.4	97.1	81.7
pYIN	97.2	88.6	97.4	80.3
SWIPE	97.2	89.3	97.7	84.3

Table 4: Violin Raw Pitch Accuracy (RPA, %) results. Note that the training set of CREPE involved the Bach10 dataset. vMUSC and vCREPE contain an additional Viterbi decoding stage.

Table 4 summarizes the pitch estimation results. First, all the pitch estimators achieve high accuracies on both datasets. For the URMP dataset which is unseen to all the models, vMUSC (MUSC with Viterbi decoding) outperforms the common state-of-the-art pitch estimators in terms of RPA50 and RPA10. For the Bach10 dataset, which is included in the training samples of the pre-trained CREPE model, the CREPE expectedly yields the best RPA values. Note that even though our model was not trained with these test samples from Bach10, MUSC remains to be competitive (e.g., 98.4% versus 98.6% RPA50 in Bach10).

6. CONCLUSION

In this paper, we introduced MUSC, an AMT system tailored for violin transcription through high-precision pitch bend estimation, and the capability of learning from piecewise weak labels. We showed that, by only utilizing 120 scores, we were able to obtain state-of-the-art transcription and pitch estimation results for the violin. We also shared our descriptive violin transcription dataset to the MIR community. In the future, we will focus on improving the note postprocessing and alignment stages of the MUSC in order to specialize better for the string repertoire, and use it as a large-scale dataset curation tool for strings music, ethnomusicology, and music education research. We believe that the descriptive music transcription capabilities of the MUSC will accelerate the research in music education, ethnomusicology, and expressive performance generation.

7. ACKNOWLEDGEMENTS

This research is funded by the project Musical AI - PID2019-111403GB-I00/AEI/10.13039/501100011033 funded by the Spanish Ministerio de Ciencia, Innovación y Universidades (MCIU) and the Agencia Estatal de Investigación (AEI), and by the German Research Foundation (DFG MU 2686/10-2).

[3] https://github.com/marl/crepe
[4] https://librosa.org/
[5] https://github.com/groupmm/libf0

8. REFERENCES

[1] 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.

[2] B. Maman and A. H. Bermano, "Unaligned supervision for automatic music transcription in the wild," in *Proceedings of the International Conference on Machine Learning (ICML)*, July 2022, pp. 14 918–14 934.

[3] 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. [Online]. Available: https://arxiv.org/abs/2111.03017

[4] 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*, H. T. Shen, Y. Zhuang, J. R. Smith, Y. Yang, P. Cesar, F. Metze, and B. Prabhakaran, Eds., Virtual Event, China, 2021, pp. 3918–3926.

[5] I. Simon, J. Gardner, C. Hawthorne, E. Manilow, and J. Engel, "Scaling polyphonic transcription with mixtures of monophonic transcriptions," in *Proc. International Society for Music Information Retrieval Conference (ISMIR)*, Bengaluru, India, 2022, pp. 749–756.

[6] S. Ewert and M. B. Sandler, "Piano transcription in the studio using an extensible alternating directions framework," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 11, pp. 1983–1997, 2016.

[7] R. Kelz and G. Widmer, "Towards interpretable polyphonic transcription with invertible neural networks," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Delft, The Netherlands, November 2019, pp. 376–383.

[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," pp. 246–253, 2021.

[10] R. Schramm and E. Benetos, "Automatic transcription of a cappella recordings from multiple singers," in *Proceedings of the AES International Conference on Semantic Audio*, Erlangen, Germany, 2017, pp. 108–115.

[11] R. Nishikimi, E. Nakamura, S. Fukayama, M. Goto, and K. Yoshii, "Automatic singing transcription based on encoder-decoder recurrent neural networks with a weakly-supervised attention mechanism," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Brighton, UK, 2019, pp. 161–165.

[12] T.-W. Su, Y.-P. Chen, L. Su, and Y.-H. Yang, "TENT: Technique-embedded note tracking for real-world guitar solo recordings," *Transactions of the International Society for Music Information Retrieval (TISMIR)*, vol. 2, no. 1, July 2019.

[13] A. Wiggins and Y. E. Kim, "Guitar tablature estimation with a convolutional neural network." in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Delft, Netherlands, November 2019, pp. 284–291.

[14] J. Abeßer and M. Müller, "Jazz bass transcription using a U-net architecture," *Electronics*, vol. 10, no. 6, p. 670, 2021.

[15] M. A. Kaliakatsos-Papakostas, A. Floros, M. N. Vrahatis, and N. Kanellopoulos, "Real-time drums transcription with characteristic bandpass filtering," in *Proceedings of the Audio Mostly: A Conference on Interaction with Sound*, Corfu, Greece, September 2012, pp. 152–159.

[16] C. Southall, R. Stables, and J. Hockman, "Automatic drum transcription using bi-directional recurrent neural networks," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, New York City, New York, USA, August 2016, pp. 591–597.

[17] K. Choi and K. Cho, "Deep unsupervised drum transcription," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Delft, The Netherlands, 2019, pp. 183–191.

[18] C. Seeger, "Prescriptive and descriptive music-writing," *The Musical Quarterly*, vol. 44, no. 2, pp. 184–195, 1958.

[19] P. C. Greene, *Violin performance with reference to tempered, natural, and Pythagorean intonation*. University of Iowa Press, 1937.

[20] J. M. Geringer, "Eight artist-level violinists performing unaccompanied bach: Are there consistent tuning patterns?" *String Research Journal*, vol. 8, no. 1, pp. 51–61, 2018.

[21] G. N. Swift, *The violin as cross cultural vehicle: Ornamentation in South Indian violin and its influence on a style of Western violin improvisation*. Wesleyan University, 1989.

[22] C. Weiß and G. Peeters, "Learning multi-pitch estimation from weakly aligned score-audio pairs using a multi-label CTC loss," in *Proceedings of the IEEE*

Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), New Paltz, USA, 2021, pp. 121–125.

[23] 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. [Online]. Available: https://openreview.net/forum?id=r1lYRjC9F7

[24] 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.

[25] M. Dorfer, A. Arzt, and G. Widmer, "Towards score following in sheet music images," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, New York City, New York, USA, 2016, pp. 789–795.

[26] T. Kwon, D. Jeong, and J. Nam, "Audio-to-score alignment of piano music using RNN-based automatic music transcription," in *Proceedings of the Sound and Music Computing Conference (SMC)*, Espoo, Finland, 2017, pp. 380–385.

[27] 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.

[28] Y. Özer, M. Istvanek, V. Arifi-Müller, and M. Müller, "Using activation functions for improving measure-level audio synchronization," in *Proc. International Society for Music Information Retrieval Conference (ISMIR)*, Bengaluru, India, 2022, pp. 749–756.

[29] J. W. Kim, J. Salamon, P. Li, and J. P. Bello, "CREPE: A convolutional representation for pitch estimation," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Calgary, Canada, 2018, pp. 161–165.

[30] 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 *Proceedings of the Annual Conference of the International Speech Communication Association (Interspeech)*, 2020, pp. 5036–5040.

[31] N. C. Tamer, Y. Özer, M. Müller, and X. Serra, "TAPE: An end-to-end timbre-aware pitch estimator," in *Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2023.

[32] M. Müller, *Fundamentals of Music Processing – Using Python and Jupyter Notebooks*, 2nd ed. Springer Verlag, 2021.

[33] M. Mauch and S. Dixon, "pYIN: A fundamental frequency estimator using probabilistic threshold distributions," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Florence, Italy, 2014, pp. 659–663.

[34] S. Salvador and P. Chan, "FastDTW: Toward accurate dynamic time warping in linear time and space," in *Proceedings of the KDD Workshop on Mining Temporal and Sequential Data*, 2004.

[35] S. Dixon and G. Widmer, "MATCH: A music alignment tool chest," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, London, UK, 2005, pp. 492–497.

[36] 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.

[37] R. B. Dannenberg and N. Hu, "Polyphonic audio matching for score following and intelligent audio editors," in *Proceedings of the International Computer Music Conference (ICMC)*, San Francisco, USA, 2003, pp. 27–34.

[38] N. C. Tamer, P. Ramoneda, and X. Serra, "Violin etudes: a comprehensive dataset for f0 estimation and performance analysis," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Bengaluru, India, 2022, pp. 517–524.

[39] 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, 2019.

[40] Z. Duan, B. Pardo, and C. Zhang, "Multiple fundamental frequency estimation by modeling spectral peaks and non-peak regions," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 8, pp. 2121–2133, 2010.

[41] M. Mauch, C. Cannam, R. Bittner, G. Fazekas, J. Salamon, J. Dai, J. Bello, and S. Dixon, "Computer-aided melody note transcription using the Tony software: Accuracy and efficiency," in *Proceedings of the International Conference on Technologies for Music Notation and Representation*, 2015.

[42] A. de Cheveigné and H. Kawahara, "YIN, a fundamental frequency estimator for speech and music." *Journal of the Acoustical Society of America (JASA)*, vol. 111, no. 4, pp. 1917–1930, 2002.

[43] A. Camacho and J. G. Harris, "A sawtooth waveform inspired pitch estimator for speech and music," *The Journal of the Acoustical Society of America*, vol. 124, no. 3, pp. 1638–1652, 2008.

POLYFFUSION: A DIFFUSION MODEL FOR POLYPHONIC SCORE GENERATION WITH INTERNAL AND EXTERNAL CONTROLS

Lejun Min[1,2,4] **Junyan Jiang**[1,2] **Gus Xia**[1,2] **Jingwei Zhao**[3]
[1] Music X Lab, Computer Science Department, NYU Shanghai
[2] MBZUAI [3] Institute of Data Science, NUS
[4] Zhiyuan College, Shanghai Jiao Tong University
aik2mlj@gmail.com, {jj2731, gxia}@nyu.edu, jzhao@u.nus.edu

ABSTRACT

We propose Polyffusion, a diffusion model that generates polyphonic music scores by regarding music as image-like piano roll representations. The model is capable of controllable music generation with two paradigms: *internal* control and *external* control. Internal control refers to the process in which users pre-define a part of the music and then let the model infill the rest, similar to the task of masked music generation (or music inpainting). External control conditions the model with external yet related information, such as chord, texture, or other features, via the cross-attention mechanism. We show that by using internal and external controls, Polyffusion unifies a wide range of music creation tasks, including melody generation given accompaniment, accompaniment generation given melody, arbitrary music segment inpainting, and music arrangement given chords or textures. Experimental results show that our model significantly outperforms existing Transformer and sampling-based baselines, and using pre-trained disentangled representations as external conditions yields more effective controls. [1]

1. INTRODUCTION

Diffusion models [1, 2], as a new class of generative models, have been successful in generating high-quality samples of image data and beyond. They achieve state-of-the-art sample quality on a number of image generation benchmarks [3, 4], and also show strong results for the generation of various media such as audio [5, 6], video [7–9], and text [10, 11].

Symbolic music generation, a task very different from audio generation, has highly discrete outputs and is often described in terms of constraint optimization problems [12, 13]. Despite the improvement of deep music generative modeling [14, 15], symbolic music generation still suffers from the lack of controllability and consistency at different time scales [16]. In our study, we experiment with the idea of using diffusion models to approach controllable symbolic music generation.

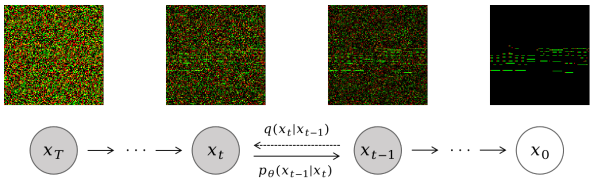


Figure 1: The forward and reverse process of the proposed diffusion model trained on piano roll representations. The red dot at the front of each note denotes its onset; the green bar following it denotes its sustain. Notice that the image axes are swapped for proper visualization.

Inspired by the high-quality and controllable image generation that diffusion models have achieved in computer vision, we devise an image-like piano roll format as the input, and used a UNet-based diffusion model to stepwise denoise a randomly sampled piano roll, as illustrated in Figure 1. We show in our experiments and demos that our design provides excellent generation results.

Besides unconditional generation, the model also accepts two categories of controls, namely internal control and external control:

- **Internal Control (Inpainting):** By masking out part of the given piano roll, we can specify the remaining area to be generated, thus implicitly conditioning the generation to fit in the masked part. We regard this strategy as a generalized music inpainting method.

- **External Control (Conditional Generation):** By adopting the cross-attention mechanism of Latent Diffusion [17], we can explicitly control the music generation on given external conditions such as chords and textures. They are first encoded into latent representations using pre-trained, disentangled variational autoencoders (VAEs), and then fed into the backbone UNet of the diffusion model to condition the denoising process. We show that the generated music complies with the given conditions well. We also add classifier-free guidance to control the variance of the generation.

[1] Demo page: https://polyffusion.github.io/. Code repository: https://github.com/aik2mlj/polyffusion

© L. Min, J. Jiang, G. Xia, and J. Zhao. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).
Attribution: L. Min, J. Jiang, G. Xia, and J. Zhao, "Polyffusion: A Diffusion Model for Polyphonic Score Generation with Internal and External Controls", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.*, Milan, Italy, 2023.

These controls of diffusion models enable us to unify a wide spectrum of creative music tasks that previously require separate modeling and training. In this paper, we showcase the following scenarios:

- **Melody generation given accompaniment** by generation with the accompaniment part being masked out.
- **Accompaniment generation given melody** by generation with the melody part being masked out.
- **Arbitrary music segment inpainting** by generation with any time segments being masked out.
- **General music arrangement given chords or textures** by conditioning on external chord or texture signals.

2. RELATED WORK

We review three realms of related work: 1) music inpainting, which is related to our internal control method, 2) conditioned music generation with external signals, which is related to our external control method, and 3) recent progress on diffusion-based modeling in the music domain.

2.1 Music Inpainting

Music inpainting is a controlled music generation task that regulates the generation with pre-defined musical contexts. We see various studies on polyphonic music inpainting. For example, DeepBach [18] develops a context-aware recurrent neural network (RNN) capable of inpainting missing notes for chorales in the style of Bach. Coconet [19] uses blocked Gibbs sampling to repeatedly rewrite a masked music score. Chang et al. [20] achieve variable-length music score inpainting. Music SketchNet [21] and MusIAC [22] introduce various controls to the inpainting task under VAE-based and Transformer-based framework respectively. Comparatively, diffusion models naturally possess the inpainting ability via masked generation [23], and there is no need to train or fine-tune a task-specific model for inpainting.

Though the current inpainting tasks mostly apply masks over a continuous period of time, the inpainted area, in theory, can be any note in the score (any area of a piano roll). In this study, we show that our image-like representation enables both part-wise and time-wise inpainting. The former refers to inpainting melody or accompaniment part given the other part, while the latter refers to infilling notes falling in arbitrary time segments.

2.2 Music Generation Conditioned on External Signals

External control signals are also one of the mainstream methods to control the music generation process. Common scenarios include generating music given chords [18, 24–26], lyrics [27], and other relevant features such as note density and voicing numbers [28].

Our study focuses on polyphonic score generation controlled by external chords and textures. In particular, the "control by texture" task has great practical value in both music arrangement and composition style transfer [29], while very few existing models could realize this function.

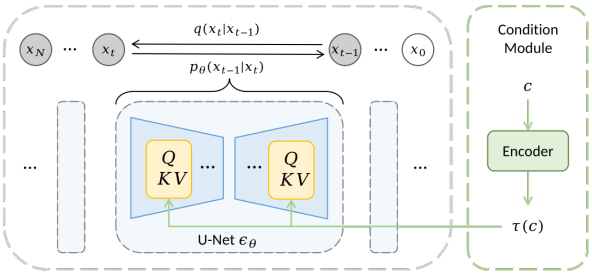


Figure 2: The model structure with an additional condition module for external control. Each UNet unit ϵ_θ applies one denoising step during the reverse process. External condition signals are encoded by pre-trained encoders and fed into the cross-attention layers, which are represented by the yellow squares in the UNet unit.

2.3 Diffusion Models for Music Generation

Recently, we have seen several attempts to introduce diffusion models to symbolic music tasks. Mittal et al. [30] generate monophonic music by training a diffusion model on the latent representations learned by MusicVAE [31]. Cheuk et al. [32] brings diffusion models to the music transcription task by adapting the piano roll format into the DiffWave [5] structure. It is relevant to our study as the model can also output piano rolls. However, the model focuses on transcription instead of generation by relying on a ground-truth spectrogram as its control. In general, for symbolic music generation, conditioning diffusion models on external controls is still an area to be explored.

3. METHODS

3.1 Data Representation

Our image-like piano roll representation is a 2-channel *binary* tensor $x \in \mathbb{R}^{2\times T\times P}$. The generation task targets 8-bar (32-beat) long music segments, with 1/4 beat as the time step, resulting in $T = 128$ time steps per sample. We use a MIDI pitch range 0...127, resulting in $P = 128$ pitch bins. Each entry $x(c, t, p)$ represents whether there is a note onset (for $c = 0$) or sustain (for $c = 1$) at time step t and MIDI pitch p.

3.2 Diffusion Model

Diffusion models [1, 2] are latent-variable models comprised of a forward (diffusion) process which gradually disrupts the structure of data x_0 and a reverse (denoising) process that learns to recover the original data x_0 from the noisy input. In our study, x_0 denotes the clean piano roll. The forward process iteratively adds Gaussian noise in N diffusion steps:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I) \quad (1)$$

$$q(x_{1:N}|x_0) = \prod_{t=1}^{N} q(x_t|x_{t-1}) \quad (2)$$

where $\beta_1, \beta_2, \ldots, \beta_N$ are a series of variance scheduling parameters. The reverse process requires the model to parameterize a Markov chain that iteratively reconstructs the piano roll x_0 from a corrupted input $x_N \sim \mathcal{N}(0, I)$.

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \sigma_\theta(x_t, t)) \quad (3)$$

$$p_\theta(x_{0:N}) = p(x_N) \prod_{t=1}^{N} p_\theta(x_{t-1}|x_t) \quad (4)$$

During training, we optimize the model parameters ϵ_θ by minimizing the following target:

$$L(\theta) = \mathbb{E}_{x_0,\epsilon,t}\left[\left\|\epsilon - \epsilon_\theta\left(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t\right)\right\|^2\right] \quad (5)$$

where t is uniformly sampled from $[1, N]$ and $\epsilon \sim \mathcal{N}(0, I)$, $\alpha_t := 1 - \beta_t$, $\bar{\alpha}_t := \prod_{s=1}^{t} \alpha_s$. As shown in Figure 2, our unconditional model structure is based on [2], an image-oriented diffusion model using a 2-D UNet as its backbone ϵ_θ.

3.3 Internal Control (Inpainting)

Internal control refers to the use of the music notes themselves to regulate and influence the generation process, and we regard music inpainting as a means of internal control.

Specifically, we denote the given piano roll sample as s and the mask as m. At each step t during inference sampling, the fixed area of the image is diffused with the forward process $q(s_t|s) = \mathcal{N}(s_t; \sqrt{\bar{\alpha}_t}s, (1-\bar{\alpha}_t)I)$ and put together with the denoising sample s_{t-1}. Algorithm 1 [23] shows the detailed implementation of this inpainting process.

Algorithm 1 Inpainting Process

Input: inpainting mask m, original sample s, $x_N \sim \mathcal{N}(0, I)$

1: **for** $t = N, \ldots, 1$ **do**
2: $\epsilon_1, \epsilon_2 \sim \mathcal{N}(0, I)$ if $t > 1$, else $\epsilon_1 = \epsilon_2 = 0$
3: $y = \sqrt{\bar{\alpha}_t}s + \sqrt{1-\bar{\alpha}_t}\epsilon_1$ if $t > 1$, else s
4: $x_{t-1} = \mu_\theta(x_t, t) + \sigma_\theta(x_t, t)\epsilon_2$
5: $x_{t-1} = x_{t-1} \odot (1-m) + y \odot m$
6: **end for**
7: **return** x_0

3.4 External Control (Conditional Generation)

External control means using external signals to condition the generation process. We aim to incorporate a general strategy that does not place strong assumptions on the *format* of input control signals. To this end, we use the cross-attention mechanism [33] for conditional generation introduced by Latent Diffusion [17] since it is insensitive to the dimension of the condition signals. We also adopted the strategy used by Rombach et al. [17], which augments the backbone UNet structure with cross-attention layers that map condition signals into the UNet intermediate latent representations.

Formally, to preprocess the external musical signal c, we introduce a corresponding encoder τ that projects c to a latent representation $\tau(c)$. The encoder τ is pre-trained and fixed during diffusion model training. The cross-attention layers then map $\tau(c)$ to the intermediate layers of the UNet (as shown in Figure 2). The conditional training objective is

$$L_{\text{cond}}(\theta) := \mathbb{E}_{x_0,c,\epsilon,t}\left[\left\|\epsilon - \epsilon_\theta\left(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t, \tau(c)\right)\right\|^2\right] \quad (6)$$

We use classifier-free guidance (CFG) [34] to enable both conditioned and unconditioned generation by controlling the intensity of the condition signals during sampling. We refer readers to [34] and [35] for details on CFG.

4. CONTROLLABLE MUSIC GENERATION

In this section, we present four general musical applications our model empowers with internal and external controls: 1) melody generation given accompaniment, 2) accompaniment generation given melody, 3) arbitrary music segment inpainting, and 4) music arrangement given chords or textures. For each application, we provide non-cherry-picked generated samples as a case study. We also refer readers to our demo page for more examples.

4.1 Melody Generation Given Accompaniment

This task is achieved by internal control — to pre-define the accompaniment part and let the model infill the upper melody. Figure 3(a) shows an example of pop song melody generation given the accompaniment. We see that the melody is consistent with the underlying chords of the given accompaniment, and maintains an overall consistent rhythmic pattern, except for a 16th-note jump at the beginning of the 3rd bar.

4.2 Accompaniment Generation Given Melody

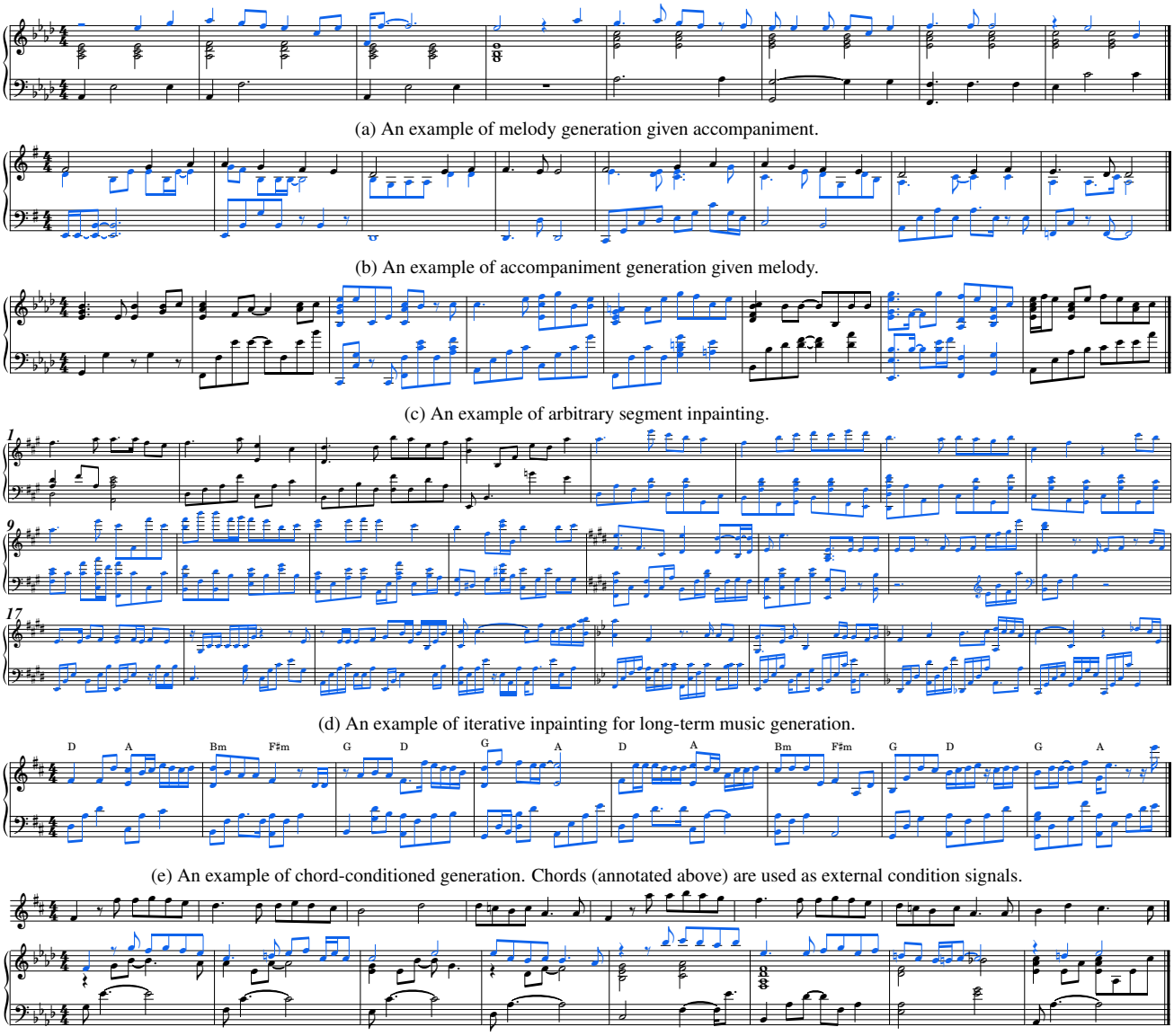
Similarly, given a lead melody, we can inpaint its corresponding lower accompaniment. Figure 3(b) shows an example, in which we see that the generated chord sequence suits the key (E minor) of the melody well, realized by a consistent arpeggio texture. The generated counter-melody also fills in the gaps between melody onsets well.

4.3 Arbitrary Music Segment Inpainting

The common scenario of music inpainting, also called music infilling [20], is to generate a music segment that fills in the gap between given past and future contexts. For our model, this task can be fulfilled by masking out the full pitch range of selected bars for inpainting.

Figure 3(c) shows an example of the inpainting process of the 3rd, 4th, 5th, and 7th bars, given the rest as fixed contexts. In the example, the model is capable of generating a full cadence connecting the 7th and the 8th bar, and also a nice applied chord in the non-diatonic progression Gm-Adim-B♭m connecting the 5th and the 6th bar.

We also extend the problem setup and let the diffusion model generate *long-term* music by iteratively inpainting



(a) An example of melody generation given accompaniment.

(b) An example of accompaniment generation given melody.

(c) An example of arbitrary segment inpainting.

(d) An example of iterative inpainting for long-term music generation.

(e) An example of chord-conditioned generation. Chords (annotated above) are used as external condition signals.

(f) An example of texture-conditioned melody generation. The texture of a given melody (the staff above) is used as external condition signals.

Figure 3: Generated samples in various tasks of controllable music generation. The generated parts are marked in blue. These examples have corresponding hearable demos on the demo page.

the future given the past. Figure 3(d) shows an example of a 24-bar generation based on a 4-bar prompt. The model generates 4 bars during each inference and finishes the process with five iterations. We see that the generated music contains a smooth chord progress, with a key modulation towards the end. The long-term textural structure is coherent, however lacking a consistent music theme.

4.4 Music Arrangement Given Chords or Textures

Inspired by the *chord-texture* disentanglement work [13, 29], we choose these two factors as the external condition signals for polyphonic generation. In our context, chords refer to the harmonic information, and textures refer to the rhythmic information. The latent chords and textures are encoded using pre-trained VAEs and cross-attended with the backbone UNet.

Beat-wise chords are first extracted by rule-based methods [36, 37], in which we adopted a 36-D chord representation consisting of a 12-D one-hot root encoding, a 12-D one-hot bass encoding and 12-D multi-hot chroma encoding. We then use a chord VAE [13] to extract a 512-D representation for each 8-bar chord sequence. For texture conditioning, we encode each 2-bar segment with the pre-trained texture encoder in [13] and then concatenate four encoded 256-D representations into a 1024-D vector as an 8-bar texture representation.

Figure 3(e) demonstrates an example of polyphonic music generation conditioned on chords. In the example, the accompaniment and the melody are mostly chord notes, with a certain degree of non-chord passing and neighboring tones that increase the interestingness of the song.

To show the complex combinations of conditions that the model can handle, we showcase a "texture-specified melody generation" for a given accompaniment segment as an example of the combination of internal and external

controls. As shown in Figure 3(f), We generate the melody part of a given accompaniment segment conditioned on the encoded texture representations of a given melody line. The result preserves a similar rhythmic pattern and fits the tonality of the new accompaniment.

5. EXPERIMENTS

5.1 Dataset and Training

We train our model using the POP909 dataset [38], a pop song dataset containing around 1K MIDI files. We only keep the pieces with 2/4 and 4/4 meters and cut them into 8-bar music segments with 1-bar hopping size, which results in 64K samples in total. The dataset is randomly split into the training set (90%) and validation set (10%) on a song level. The training samples are randomly transposed to all 12 keys for data augmentation.

The classifier-free guidance technique stated in Section 3.4 combines unconditional and conditional training. We adopt the implementation of DDPM and cross-attention layers in [39]. With 1K total diffusion steps, the model converges around 50 epochs (200K steps) on Adam Optimizer [40] with a constant learning rate 5e-5.

To turn the generated 2-channel piano roll representations into MIDI files, we round them to $\{0, 1\}$ and neglect notes without an onset. In practice, the generation process of 160 8-bar samples report zero invalid notes.

5.2 Evaluation

To validate the generation quality and control effectiveness of our model, we conducted both objective and subjective evaluations on 5 tasks: (1) unconditional generation, (2) accompaniment generation, (3) segment inpainting, (4) chord-conditioned generation, and (5) texture-conditioned generation. Tasks 2-3 focus on the evaluation of internal controls, and tasks 4-5 focus on external controls. Table 1 summarizes the evaluation method for each task.

5.2.1 Evaluation Metrics

Objective metrics: To objectively measure the music quality for all 5 tasks, we use the averaging overlapped area of pitch distribution ($\mathcal{D}_{\mathrm{P}}$) and duration distribution ($\mathcal{D}_{\mathrm{D}}$) from [41], which measure the distribution similarity of pitch and duration between the generated samples and ground truth. Additionally, we introduce *chord distance* (CD) [41] and *onset distance* (OD) to evaluate the efficacy of external control. These metrics measure the ℓ_2 distance of chord (for task 4) and onset distribution (for task 5) between the generated samples and the chord/texture condition.

Subjective metrics: Subjective metrics include *creativity* (C), *naturalness* (N), and *musicality* (M), which provide a perceptive evaluation complementing the objective musical quality metrics. To demonstrate the efficacy of internal control, we pick accompaniment generation as an example and add a *fitness* (F) metric to evaluate how well the generated parts fit in with the given melody.

5.2.2 Baseline models

We use two types of models as our baselines:

Transformer models: As suggested in the polyphonic representation disentanglement study [13], applying a Transformer on disentangled latent codes yields better results than raw token predictions. Following [13], we train a Transformer to predict the chord and texture representations from melody representations. For unconditional generation (task 1), we sample the latent spaces of the first 2-bar melody and then predict its accompaniment and the following content. For accompaniment generation (task 2) and external conditioning (tasks 4-5), the melody (task 2), chord (task 4), or texture (task 5) latent representation is directly encoded as the condition for the Transformer. We adopt the XLNet-based model proposed in [20] for the music segment inpainting task (task 3).

Sampling-based models: We adopt the VAE-based disentanglement model in [13] and generate music segments by sampling the latent spaces. For unconditional generation (task 1), we sample from the chord and texture latent spaces of the first and the last 2 bars, then linearly interpolate the middle latent codes to form a coherent 8-bar segment. For inpainting (task 3), we also use linear interpolation on latent codes to infill the missing bars. For external conditioning (tasks 4-5), the chord (task 4) or texture (task 5) latent component is directly encoded from the given condition.

5.3 Comparative Results

We calculate the average of each objective metric on 160 generated samples for each task. As shown in Table 2, Polyffusion and its variations achieve the highest objective scores in tasks 1-4. For controllability, our model yields competitive results on segment inpainting and chord-conditioned generation. For the texture-conditioned generation task, our model does not perform as well as the baseline but is capable of preserving the general musical texture, since the baseline model is explicitly trained on texture reconstruction targets, while the texture condition of our model only serves as a hint for the generation.

We also show the effectiveness of classifier-free guidance in Table 2. With a guidance scale of 5, the model (Polyffusion-S5) shows improved controllability on both chord conditioning and texture conditioning. Notably, a large guidance scale for chord conditions negatively impacts the $\mathcal{D}_{\mathrm{D}}$ metric. We speculate that this is because notes regular in length provide clearer chord context, which can be noticed in the guidance demos.

For subjective evaluation, we invite participants to rate the generation quality via a double-blind online survey. Our survey consists of 4 groups of samples of unconditional generation and accompaniment generation, respectively. Each group contains a ground-truth piece, generated samples by Polyffusion and all baselines with random orders. 36 participants completed our survey. Each participant rated 4 random groups on average based on a 5-point scale. The evaluation results are shown in Figure 4 and 5. The height of each bar represents the mean rating,

	(1) Uncond. Gen.	(2) Acc. Gen.	(3) Seg. Inp.	(4) Chord Cond.	(5) Texture Cond.
Objective Metrics	$\mathcal{D}_{\mathrm{P}}$, $\mathcal{D}_{\mathrm{D}}$	$\mathcal{D}_{\mathrm{P}}$, $\mathcal{D}_{\mathrm{D}}$	$\mathcal{D}_{\mathrm{P}}$, $\mathcal{D}_{\mathrm{D}}$	$\mathcal{D}_{\mathrm{P}}$, $\mathcal{D}_{\mathrm{D}}$, CD	$\mathcal{D}_{\mathrm{P}}$, $\mathcal{D}_{\mathrm{D}}$, OD
Subjective Metrics	C, N, M	C, N, M, F	N/A	N/A	N/A
Generative Length	8 bars	8 bars	4 bars	8 bars	8 bars
Transformer Baselines	Wang	Wang	Chang	Wang	Wang
Sampling Baselines	Wang	N/A	Wang	Wang	Wang

Table 1: Specifications of the evaluation tasks and the baseline models. C, N, M, F in subjective metrics mean creativity, naturalness, musicality, and fitness respectively. *Wang* refers to the Transformer models (for Transformer baselines) and VAE-based models (for sampling baselines) in [13]; *Chang* refers to the XLNet-based model in [20].

	Uncond. Gen.		Acc. Gen.		Seg. Inp.		Chord Cond.			Texture Cond.		
	$\mathcal{D}_{\mathrm{P}}\uparrow$	$\mathcal{D}_{\mathrm{D}}\uparrow$	$\mathcal{D}_{\mathrm{P}}\uparrow$	$\mathcal{D}_{\mathrm{D}}\uparrow$	$\mathcal{D}_{\mathrm{P}}\uparrow$	$\mathcal{D}_{\mathrm{D}}\uparrow$	$\mathcal{D}_{\mathrm{P}}\uparrow$	$\mathcal{D}_{\mathrm{D}}\uparrow$	CD $\downarrow$	$\mathcal{D}_{\mathrm{P}}\uparrow$	$\mathcal{D}_{\mathrm{D}}\uparrow$	OD $\downarrow$
Polyffusion	**0.89**	**0.93**	**0.89**	**0.96**	**0.90**	**0.93**	0.90	**0.96**	0.75	0.88	**0.98**	1.85
Polyffusion-S5	0.89	0.93	0.89	0.96	0.90	0.93	**0.92**	0.81	**0.51**	0.87	0.97	1.75
Polyffusion-A	0.89	0.93	0.89	0.96	0.90	0.93	0.90	0.94	0.79	**0.95**	**0.98**	4.37
Transformer	0.78	0.84	0.88	0.89	**0.90**	0.83	0.87	0.88	0.56	0.84	0.93	**0.13**
Sampling	0.86	0.90	N/A	N/A	0.89	0.91	0.86	0.90	0.70	0.91	0.93	0.20

Table 2: The objective evaluation and ablation study results. The statistics of generation, accompaniment generation and segment inpainting are identical for three Polyffusion models (hence gray-out for the latter two models) since they share the same internal control method.

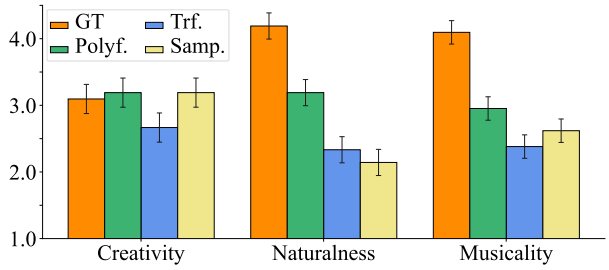


Figure 4: Subjective evaluation for unconditional generation.

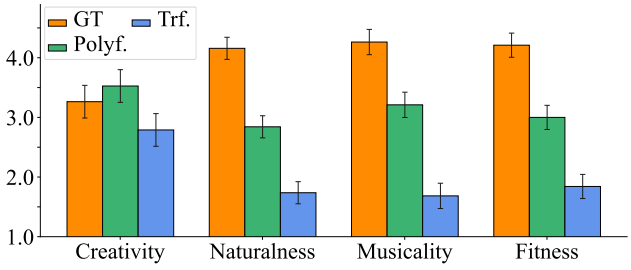


Figure 5: Subjective evaluation for accompaniment generation.

and the error bars are MSEs computed by within-subject ANOVA [42]. We report a significantly better performance (p-value < 0.05) of Polyffusion than baseline models in *naturalness* and *musicality* for both tasks and in *fitness* for accompaniment generation. Interestingly, Polyffusion even outperforms the ground truth on the *creativity* metric.

5.4 Ablation Study

We perform an ablation test on the use of VAE encoders for condition signals. For both chord conditioning and texture conditioning, we remove the corresponding pre-trained encoders. The ablated model of chord conditioning uses concatenated 36-D chord vectors as the condition signals. The ablated model of texture conditioning uses a modified piano roll representation [13]. Both models are trained with the same settings as the proposed model. Table 2 shows that the ablated models (Polyffusion-A) perform worse than the proposed models on the controllability metrics (CD & OD), showing the advantage of using disentangled latent representations as condition signals for diffusion models.

6. CONCLUSION AND FUTURE WORK

In this paper, we propose a diffusion model for polyphonic symbolic music generation. We show that an image-like piano roll representation is effective for modeling the musical context for a high-quality score generation. We specify two methods for controllable generation: internal control via masked generation, and external control via conditioning using cross-attention. Experiments show that our method achieves higher quality and controllability compared to the Transformer and sampling-based baselines on both internal and external control tasks.

We regard the diffusion framework as a prospective direction for future work on controllable music generation, since it achieves fine-grained controls over high-quality generation and enables a wide spectrum of arrangement applications. Currently, our generation is limited to quantized music scores without performance features. We plan to extend this methodology to expressive performance modeling. Several new controls can also be introduced to facilitate human-AI co-creation of symbolic music, e.g., hierarchical structure controls (e.g., music segment labels) and multimodal controls (e.g., text descriptions).

7. REFERENCES

[1] 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.

[2] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in Neural Information Processing Systems*, vol. 33, pp. 6840–6851, 2020.

[3] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," *Advances in Neural Information Processing Systems*, vol. 34, pp. 8780–8794, 2021.

[4] J. Ho, C. Saharia, W. Chan, D. J. Fleet, M. Norouzi, and T. Salimans, "Cascaded diffusion models for high fidelity image generation." *J. Mach. Learn. Res.*, vol. 23, pp. 47–1, 2022.

[5] Z. Kong, W. Ping, J. Huang, K. Zhao, and B. Catanzaro, "Diffwave: A versatile diffusion model for audio synthesis," *arXiv preprint arXiv:2009.09761*, 2020.

[6] N. Chen, Y. Zhang, H. Zen, R. J. Weiss, M. Norouzi, and W. Chan, "Wavegrad: Estimating gradients for waveform generation," *arXiv preprint arXiv:2009.00713*, 2020.

[7] R. Yang, P. Srivastava, and S. Mandt, "Diffusion probabilistic modeling for video generation," *arXiv preprint arXiv:2203.09481*, 2022.

[8] J. Ho, T. Salimans, A. Gritsenko, W. Chan, M. Norouzi, and D. J. Fleet, "Video diffusion models," *arXiv preprint arXiv:2204.03458*, 2022.

[9] W. Harvey, S. Naderiparizi, V. Masrani, C. Weilbach, and F. Wood, "Flexible diffusion modeling of long videos," *arXiv preprint arXiv:2205.11495*, 2022.

[10] X. L. Li, J. Thickstun, I. Gulrajani, P. Liang, and T. B. Hashimoto, "Diffusion-lm improves controllable text generation," *arXiv preprint arXiv:2205.14217*, 2022.

[11] S. Gong, M. Li, J. Feng, Z. Wu, and L. Kong, "Diffuseq: Sequence to sequence text generation with diffusion models," *arXiv preprint arXiv:2210.08933*, 2022.

[12] F. Pachet and P. Roy, "Musical harmonization with constraints: A survey," *Constraints*, vol. 6, no. 1, pp. 7–19, 2001.

[13] Z. Wang, D. Wang, Y. Zhang, and G. Xia, "Learning interpretable representation for controllable polyphonic music generation," *arXiv preprint arXiv:2008.07122*, 2020.

[14] 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.

[15] 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.

[16] J.-P. Briot and F. Pachet, "Deep learning for music generation: challenges and directions," *Neural Computing and Applications*, vol. 32, no. 4, pp. 981–993, 2020.

[17] 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.

[18] 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.

[19] C.-Z. A. Huang, T. Cooijmans, A. Roberts, A. Courville, and D. Eck, "Counterpoint by convolution," *arXiv preprint arXiv:1903.07227*, 2019.

[20] C.-J. Chang, C.-Y. Lee, and Y.-H. Yang, "Variable-length music score infilling via xlnet and musically specialized positional encoding," *arXiv preprint arXiv:2108.05064*, 2021.

[21] K. Chen, C.-i. Wang, T. Berg-Kirkpatrick, and S. Dubnov, "Music sketchnet: Controllable music generation via factorized representations of pitch and rhythm," *arXiv preprint arXiv:2008.01291*, 2020.

[22] 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.

[23] A. Lugmayr, M. Danelljan, A. Romero, F. Yu, R. Timofte, and L. Van Gool, "Repaint: Inpainting using denoising diffusion probabilistic models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 11 461–11 471.

[24] I. Simon, D. Morris, and S. Basu, "Mysong: automatic accompaniment generation for vocal melodies," in *Proceedings of the SIGCHI conference on human factors in computing systems*, 2008, pp. 725–734.

[25] Y.-S. Huang and Y.-H. Yang, "Pop music transformer: Generating music with rhythm and harmony," *arXiv preprint arXiv:2002.00212*, 2020.

[26] 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.

[27] Z. Ju, P. Lu, X. Tan, R. Wang, C. Zhang, S. Wu, K. Zhang, X. Li, T. Qin, and T.-Y. Liu, "Telemelody: Lyric-to-melody generation with a template-based two-stage method," *arXiv preprint arXiv:2109.09617*, 2021.

[28] J. Zhao and G. Xia, "Accomontage: Accompaniment arrangement via phrase selection and style transfer," *arXiv preprint arXiv:2108.11213*, 2021.

[29] S. Dai, Z. Zhang, and G. G. Xia, "Music style transfer: A position paper," *arXiv preprint arXiv:1803.06841*, 2018.

[30] G. Mittal, J. Engel, C. Hawthorne, and I. Simon, "Symbolic music generation with diffusion models," *arXiv preprint arXiv:2103.16091*, 2021.

[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*. PMLR, 2018, pp. 4364–4373.

[32] 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," *arXiv preprint arXiv:2210.05148*, 2022.

[33] 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.

[34] J. Ho and T. Salimans, "Classifier-free diffusion guidance," *arXiv preprint arXiv:2207.12598*, 2022.

[35] S. Dieleman, "Guidance: a cheat code for diffusion models," 2022. [Online]. Available: https://benanne.github.io/2022/05/26/guidance.html

[36] B. Pardo and W. P. Birmingham, "Algorithms for chordal analysis," *Computer Music Journal*, vol. 26, no. 2, pp. 27–49, 2002.

[37] 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 15th International Conference on Music Information Retrieval*, 2014.

[38] 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.

[39] N. W. Varuna Jayasiri, "labml.ai annotated paper implementations," 2020. [Online]. Available: https://nn.labml.ai/

[40] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[41] 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.

[42] H. Scheffe, *The analysis of variance*. John Wiley & Sons, 1999, vol. 72.

THE COORDINATED CORPUS OF POPULAR MUSICS (COCOPOPS): A META-CORPUS OF MELODIC AND HARMONIC TRANSCRIPTIONS

Claire Arthur
Georgia Institute of Technology
School of Music
claire.arthur@gatech.edu

Nathaniel Condit-Schultz
Georgia Institute of Technology
School of Music
natcs@gatech.edu

ABSTRACT

This paper introduces a new corpus, CoCoPops: The Coordinated Corpus of Popular Musics. The corpus can be considered a "meta corpus" in that it both extends and combines two existing corpora—the widely-used McGill Billboard corpus the and RS200 corpus. Both the McGill Billboard corpus and the RS200 contain expert harmonic annotations using different encoding schemes and each represent harmony in fundamentally different ways: Billboard using a root-quality representation and the RS200 using Roman numerals. By combining these corpora into a unified format, using the well-known `**kern` and `**harm` representations, we aim to facilitate research in computational musicology, which is frequently burdened by corpora spread across multiple encoding formats. The format will also facilitate cross-corpus comparison with the large body of existing works in `**kern` format. For a 100-song subset of the CoCoPops-Billboard collection, we also provide participant ratings of continuous valence and arousal ratings, along with the RMS (Root Mean Square) signal level and associated timestamps. In this paper we describe the corpus and the procedures used to create it.

1. INTRODUCTION

In 2011, Burgoyne et al. [1] introduced a dataset that would have a lasting influence in the ISMIR community: the McGill Billboard corpus, a set of expert harmonic analyses of commercial pop songs. This dataset—and the Harte [2] standard for encoding chord symbols that it adopted—has become a standard in the MIR community, for example, being used as training and testing data in the MIREX competition for Audio Chord Estimation since 2008. Around the same time, Trevor de Clercq and David Temperley independently created another rock music dataset—the RS200 corpus—which would ultimately consist of 200 harmonic *and* melodic transcriptions [3, 4]; Though perhaps less well known in the MIR community, their corpus has been the basis for several computational musicology papers [4]. While other datasets of popular-style music harmony have been released (e.g., Isophonics [2]), the Billboard and RS200 datasets stand out for their use of experts to encode the annotations, the rigor of their sampling methodologies, and the detail of their procedural documentation.

The field of computational musicology suffers from perennial data scarcity [5]; What few symbolic corpora exist are largely biased towards Western classical music [6], which is relatively easy to digitize due to its basis in notated scores. Unlike classical music, popular music must generally be transcribed from audio recordings, with melody transcription being a particularly time-consuming task. Although more open-source data can be found (e.g., crowd-sourced arrangements from www.musescore.com) and MIR algorithms for tasks such as source separation and automatic transcription are improving, both procedures are prone to high levels of error that is undesirable for either computational music analysis or training machine learning models [6]. The RS200 is still the only major corpus of expert melodic transcriptions of popular music; the pairing of these melodic transcriptions with harmonic analyses affords sophisticated analysis of tonality in popular music.

In this paper we present a corpus which extends the Billboard corpus to include expert-transcribed melodies for a sizable subset of the original corpus (214 songs presently). By adding melodic transcriptions to an existing corpus of harmonic annotations (the Billboard corpus), we create a dataset fully comparable to the RS200. We also translate both the Billboard and RS corpora into humdrum data formats, creating two comparable datasets which together form a super-corpus we call the Coordinated Corpus of Popular Music (CoCoPops). In addition to melodic and harmonic transcriptions, CoCoPops includes entirely new annotations of rhyme schemes in both subcorpora and continuous valence and arousal ratings in a 100-song subset. Like the When In Rome project [7], CoCoPops aims to facilitate musicological and MIR research by making a large body of data available in a consistent, standard format. In the sections that follow, we describe in detail the original two datasets that CoCoPops is built on, the procedures we used to generate new data, and the content of CoCoPops.

© C. Arthur and N. Condit-Schultz. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** C. Arthur and N. Condit-Schultz, "The Coordinated Corpus of Popular Musics (CoCoPops): A Meta-Corpus of Melodic and Harmonic Transcriptions", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

```
# title: Honky Tonk Women
# artist: The Rolling Stones
# metre: 4/4
# tonic: G

0.0     silence
0.424489795     A, intro, | N | N | N | N |
9.2     | G:maj | G:maj | G:maj | G:maj |
17.689795918    B, verse, | G:maj | G:maj | C:maj . . C:sus4 | C:maj |, (voice
26.048979591    | G:maj | A:maj | D:maj . . D:sus4 | D:maj |
34.351020408    | G:maj | G:maj | C:maj . . C:sus4 | C:maj |
42.579591836    | G:maj | D:maj | G:maj | G:maj |
50.669387755    C, chorus, | G:maj | D:maj | G:maj | G:maj |
58.734693877    | G:maj | D:maj | G:maj | G:maj |
66.620408163    B, verse, | G:maj | G:maj | C:maj . . C:sus4 | C:maj |
74.620408163    | G:maj | A:maj | D:maj . . D:sus4 | D:maj |
82.579591836    | G:maj | G:maj | C:maj . . C:sus4 | C:maj |
90.553151927    | G:maj | D:maj | G:maj | G:maj |
98.489795918    C, chorus, | G:maj | D:maj | G:maj | G:maj |
106.375510204   | G:maj | D:maj | G:maj | G:maj |, voice)
114.188639455   B, solo, | G:maj | G:maj | C:maj . . C:sus4 | C:maj |, (guitar
121.991836734   | G:maj | A:maj | D:maj . . D:sus4 | D:maj |
129.75510204    | G:maj | G:maj | C:maj . . C:sus4 | C:maj |
137.534693877   | G:maj | D:maj | G:maj | G:maj |, guitar)
145.255873015   C, chorus, | G:maj | D:maj | G:maj | G:maj |, (voice
153.028571428   | G:maj | D:maj | G:maj | G:maj |
160.718367346   C, chorus, | G:maj | D:maj | G:maj | G:maj |
168.440816326   | G:maj | D:maj | G:maj G:maj/3 G:maj/11 G:maj/#11 | G:maj/5 G:maj |
179.118730158   silence
182.282448979   end
```

Figure 1. Sample annotation file from the original McGill Billboard corpus ("Honky Tonk Woman," The Rolling Stones).

2. BACKGROUND

2.1 The McGill Billboard Corpus

The McGill Billboard [1] corpus contains annotations of 739 [1] unique songs, all sampled from the Billboard Hot 100 charts between 1958 (when Billboard magazine began publishing this chart) and 1991. The authors used a stratified sampling procedure to gather as representative a sample as possible, sampling a (roughly) equal number of songs from each of three "eras" (60s, 70s, 80s) while also accounting for chart position (1–100).

The McGill Billboard transcription process involved a team of more than two dozen people, included "auditions to identify musicians with sufficient skill to transcribe reliably and efficiently," and cost upwards of $20,000 [1]. The process included creating manual annotations of the chords, formal sections (e.g., verse, chorus), phrases (loosely defined), key(s), and meter in each sampled song, conducted by two independent annotators. A third "meta-annotator" compared the two versions for differences and combined them into a single, final transcription.

The McGill Billboard chord annotations are encoded using the representation scheme proposed by Harte in 2005 [8] and later expanded and revised in 2010 [2]. This representation uses a syntax that is common in popular music lead sheets, where chords are represented as a root note with a set of intervals above the root, with the most common chord types given a list of shorthand symbols (e.g., `C:maj`, `A:7`). The McGill annotations are encoded in plain-text files with line breaks representing new phrases, each line tagged with the dominant instrument (or vocals) in that phrase. An example of an original file from the McGill Billboard corpus is shown in Figure 1. [2]

2.2 The RS200 Corpus

The Rolling Stone corpus was first described in a paper published in 2011 [3], initially dubbed the RS5x20 corpus. This original 100-song corpus (RS5x20) contained harmonic annotations of the top 20 songs listed, for each of five decades from the 1950s through the 1990s, on Rolling Stone magazine's list of the "500 Greatest Songs of All Time" (as first published in 2004). The corpus was later expanded to 200 songs (the RS200 corpus), and also added melodic transcriptions for each song [4], making it the first public corpus of expert melodic transcriptions of popular music. Since the remaining 400 songs on Rolling Stone's list were not chronologically balanced, the second set of 100 songs was chosen based on rank position alone. While the Billboard charts are based on commercial sales, the Rolling Stone list was based on votes from experts (specifically, "172 rock stars and leading authorities"). Although one may suspect that these two corpora would substantially overlap, in fact there are only fifteen songs in common.

The RS200 annotations are spread over multiple separate files per song: one with the timestamps, two with the harmonic analyses (one per annotator), another with the melody transcription, and (for an 80-song subset) a fifth with lyrics. Unlike the Billboard corpus, the RS200 chords are annotated using Roman numerals; Similarly, the melody transcriptions are encoded as scale-degree annotations, with direction markers to clarify octave and contour. Rhythmic durations are not encoded at all, only the timing of note onsets: each measure of music is divided into regular steps representing metric positions, with notes placed at steps indicating onsets and dots representing empty steps. The number of steps per measure is dynamic, depending on the meter and the lowest metric position needed to represent onsets in that measure. For instance, a measure that contains only one note that arrives on the second half of the first beat (e.g., the "and of 1") requires division into eighth notes, so that measure will have eight steps with only a note at the fourth step and the rest dots. However, a measure with only a single note that lands on the downbeat can be represented with just one token. Sample files from the RS200 corpus can be seen in Figure 2.

2.3 Related Work

The most closely-related work to ours is another extension to the McGill Billboard corpus by Christopher White et al. [9], which adds timbral and textural annotations to the entire Billboard corpus. Annotators of this corpus listened to the songs and notated "all moments of change" within each track according to three broad categories: the "domain" of change (such as the instrument group, harmony, lyrics, texture, etc.); the "genera" of each change within the relevant domain (such as a change to "solo" within a texture category); and an "event type" which solely denotes one of three options: a change, entry, or exit. We intend to work with the authors for a future release of CoCoPops to incorporate this textural and timbral information as well.

A major drawback of both the Billboard and Rolling Stone samples is their overwhelming bias towards music from before 1991. Two recent projects have sought to right this imbalance by creating corpora of more modern popular music to complement the Billboard sample: White et

[1] Note that a small subset has been withheld from the public to serve as testing data for the MIREX competition.

[2] A separate set of `mirex` text-files includes only the chords, but with a timestamp for every chord.

```
1.705102040     1     [E] R . . bVII | i | . . . bVII | i | . . . bVII |     % Whole Lotta Love
4.376530612     2     i . . bVII | i . . bVII | i . . bVII | i . . bVII |
7.084693877     3     i . . bVII | i . . bVII | i . . bVII | i . . bVII |     [E] [OCT=4] R*4 | _ _ ...5 .5.. |
9.778571428     4     i . . bVII | i . . bVII | i . . bVII | i . . bVII |     .54. b3b7.. ...1 b711. | .b3.. 1b7.. ...^5 .55. | 5..4 .b3.. _ 1.1. | #23.1 b7... _ _ |
12.498979591    5     i . . bVII | i . . bVII | i . . bVII | i . . bVII |     ^45.4 ..b3. 1b7.. _ | .51b7 1..b3 ..b7. _ | .5^55 445. 4b3.. _ | .1^55 445. 4.b3. _ |
15.251020408    6     i . . bVII | i . . bVII | i . . bVII | i . . bVII |
17.953061224    7     i . . bVII | i . . bVII | i . . bVII | i . . bVII |     5... _ ..vb7b7 b7b7b7b7 | 1... _ ..b7b7 b7b7b7b7 | 1... _ ..b7b7 b7b7b7b7 | 1... _ ..b7b7 b7b7b7b7 | 1. .. .. ^55 |
20.667346938    8     i . . bVII | I | R | i | | | | | . . . bVII | i . .
23.381632653    9     bVII | i . . bVII | i . . bVII | i . . bVII | i . .     ..54 .b3.. ...4 b34.5 | ..54 b3... _ v5..^5 | ..5. .54. 4b3b34 b3b34. | 4..4 .b3.1 b7.b7. _ |
26.101020408    10    bVII | i . . bVII | i . . bVII | i . . bVII | i . .     .b7^45 .45. 5.4. b3.b7. | .51b7 11.1 b7... _ | .5^55 445. 4b3.. v5..^5 | .v5^55 445. 4b3.7 ...1 |
28.795918367    11    bVII | i . . bVII | i . . bVII | i . . bVII | i . .
31.513265306    12    bVII | i . . bVII | [0] i | | IV | [4/4] . . . bVII |     _ #45v1. _ b7b7b7b7 | 1... _ ..11 b7b7b71 | _ _ ..b7b7 b7b7b7b7 | 1... _ ..b7b7 b7b7b71 |
34.195918367    13    i . . bVII | i . . bVII | i . . bVII | i . . bVII |     R*7 |
36.840816326    14    i . . bVII | i . . bVII | i . . bVII | i . . bVII |
39.420408163    15    i . . bVII | i . . bVII | i . . bVII | i . . bVII |     .. .. .. ^55 | ...5 4.3. _ 4.45 | .b5.#2 3.1b7 .1v4. _ |
41.990816326    16    i . . bVII | i . . bVII | i . . bVII | i . . bVII |     ^1.^55 .54. 4b3.b3 4.4. | .5.4 .b3.b3 .1.b7 _ | .1^5. 5.4b3 .4.b3 ..b7. | .b71b7 1b73. 1b7.. _ |
44.546938775    17    i . . bVII | i . . bVII | i . . bVII | i . . bVII |     .1^55 #43#43 #435. #43.. | .1^55 554. .b3.b7 _ | _ ..^5. .2.. _ | ...5 .#4.3 _ _ |
47.160204081    18    i . . bVII | i . . bVII |
49.768367346    19                                                          _ _ ..vb7b7 b7b7b7b7 | 1... _ ..b7b7 b7b7b71 | _ _ ..b7b7 b7b7b71 | _ _ ..b7b7 b7b7b71 |
52.369387755    20
54.989795918    21                                                          [0] ..^5.5#45. | #23vb7.^5.5.5.4.#23v5. | .^5 |
57.593877551    22
60.202040816    23                                                          [4/4] R*4 | ..45 .v1.. _ _ | R*2 | ...^5 .4b3. 1b7.. _ | ...^4 4b344 5.4. b3... |
62.795918367    24                                                          .5.5 | .5.5 | .5.5 | .4.. | .. .. .3 .. | 3. .3 .. 3. |
65.386734693    25
67.965306122    26                                                          ...vb7 .b7.. .b7.. b7... | | ..^5#4 5.55 .#4.. _ | .b35#4 5.55 #4#4.. _ |
70.492857142    27                                                          .b3#4#4 5.55 #4#4.. _ | v1.....^1.1.b7. 5.....4..b3.. ...b7..... == | |
73.061882086    28
75.608163265    29
78.115306122    30
```

Figure 2. Sample annotation files from the RS200 corpus ("Whole Lotta Love," Led Zeppelin). The image shows three files overlaid on top of each other from left to right: timestamps of each measure, key and chord annotations, and melody transcription.

al. [10] introduce the "Millenial corpus," a dataset of expert melodic transcriptions of twenty five popular songs written between 2015 and 2019. Beach and Arthur [6] created a much larger corpus of popular songs with annotations, although their annotations are derived algorithmically from the audio, and are quite noisy.

Our aim to combine two existing corpora into a single, homogeneous dataset is inspired by Mark Gotham's "when in Rome" project [7], which merges and reformats several existing classical corpora with Roman numeral annotations into a single collection in a common format. Our project to gather valence and arousal data for the Billboard sample was similarly inspired by the DEAM dataset: a dataset containing dynamic annotations of valence and arousal for 1,809 non-copyrighted (Creative Commons license) songs and song excerpts [11]. The majority of these annotations are of short excerpts ($\approx 45_s$) across numerous musical styles (folk, world, jazz, instrumental, pop); however, the dataset also includes ratings for 56 complete songs, which provide the most valuable information, according to the creators [11]. In addition, the quality of the audio recordings (and the musical content) in the DEAM sample is highly variable, as these recordings do not represent professionally published works. Our valence- and arousal-ratings for 100 complete, successful, commercial recordings will serve as a useful complement to the DEAM sample.

3. CORPUS OVERVIEW

The CoCoPops corpus consists of two collections: the Billboard and RS200 subcorpora. Each collection contains one file per song. In the CoCoPops-Billboard collection, all 739 of the original McGill Billboard files have an equivalent humdrum file. The contents of each file, however, vary: All 739 files contain all the originally encoded information (chords, keys, formal section labels, timestamps, phrase information) from the original McGill dataset, but all converted to humdrum format, and with a significant number of corrections (see Section 5). At present, 214 out of the 739 files include new expert melodic and lyric transcriptions, as well as an encoding of the rhyme scheme; 100 of those 214 songs also contain continuous user ratings of valence and arousal, as well as rolling RMS (root mean square) amplitude values of the audio, to approximate the changing sound level of the music—both sampled at a rate of 2_{Hz}. A sample CoCoPops-Billboard file is shown in Figure 3. In the CoCoPops-RS200 collection, each file contains the information originally spread over separate files—e.g., melody, harmony, time stamps, lyrics—in a single humdrum file. Unlike the original Billboard annotations which used Harte's encoding scheme (i.e., root+quality), the RS200 were originally annotated with Roman numerals. To facilitate analysis, we provide both types of harmonic annotations in both collections. In addition, since the original RS200 contained two independent transcriptions of the harmony, each CoCoPops-RS200 file includes two Roman numeral annotations (i.e., two `**harm` spines) side-by-side. Eighty of the files also include lyrics and syllable stress information.

The humdrum syntax is a plain-text format for representing musical information, organized into tab-delineated columns—called "spines"—representing different streams of data [12] (see `www.humdrum.org` for more information). Within the general humdrum syntax, various specific representation schemes can be defined [3] : Two of the most common representation schemes include the widely-known `**kern` representation of pitch information, the `**silbe` representation of lyrics, and the `**harm` representation of harmonic information in Roman-numeral format. Other relevant representations for the present collections include `**harte`—a humdrum representation for root+quality-style harmonic annotations (near-identical to the original annotation scheme used in the McGill Billboard corpus. This scheme is based on the syntax proposed by Chris Harte [2, 8] and the humdrum representation is described in Arthur et al. [13]); and `**rhyme`—a representation for rhyme schemes [14].

In the following sections we describe our procedures for gathering new data (e.g., melodic transcriptions), and

[3] Chapter 18 of the Humdrum User Guide illustrates how to create new humdrum representations.

**kern	**silbe	**rhyme	**harm	**harte	**phrase	**timestamp	**leadinstrument
*ICvox	*	*	*M4/4	*M4/4	*	*	*
*G:	*	*	*G:	*G:	*	*	*
*M4/4	*	*	*	*	*	*	*
*clefG2	*	*	*	*	*	*	*
r;	.	.	1r;	r	.	0	r
r;	.	.	1r;	r	.	0.511	r
=1	=1	=1	=1	=1	=1	=1	=1
8r	.	.	1I	G:maj	newline	7.616	voice
8b-	My	.	.	.	.	.	.
8b-	ba-	.	.	.	.	.	.
8b-	-by	.	.	.	.	.	.
8b-	whis-	.	.	.	.	.	.
8a	-pers	.	.	.	.	.	.
8g	in	.	.	.	.	.	.
8d	my	.	.	.	.	.	.
=2	=2	=2	=2	=2	=2	=2	=2
*>Letter>A	*>Letter>A	*>Letter>A	*>Letter>A	*>Letter>A	*>Letter>A	*>Letter>A	*>Letter>A
*>Label>Verse	*>Label>Verse	*>Label>Verse	*>Label>Verse	*>Label>Verse	*>Label>Verse	*>Label>Verse	*>Label>Verse
8e	ear	A	1I	G:maj	newline	10.017	voice
8d	_	.	.	.	.	.	.
4r	.	.	.	.	.	.	.
4r	.	.	.	.	.	.	.
8r	.	.	.	.	.	.	.
[8g	Mmm	.	.	.	.	.	.
=3	=3	=3	=3	=3	=3	=3	=3
2.g]	_	.	1IV	C:maj	.	.	.
8g	Sweet	.	.	.	.	.	.
[8e	no-	.	.	.	.	.	.
=4	=4	=4	=4	=4	=4	=4	=4
8e]	_	.	1I	G:maj	.	.	.
4.d	-thin's	.	.	.	.	.	.

Figure 3. Sample file from the CoCoPops corpus. This file ("Sweet Nothins,", Brenda Lee) includes the original McGill Billboard information alongside new melody and lyric information. Files in the valence and arousal subset (see Section 6) include three additional spines.

how we converted the preexisting datasets into humdrum formats.

4. MELODY TRANSCRIPTION

In the early stages of our project, we worked with four collaborators [4] to define transcription guidelines which could be applied consistently. We elected to transcribe only vocal parts, with focus on the "lead" vocal melody in each song—however, we agreed to encode important vocal harmonies or other "backing" vocals as needed. The vocal performances in the sample are often challenging to transcribe, including unpitched or quasi-pitched vocals, "blue" notes, glissandi, loose rhythms, and syncopation. Our goal was to create readable transcriptions using conventional musical syntax (beat positions, durations, notes) rather than mechanical, empirical terms (milliseconds, F0, etc.). This requires significant interpretation and quantization; However, we took care to not over-simplify melodies such that they became melodic reductions. Our transcriptions generally interpret rhythms using a 16th-note grid, but triplets and 32nd-notes are used sparingly at slow tempos; Similarly, pitches are encoded in standard western pitch categories (e.g., C#5, B4), ignoring most glissandi and blue notes. However, many vocal performances simply cannot be faithfully represented in traditional score categories: as such, we included provisions for indicating, as needed, unpitched or approximate pitch, "free" or approximate rhythms, glissandi, and blue notes—the complete details of these encodings are documented directly in the CoCoPops repository.

Ultimately, ten individuals contributed to our 214 melodic transcriptions: 94 transcriptions by the authors; 40 transcriptions by our four early collaborators, all graduate students in music performance or theory; 10 transcriptions by three (paid) undergraduate music students; and 70 transcriptions by one (paid) professional jazz performer, also a graduate student in jazz performance at the time. When transcribers were uncertain of their transcriptions, a second transcriber would collaborate on the final version. We gave our paid transcribers detailed instructions and have personally vetted and edited all transcriptions for consistency. The complete transcription guidelines are provided in the supplementary materials.

The exact audio files used for the original McGill transcriptions are not publicly available; for our transcriptions we accessed targeted songs via YouTube, taking care to confirm that each recording was the correct Billboard Hot 100 single. Unfortunately, some of the original McGill transcriptions do *not* match the targeted single, instead matching an album version, live version, or some other version of the same song; In a few cases, we could not find any recording that clearly matched the transcription. To improve consistency, we elected to modify the harmonic transcriptions for sixteen tracks to match the correct, single version from the Hot 100 chart. In most cases, these versions were very similar but slightly longer or shorter; in a few cases, the alternate version was in a different key or contained other significant differences. For these sixteen altered versions, the original timestamps were discarded and replaced with corrected timestamps in the correct single version, as available on YouTube. The CoCoPops repository includes files with links to each song's reference recording on YouTube, as well as MusicBrainz MBIDs for our 214-song melodic transcription subset.

5. CONVERTING EXISTING DATA

To create the new data, we converted the preexisting Billboard and RS data into humdrum format. During this process, we noted some errors in the Billboard transcriptions, which we corrected in our new data. Our expertise (education/credentials) in music performance and analysis are comparable to the original transcribers'. Most of these er-

[4] Thanks to Hubert Léveillé Gauvin, Gary Yim, Dana DeVlieger, Lissa Reed.

rors are unambiguous—for instance, a measure of music missing or a clear change of key that is not indicated. In only a few cases our "corrections" might by considered debatable. All errors and corrections are documented in our corpus repository. Each file in CoCoPops also includes a wealth of meta-data, including track information—title, original artist, release date, etc.—and sampling information, like the rank on the Rolling Stone 500 list and chart position on the Billboard Hot 100.

5.1 Billboard Data

We created a custom R script to convert the original Billboard corpus files (available at ddmal.music.mcgill.ca/research) into a humdrum representation. The harmonic annotations are encoded in a `**harte` spine with the timestamps in a `**timestamp` spine. Along with the `**harte` representation, we also include a `**harm` spine in each file: the humdrum standard for representing Roman numerals. Whereas the original harmonic transcriptions focus on the literal pitch-content played by rhythm-section instruments (ignoring vocal parts), Roman numerals represent harmony at a higher level of abstraction, incorporating the broader tonal context. This means that, for example, open-fifth "power chords" are interpreted as major or minor triads (numerals) based on the key, context, and vocal melody. For illustration, the original transcription of the track "I'm Going Down," by Bruce Springsteen, consists entirely of two repeated patterns: `A:5-E:5-F#:5-D:5` and `A:maj-E:maj-F#:min-D:maj`. We interpret both of these patterns as `I-V-vi-IV`. To create this `**harm` information, we wrote an R script to parse each file and replace under-specified chords (like `C5`) with the full triad expected given the key-signature and/or explicit triads indicated on the same root in the same song. This process was effective in the vast majority of cases; however, for songs with ambiguous modality we identified the triad manually. The harmonic rhythm is also indicated in the `**harm` spine using standard humdrum rhythmic duration tokens.

The original McGill data includes two, parallel, formal encodings: named sections (e.g., verse, chorus) and abstract letters (e.g., AABA). These parallel encodings are not redundant, as the transcribers used letters to indicate more abstract repetition (mainly of chord progressions)—for example, a guitar solo section which reuses the chord progression from the verse will be labeled "solo", but use the same letter designation (e.g., `A`) as the verse. We encode both formal representations, independently, in hierarchical https://www.humdrum.org/guide/ch20/: Abstract formal labels are encoded in interpretation records of the form `*>Letter>A`; formal names are encoded in separate records of the form `*>Label>Verse`. Phrases in the music (originally represented with line breaks) are indicated by the presence of the token `newline` in a `**phrase` spine, with a parallel `**leadinstrument` spine for lead-instrument annotations.

Transcribers worked in music notation software of their choice (e.g., Musescore, Sibelius) transcribing pitch, rhythm, and lyrics. The transcription was then exported into musicXML format. We wrote a Haskell program to parse musicXML scores into humdrum notation (`**kern` for pitch/rhythm, and `**silbe` for lyrics), and align this information with the already generate humdrum data described in the previous paragraphs. When transcribers included more than one vocal part for a song, each part appears as a separate pair of spines(`**kern` and `**silbe`) in the humdrum file.

5.2 Rolling Stone Data

The RS200 dataset is available at rockcorpus.midside.com, with data for each song encoded in four or five separate files—Figure 1 shows three such files. In addition, David Temperley provided us with files indicating the hierarchical structure built into their original transcriptions, which can be interpreted as formal labels. We created a Haskell program to parse these files and generate a single humdrum-syntax file for each track.[5] In some cases, we had to correct inconsistencies between harmonic and melodic transcriptions—e.g., music notated as 4/4 in the harmonic analysis but 12/8 in the melodic transcription. Each humdrum file created includes two `**harm` spines, representing Temperley and de Clercq's separate harmonic transcriptions, labeled with comment tokens '!D.T.' or '!T.d.C' respectively. The RS200's original step-sequencer-like approach to rhythm transcription is faithfully encoded using humdrum's "timebase" function where `*tb` interpretations indicate the duration of each step. For the 80-song subset with lyrics, a `**silbe` spine indicates the lyric alongside a `**stress` spine to indicate three levels of lexical/prosodic stress.

The original RS200 transcriptions indicate only tonal center (tonic), not mode, which can be ambiguous in popular music [3]. For consistency with the Billboard data, the key in each `**harm` spine is indicated as either major or minor, depending on what would be the most likely interpretation. The RS200 melodic transcriptions *do* include key-signature-like indications of raised/lowered scale degrees. Using these scale indications and the humdrumR package [16], we were able to convert the original scale-degree representation to `**kern` in the final dataset.

6. VALENCE AND AROUSAL SUBSET

In addition to the musical data itself, we gathered continuous-response ratings of perceived valence and arousal, in a 100-song subset of the Billboard data. Valence and arousal are the two core dimensions of Russel's circumplex model of affect [17], and, while perhaps limiting [18, 19], has been used widely in both music perception research [18, 20] and music emotion recognition (MER) [21–23]. We focused on valence and arousal due to their simplicity (i.e., only two variables) and ubiquity

[5] Though the music21 Python library [15] includes a parser for the RS200 harmonic transcriptions, it was easier to assure consistency and alignment between melodic, harmonic, lyrical, and formal information by using a single custom parser.

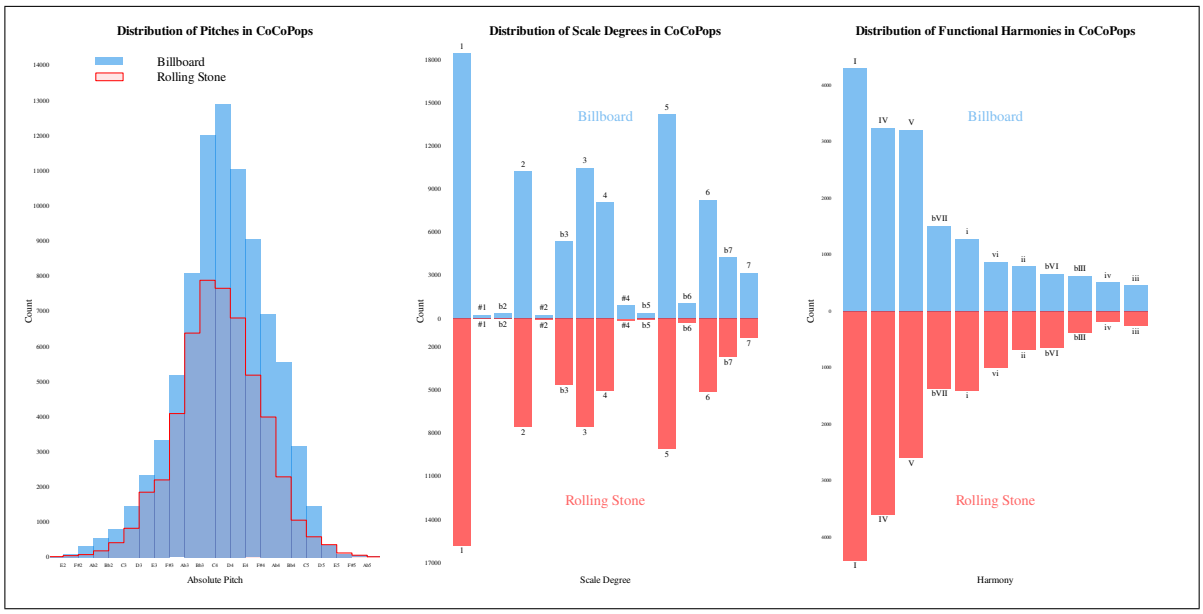


Figure 4. Left: distribution of absolute pitches in each corpus. Middle: distribution of 15 most common scale degrees in each corpus. Right: distribution of ten most common functional harmonies in each corpus (11 in total), sorted by rank in the Billboard data. (Only Temperley's harmonic annotations are counted; Immediate repetitions of a chord are not counted.)

in the literature, though it is acknowledged that there are likely additional, overlooked dimensions such as tension and power [24]. Since arousal is highly correlated with sound level, we also include the rolling RMS values for each track in an `**rms` spine.

6.1 Perceptual Data

Perceptual data was gathered in a human-subject experiment, approved by Georgia Tech's Institutional Review Board (protocol H22086). Eighty participants took part in our experiment, each paid $15 for their time. All participants were students at Georgia Tech, and were mainly non-music majors. Experiments took place in person, in a sound-attenuated booth using professional-quality loudspeakers set at the same fixed sound-level for all participants. Participants used a physical slider (Monogram Creative) to make their continuous ratings, with the slider position sampled every 500_{ms}.

The concepts of valence and arousal were explained to each participant in simple terms: arousal being how calm-energetic they perceived the music to be at any given moment, and valence being the polarity (negative-positive) of the music [25]. Participants were instructed to rate what they perceived the music to express, not necessarily what they themselves felt. Since continuously tracking valence and arousal simultaneously is challenging, we had participants rate each independently—the same approach taken for the DEAM dataset [11]. The authors of the DEAM project also reported an increase in the usability (i.e. variation) [11] of the ratings when they used full songs as opposed to shorter clips; Accordingly, participants in our experiment listened to the full songs. To encourage sustained engagement and attention throughout the experiment, we had each participant rate only ten songs. Participants were randomly assigned to rate valence in five songs and arousal in the other five, with the order of tasks counterbalanced. Ultimately, each of the 100 songs was independently rated for valence and arousal by eight different participants (four for valence and four for arousal). The full experiment took approximately forty minutes.

Files in the 100-song subset include independent `**valence`, `**arousal`, and `**rms` spines. The four independent arousal and valence ratings are encoded in the same spine, in space-separated humdrum sub-tokens.

7. SUMMARY

The CoCoPops corpus includes complete melodic and harmonic data for 398 unique popular songs released between 1949 and 2002. 95% of songs (379) come from the years 1956–1991 with more than half (203) from the years 1965–1980. The corpus includes 145,822 note onsets (86,215 in the Billboard subset), 37,010 chord changes (19,682 in the Billboard subset), and 63,809 words in the lyrics (48,018 in the Billboard subset). Figure 4 shows the distributions, in each subcorpus, of three fundamental pitch parameters—absolute pitch height, scale degree, and the ten most frequent Roman numerals. Though the two subcorpora originate in data generated by different sampling criteria and different measurement/encoding procedures (see Section 5), these distributions are nonetheless broadly similar, highlighting the potential value of treating these two separate subcorpora as a single united corpus.

The CoCoPops dataset is hosted at github.com/Computational-Cognitive-Musicology-Lab/CoCoPops, shared under a CC-BY-4.0 license. Many further methodological and encoding details are included in the repository files, as well as our recommendations about the usage, distribution, and citation of the data.

8. REFERENCES

[1] 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*, A. Klapuri and C. Leider, Eds., Miami, FL, 2011, pp. 633–638.

[2] C. Harte, "Towards automatic extraction of harmony information from music signals," Doctor of Philosophy, University of London, Department of Electronic Engineering, Queen Mary, University of London, 2010.

[3] T. De Clercq and D. Temperley, "A corpus analysis of rock harmony," *Popular Music*, vol. 30, no. 1, pp. 47–70, 2011, publisher: Cambridge University Press. [Online]. Available: https://www.cambridge.org/core/journals/popular-music/article/abs/corpus-analysis-of-rock-harmony/C5210A8EC985DDF170B53124F4464DA4

[4] D. Temperley and T. d. Clercq, "Statistical analysis of harmony and melody in rock music," *Journal of New Music Research*, vol. 42, no. 3, pp. 187–204, 2013. [Online]. Available: https://doi.org/10.1080/09298215.2013.788039

[5] D. Huron, "On the virtuous and the vexatious in an age of big data," *Music Perception: An Interdisciplinary Journal*, vol. 31, no. 1, pp. 4–9, 2013. [Online]. Available: https://www.jstor.org/stable/10.1525/mp.2013.31.1.4

[6] B. Clark and C. Arthur, "Is melody 'dead'?: A large-scale analysis of pop music melodies from 1960 through 2019," *Empirical Musicology Review*, In Press.

[7] 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 (TISMIR)*, vol. 3, no. 1, p. 42, 2020. [Online]. Available: https://hal.science/hal-02934374

[8] C. Harte, M. B. Sandler, S. A. Abdallah, and E. Gómez, "Symbolic representation of musical chords: A proposed syntax for text annotations." in *Proceedings of the International Society for Music Information Retrieval*, 2005, p. 66–71.

[9] C. W. White, J. Fulmer, B. Cordova, A. Black, C. Danitz, W. Evans, A. Fischer, R. Greene, J. He, E. Kenyon, J. Miller, M. Moylan, A. Ring, E. Schwitzgebel, and Y. Wang, "A new corpus of texture, timbre, and change in 20th-century American popular music," in *Future Directions of Music Cognition*. The Ohio State University Libraries, 2021. [Online]. Available: https://kb.osu.edu/handle/1811/93133

[10] C. W. White, J. Pater, and M. Breen, "A comparative analysis of melodic rhythm in two corpora of American popular music," *Journal of Mathematics and Music*, vol. 16, no. 2, pp. 160–182, 2022. [Online]. Available: https://doi.org/10.1080/17459737.2022.2075946

[11] 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. [Online]. Available: https://dx.plos.org/10.1371/journal.pone.0173392

[12] C. S. Sapp, "Online database of scores in the humdrum file format." in *Proceedings of the International Society for Music Information Retrieval*, 2005, p. 664–665.

[13] C. Arthur, F. Lehman, and J. McNamara, "Presenting the SWTC: A symbolic corpus of themes from John Williams' *Star Wars* episodes I–IX," *Empirical Musicology Review*, In Press.

[14] N. Condit-Schultz, "MCFlow: A digital corpus of rap transcriptions," *Empirical Musicology Review*, vol. 11, no. 2, p. 124–147, 2016.

[15] M. S. Cuthbert and C. Ariza, "Music21: A toolkit for computer-aided musicology and symbolic music data." in *Proceedings of the International Society for Music Information Retrieval*, 2010, p. 9–13.

[16] 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*, 2019, p. 715–722.

[17] J. A. Russell, "A circumplex model of affect," *Journal of Personality and Social Psychology*, vol. 39, no. 6, pp. 1161–1178, 1980, place: US Publisher: American Psychological Association.

[18] A. Micallef Grimaud and T. Eerola, "An interactive approach to emotional expression through musical cues," *Music & Science*, vol. 5, p. 20592043211061745, 2022, publisher: SAGE Publications Ltd. [Online]. Available: https://doi.org/10.1177/20592043211061745

[19] G. L. Collier, "Beyond valence and activity in the emotional connotations of music," *Psychology of Music*, vol. 35, no. 1, pp. 110–131, 2007.

[20] 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. [Online]. Available: http://mp.ucpress.edu/cgi/doi/10.1525/mp.2012.30.3.307

[21] Y. Agrawal, R. G. R. Shanker, and V. Alluri, "Transformer-based approach towards music emotion recognition from lyrics," in *Advances in Information Retrieval*, ser. Lecture Notes in Computer Science, D. Hiemstra, M.-F. Moens, J. Mothe, R. Perego,

M. Potthast, and F. Sebastiani, Eds. Cham: Springer International Publishing, 2021, pp. 167–175.

[22] J. de Berardinis, A. Cangelosi, and E. Coutinho, "The multiple voices of musical emotions: source separation for improving music emotion recognition models and their interpretability," in *Proceedings of the 21st International Society for Music Information Retrieval Conference*, Online, 2020, pp. 310–317. [Online]. Available: https://livrepository.liverpool.ac.uk/3105085

[23] 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 *Proceedings of the International Society for Music Information Retrieval*. International Society for Music Information Retrieval (ISMIR), 2022, pp. 650–658.

[24] J. Cespedes-Guevara and T. Eerola, "Music communicates affects, not basic emotions – A constructionist account of attribution of emotional meanings to music," *Frontiers in Psychology*, vol. 9, 2018. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fpsyg.2018.00215/full

[25] K. N. Olsen, R. T. Dean, and C. J. Stevens, "A Continuous Measure of Musical Engagement Contributes to Prediction of Perceived Arousal and Valence," *Psychomusicology: Music, Mind, and Brain*, vol. 24, no. 2, pp. 147–156, 2014.

TOWARDS COMPUTATIONAL MUSIC ANALYSIS FOR MUSIC THERAPY

Anja Volk[1] **Tinka Veldhuis**[1] **Katrien Foubert**[2] **Jos de Backer**[2]
[1] Department of Information and Computing Sciences, Utrecht University, the Netherlands
[2] Faculty of Medicine, KU Leuven, LUCA School of Arts, Belgium
a.volk@uu.nl, tinkaveldhuis94@gmail.com, {katrien.foubert,jos.debacker}@kuleuven.be

ABSTRACT

The research field of music therapy has witnessed a rising interest in recent years to develop and employ computational methods to support therapists in their daily practice. While Music Information Retrieval (MIR) research has identified the area of health and well-being as a promising application field for MIR methods to support health professionals, collaborations with experts in this field are as of today sparse. This paper provides an overview of potential applications of computational music analysis as developed in MIR for the field of active music therapy. We elaborate on the music therapy method of improvisation, with a particular focus on introducing therapeutic concepts that relate to musical structures. We identify application scenarios for analysing musical structures in improvisations, introduce existing analysis methods of therapists, and discuss the potential of MIR methods to support these analyses. Upon identifying a current gap between high-level concepts of therapists and low-level features from existing computational methods, the paper concludes further steps towards developing computational approaches to music analysis for music therapy in an interdisciplinary collaboration.

1. INTRODUCTION

The use of music technology in the context of health and well-being is becoming increasingly important, in line with a growing interest in eHealth in medicine. Music's affordances such as emotion regulation [1], motor coordination [2], and social interaction [3], enable a broad range of therapeutic applications. They feed into research on developing music technology for various contexts of music therapy (MT) such as for supporting motor and cognitive rehabilitation through musical biofeedback [4] and through music-based applied games [5, 6], or through digital musical instruments developed for specific patient groups [7]. For a broad overview on different use cases of music technology for music therapy we refer to [8].

One of the main application fields envisioned for music technology in the context of health and well-being is the support of data analysis from therapeutic sessions, including analysis and visualizations of musical structures. The computational analysis of musical structures has been one of the main research topics of music information retrieval (MIR) over the past decades. While MIR has identified the health context as one of its future challenges [9], there exist only few attempts to employ MIR methods for the analysis of musical structures in the context of MT to date [10–15]. The goal of this paper is to provide an introduction and overview on how MIR methods for computational music analysis can be of use for active music therapy (employing music making), specifically for analysing musical improvisations from therapy sessions to support music therapists.

The contributions of this paper are threefold: First, it provides an overview of the different contexts in which music therapists analyse musical material from improvisations, and of their analytical approaches (Section 2). Second, it identifies and describes different scenarios in MT which can benefit from computational methods on music analysis. (Section 3). Third, it identifies a current gap between high-level concepts of therapists and low-level features of current computational approaches, and concludes collaboration perspectives for MIR and MT researchers on developing computational approaches to musical structure analysis of clinical improvisations (Section 4).

2. OVERVIEW ON MUSIC THERAPY

2.1 What is music therapy?

The American Music Therapy Association describes music therapy (MT) as "the clinical and evidence-based use of music interventions to accomplish individualized goals within a therapeutic relationship by a credentialed professional who has completed an approved music therapy program" [16]. In a music intervention, therapists create musical experiences in a holistic manner involving the patient's cognition, emotion, movement and social interaction, to approach issues faced by their patients. Music therapists theorize that musical processes are correlated with psychological processes [17, 18]: a musical change can indicate a change in the patient's inner state or in the interrelation between the patient and others. For instance, if a patient with ADHD learns to focus during MT, or a patient with Parkinson learns to have more control over their body while playing music, these improvements can be generalizable to other areas in their lives, because of the interdependence of human functioning [17]. For an overview on the

© A. Volk, T. Veldhuis, K. Foubert and J. de Backer. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** A. Volk, T. Veldhuis, K. Foubert and J. de Backer, "Towards Computational Music Analysis for Music Therapy", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.*, Milan, Italy, 2023.

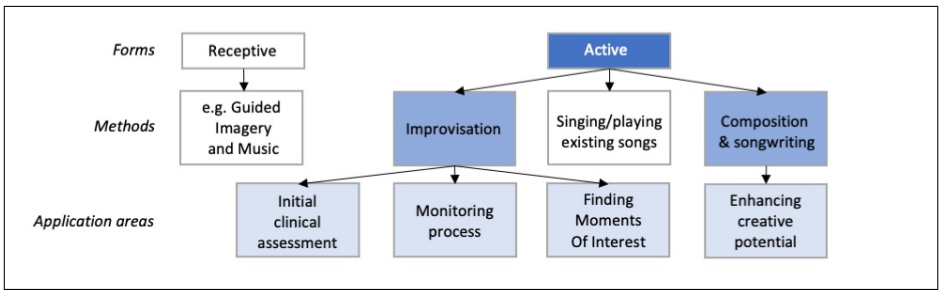


Figure 1. Main forms and methods of music therapy with application areas for computational music analysis. This paper focuses on the areas depicted by coloured blocks.

various affordances of music for therapeutic use, and clinical and non-clinical contexts of music intervention see [8].

2.2 Active music therapy

MT is divided into active (music making) and receptive (music listening) MT [19], see Figure 1. A common receptive method is called *Guided Imagery and Music* (GIM) [20], in which the patient listens to music in a relaxed state. The therapist guides the patient in bringing up the imagery that emerges from their inner process in response to the music, to explore their inner conflicts.

In active MT, creative methods such as improvisation, composition and songwriting are employed, as well as recreative methods such as singing an existing song. Creative methods are used to unravel underlying psychological patterns [21]. For instance, if the patient acts mainly as the follower in the interaction with the therapist during a musical improvisation (i.e. only the therapist initiates changes in the music), this behaviour can help to unravel interaction patterns and corresponding associations in daily life interactions of the patient.

In improvisation methods, the patient plays or sings music that they are creating themselves, either alone, with the therapist or in a group. This paper focuses on the setting of therapist and patient playing together. During a MT session, the therapist is focused on creating the music together with the patient (and the verbal evaluation of it), using musical interventions, such as changes in one or more musical parameters like timbre, dynamics, or timing, to encourage changes in the playing style of the patient. After the session, the therapist listens to the recording of the session and seeks to answer questions such as: In what way does the patient interact? Where did it feel like we were in a flow together (instead of the patient just playing for themselves) and what type of intervention caused this? In which musical parameters is the patient very rigid and what interventions change that? The therapist then seeks to draw parallels to other aspects of the patient's life and could ask the patient in the next session if the way they interact and react to the music is the same in other situations in their life. After this verbal reflection, these habits can be further explored when improvising again. In this way, therapist and patient try to slowly break out of typical habits in the process of consecutive MT sessions.

In composition methods, the patient uses their improvisations to subsequently compose music. This could be done for example by starting with a musical improvisation, then selecting from the improvisations the parts or motifs the patient finds most interesting to use in a composition, then improvising again using these motifs, and so forth, hence employing an iterative approach. This fosters interpersonal trust through a joint process working towards an explicit artistic product [22]. In song writing, the use and analysis of lyrics is important, in composition the focus is on using musical parameters and structure.

2.3 Analysis of music therapy improvisations

There exist many different approaches to analyse MT sessions. In some of them, therapists analyse only the behaviour of the patient, and not the created music. Approaches that do analyse musical structures are called *music-centered approaches*, such as the Nordoff-Robbins method [23, 24]) and the so-called *psychodynamic approach* [25]. This paper focuses on the psychodynamic approach, which suggests that producing music can help accessing the unconscious mind such that the patient's underlying issues will surface within a musical improvisation. While analysing musical structures can be useful in all MT methods and approaches, e.g. in receptive methods pattern discovery could be helpful to investigate whether specific patterns contribute to what patients prefer to listening to in specific contexts, we will focus in this paper on active, creative MT methods (see Figure 1).

Bruscia, one of the pioneers for analysis of MT improvisations (MTI), created the so-called *Improvisation Assessment Profiles* (IAPs) [26], for which the therapist fills out questionnaires based on their observations. The IAP consists of six different profiles (called *Autonomy, Integration, Tension, Variability, Salience and Congruence*). For example, the Autonomy profile explores the intermusical relationship between patient and therapist, which could show that the patient is a leader or a follower, where the patient does or does not initiate changes in the music when playing together. The therapist can observe this relationship in different musical dimensions, such as in rhythm, melody, or harmony, but also in lyrics [27]. The Tension profile shows how much tension is created through different aspects of the music, relating to questions such as: is the tempo or modality calm, or tense [28]? For a detailed description of Variability, Salience, and Congruence, we refer to [17, 28].

Therapists typically do not use all profiles and dimensions, but focus on one profile and fill out the questionnaire for all musical dimension or analyse one musical dimension in all six profiles [10]. They choose the profile and parameters based on the context: for instance, with MT for a child with ADHD, it could be relevant to analyse the presence of hyper-activity and how this changes over time, for which the therapist could use the Tension profile.

The profiles describe high-level concepts from Bruscia [17] that require fine-grained descriptions on how they might relate to concepts of musical structure for MIR researchers. For instance, when can a tempo be described as tense? According to [10, 29, 30], there exists a variety of other methods for analysing MTI, see e.g. [24, 31–33], which all address various high-level MT concepts (such as autonomy or tension). In general, these analyses are carried out by the therapist after the MT session by listening to the recorded music, using different analytical methods and the implicit musical knowledge of the therapist. There exists no systematic research on how music therapists use the different music analysis methods in their practice, as music analysis methods in MT are a particularly under-researched area according to [34].

3. APPLICATION AREAS OF COMPUTATIONAL MUSIC ANALYSIS IN MT IMPROVISATIONS

Computational methods for music analysis can be employed for several areas within the MT domain. In this section, we describe for the creative methods within active MT (namely improvisation, composition and songwriting) the following different application areas: initial clinical assessment; monitoring process; finding of so-called Moments of Interests; and enhancing the creative potential of patients within composition processes.

3.1 Psychological Assessment

Psychological assessment is the collection and analysis of information of a patient, resulting in hypotheses about the nature and causes of a patient's personality, condition, resources and potentials. In the context of MTI, the information includes musical data.

Initial clinical assessment. Hypotheses which are formed in the psychological assessment are used to determine an effective treatment program [10], considering the skills the patient currently has and what kind of therapy would fit them. Data gathered in the psychological assessment during the first therapy session is used to determine if the patients' symptoms are consistent with the diagnostic criteria for a specific mental disorder [35]. Computational analyses of the improvisations in the first therapy session can support the initial clinical assessment. For instance, computationally analysing musical timing parameters of clinical improvisations can be promising in diagnosing Borderline Personality Disorder [14].

Monitoring process. Therapists use the data gathered with psychological assessment in later stages of the therapy for monitoring the process of the patient during treatment, such as for detecting any form of progress or development of the patient. One existing approach for assessing this process is the so-called *microanalysis* [30] which focuses on small changes in social, musical, and emotional behaviour and experiences within one MT session. A computational tool could assist in performing the microanalysis on the musical content on aspects such as identifying musical dimensions of the patient's improvisations that, for instance, contain many repetitions for assessing the degree of rigidity in the playing style; identifying aspects of interventions of the therapist that caused changes in the patient's playing style on a micro level; determining the dimensions which had the greatest influence on the musical change observed in the improvisation. Gathering these insights over different sessions helps to establish what is typical of a patient's improvisational style and how it changes over time as a result of the therapeutic interventions.

Moments Of Interest. Effects of MT are often seen in specific moments within one musical improvisation session. When carrying out psychological assessment, therapists seek to identify these specific moments which can be turning points in the development of a patient. The focus of the therapist's analysis is to identify the so-called *Moments of Interest* (MOIs) [21], though there exist many other terms for MOIs, such as meaningful moments [36], pivotal moments [37], and present moments [38].

MOIs are chosen by therapists based on what they recognise as an important change [30]. It could be a mistake (e.g. patient accidentally plays an unintentional note), a mis-attunement between therapist and patient (e.g. patient does not listen to the therapist's playing which leads to unsynchronized notes), a refreshing new harmonic chord, etc. The musical events right after this moment are also of interest, since the therapist notices if the change leads indeed to something new within the improvisation (e.g. if a moment of interaction occurs between patient and therapist where they dissolve the mistake or continue on the new chord). MOIs are not described by one single form of musical change, different musical elements could be of importance in the identification of MOIs for individual diseases and patients. For instance, for patients with psychosis it could be an important change if they stop playing repetitively [39], and for borderline patients it could be an important change if they start alternating between leading and following the therapist in the improvisation [40].

3.1.1 Case study: playing styles for psychosis patients

A specifically interesting example for the potential application of computational music structure analysis, is the identification of different playing styles within MTI of patients with psychosis, including the finding of specific MOIs, namely *Moments of Synchronicity* (MOS). The spectrum of different playing styles as identified in [41] ranges from so-called *sensorial play* to *complete musical form*. In between arise Moments of Synchronicity.

Sensorial play describes a style consisting of repetitive and monotonous, or chaotic and fragmented play, with a lack of phrasing, silences and dynamics. This style is typ-

ical for patients with psychosis who are perceptually and emotionally detached from their improvisation and are not really engaging in the music, leading to an absence of interaction between patient and therapist. Typical characteristics of sensorial play are, e.g., random playing (tonal and atonal), and a significant lack of variation.

Musical form denotes a playing style that is situated on the other end of the range of observed playing styles in patients with psychosis compared to sensorial play. It arises from an inter-subjective phenomenon between patient and therapist, where both engage in a musical interaction. They experience being autonomous and equal, and are able to introduce their own new musical ideas to the improvisation. It is characterized as an improvisation where dynamical differentiation, pulse, phrasing, pauses, repetition, variation, rhythmic and melodic themes, and especially interaction between players can be observed. A clear beginning, ending and development can be identified.

Moments of Synchronicity (MOS). For patients with psychosis, the goal is to progress from sensorial play to musical form during several therapy sessions. In between, MOS between patient and therapist need to be established. These are short moments where both players have a shared feeling of an autonomous and free playing style, constituting a moment of musical interaction. Often MOS are fuelled by interventions from the therapist. In these moments, attunement/synchronicity in the musical parameters of the patient and therapist can be observed and some variation, dynamics and phrasing emerge. A pulse, combined with accents in the meter, are shared. MOS enrich the therapeutic relationship, creating moments of trust, which enable the patient to take more risks in the music playing, which in turn leads to changes in the patient. MOS are the most basic form of a MOI; after these first interactions, new interactions can emerge (see [41,42] for a detailed descriptions of the playing styles and MOS).

In sum, MOS denote specific MOIs for patients with psychosis, marking their transition from sensorial play to musical form. Recognizing these different playing styles identified by the extent as to how much musical structures are present in a MTI, delivers an interesting case study for MIR on musical structure analysis and pattern discovery.

3.2 Enhancing the creative potential for composing

The composition methods for enhancing the creative potential of patients as part of active music therapy (see Figure 1) offer a particularly interesting field of application for automatic pattern discovery. For creating compositions, patients start with improvising music. Afterwards, the patient and therapist listen to the music together and seek to find parts that the patient would like to use for composing their own musical piece. The therapist writes down the motifs they hear in musical notation, which can be time consuming. A pattern discovery tool could be used to identify and highlight all moments where the patient repeated their motifs, and preferably these motifs could be automatically transcribed to musical notation so that in the next phase of the composition these motifs could be used immediately.

4. DEVELOPING MIR METHODS FOR MT

4.1 Existing Computational Approaches in MT

A number of computational approaches have been developed to support psychological assessment in MTI, which we summarize in this section. According to [10,12], Computer Aided Music Therapy Analysis System (CAMTAS) was the first attempt to organize and analyse audio and video data specifically from MT. Developed during the mid-90s, it was used for uploading recorded data and playing back audio and video files simultaneously. Another annotation tool, the so-called Music Therapy Analysing Partitura (MAP) [29], helps to annotate events in MT based on therapist's manual transcriptions. The therapist can annotate the auditory material from a session, including the music itself, but also e.g. talking, silence, crying, and laughing, using a visual format with fixed graphical codes, allowing the therapist to view the content of one improvisation or over a whole session. Both CAMTAS and MAP rely on manual work by the therapist without any automatic detection of events from music recordings, hence using these tools is rather time-consuming [10].

Computational tools for MTI which analyse the musical content are the Music Therapy Logbook [12] and the Music Therapy Toolbox (MTTB) for MatLab [15,27,43]. The Music Therapy Logbook was developed in collaboration between MIR and MT researchers. It can be used to gather evidence of changes in a patient‘s and therapist‘s use of music over time for psychological assessment. In a proof-of-concept study using simulated MTI where one expert would improvise in the role of the patient and the other as the therapist [12], existing MIR techniques e.g. for the detection of tempo changes or the identification of rhythmic patterns, have been employed. It was tested whether computational methods can assist in evaluating whether therapist's tempo changes are effective in increasing the patient's tempo flexibility. While it was possible to identify, for instance, call and response type of play between therapist and patient, it was concluded that addressing higher-level concepts about musical interactions with computational means has yet to be fully explored in the future.

The MTTB tool takes MIDI files of therapist and patient as input and automatically extracts musical features, which it outputs into graphs depicting both the therapist and patient over time. The musical features in the MTTB are based on the Autonomy profile of the IAP [27] described in section 2.3. For instance, the density graph is calculated by averaging the number of notes played in a given time window. Since theory suggests that increasing musical density is a sign of increased arousal and therefore increased emotional and physiological density [44], density should be clinically relevant [43]. By manually reading and interpreting the two graphs of the therapist and patient, the role-relationship can be determined for the feature. A pilot study [45] investigated how the MTTB could support clinical assessment from improvisations when combined with subjective experiences of the participants, delivering first insights on how this tool might be used in the future

for investigating Bruscia's improvisation techniques, such as imitation and synchronisation through specific musical parameters. The MTTB is still under development.

For monitoring the process of patients over several therapy sessions with computational methods, the concept of *Musical Profiling* was introduced in [10], comprising three parts: *Typical Performance* for establishing a patient's individual typical playing style, *Temporal Evolution* for measuring the changes in different features in the improvisations over some time or over different sessions, and *Individual Tendencies*, measuring relations between features that are specific to that patient. In a case study with 6 participants, they used e.g. averages of the features of duration, note count, tempo, pulse clarity, dynamic centroid, and pitch centroid, to characterize a typical performance. The Musical Profiling concept is intended to contribute to establishing a systematic method of measuring and representing musical processes in improvisations in order to formalize assessment methods. To the best of our knowledge, this concept has not yet been set into practice.

In sum, there exist promising first steps in developing computational approaches to support the psychological assessment of the therapist when analysing clinical improvisations. They are not yet ready for use in clinical practice. Linking high-level MT concepts (such as moments of interest, tension, salience) to elements of the musical structure that can be extracted with computational features from the music, is as of today not a solved problem (see [11, 46] for studies on linking computational features to clinical improvisations). Moreover, from the perspective of their practical use, tools like MATLAB are not easy to use for all music therapists, and while MIDI is useful in the MT research context, most clinical contexts work with audio.

4.2 Musical structure analysis and pattern discovery

For analysing the musical content of clinical improvisations, MIR methods for musical structure analysis [47–51] as well as pattern discovery [52–64] could be of potential use in the different application fields within MT described in section 3. Techniques to identify coherent segments using concepts such as homogeneity, novelty, repetition, or regularity, developed in music structure analysis [47, 48], might be useful for describing structures emerging in clinical improvisations. Pattern discovery methods could support the identification of different playing styles such as sensorial play or musical form, taking into account the amount and kind of repetition and variation in musical patterns identified. However, these methods have been developed for different scenarios and styles, such as for popular music, jazz, classical music or folk songs. It needs to be explored in how far these techniques need to be adapted for improvisations in the MT context; e.g. in [13] it has been shown that repeated musical patterns identified in MTI were rather different from patterns typically investigated in musicological analyses of compositions and corpora.

Apart from the difference in the musical material, the analysis process of music therapists differs from musicological investigations of compositions. Typically, the therapist has participated in the improvisation and analyses afterwards the recorded musical material, taking into account their own experience during the improvisation, which might already steer the attention to certain elements of the structure. This is different from a musicological analysis of musical material that has been produced by other musicians (or musical novices).

For adapting MIR methods to the context of MT, the typical high-level concepts addressed by music therapists need to be investigated and deconstructed collaboratively in order to establish how these concepts are manifested in musical features and structures that can be described by computational means. Proof-of-concept studies such as [10, 12] provide a promising start into applying MIR features for analysing MTI. Yet, in order to develop meaningful computational features for the working context of therapists, their implicit knowledge employed in analysing clinical improvisations needs to be made more explicit. One example is given in [34] employing interviews with therapists to determine implicit and explicit knowledge in music analysis of MTI. Working towards the explicating of this implicit knowledge through collaboration would also contribute to establish how much agreement exists between different therapists using the same terminology and analysis methods. This constitutes an important step not only for developing computational methods, but also in developing assessment methods that support the development of evidence-based methods in MT.

4.3 Collaboration perspectives for computational approaches to musical structure in MT improvisations

In MIR, there exists a strong tradition of collaboration with domain experts on investigating specific musical concepts for enabling computational modeling, such as on Leitmotifs [65–67], on cadences [68–70], on similarity of folk songs belonging to a tune family [71, 72], or on melodic schemata and patterns of a certain musical style [54, 55, 72–74]. The establishment of data sets and annotations of experts regarding these concepts has been a crucial factor for enabling collaboration. We expect this to be a necessary step also for developing computational approaches to music analysis for MTI. In the following we indicate examples for the envisioned collaboration for the applications described in section 3.

Initial clinical assessment. For developing computational methods for the analysis of improvisations within the initial clinical assessment, data sets and descriptions of typical playing styles for specific patient groups need to be established. For instance, an overview of which profiles and musical dimensions therapists typically select for specific patient groups within their manual assessment using IAPs, could serve as a starting point for explicating therapists' knowledge on how to describe playing styles using musical features. In the future, once computational methods have been established, they could support therapists to initially scan *all* profiles with the help of computation, instead of manually selecting a few, ensuring that nothing has been overlooked before concentrating on selected aspects.

Monitoring process. Comparing improvisations from different sessions in order to monitor the therapeutic process requires a data-rich approach to musical structure for which computational methods are specifically apt. The concept of Temporal Evolution within Musical Profiling [10] is a first step to monitor process, using low-level musical features such as note count or pitch centroid. For getting closer to the MT practice, the high-level concepts of music analysis in MT need to be connected to appropriate models in MIR, such as identifying recurring patterns and the amount of repetitiveness and variation observed in these patterns, and the comparison of features over time.

Finding MOIs and identifying playing styles. For an overview on envisioned steps in the collaboration between MIR and MT researchers on identifying MOIs as important changes in the therapeutic process, we refer to the discussion in [8] on the creation of datasets and annotations, and the use of automatic pattern discovery and information theory. For the specific case of MOS as a progress from sensorial play to musical form (see Section 3.1.1), the identification of emerging synchronicity in the musical parameters of patient and therapist requires the modeling of musical structure as emerging from an interaction. For instance, MIR models on rhythm and meter could be adapted for detecting the establishment of a shared pulse in MOS, requiring a data collection with improvisations exhibiting different degrees of stability and variability in temporal structures with annotations on MOS as identified by therapists. For distinguishing different playing styles, such as sensorial play and musical form as described in section 3.1.1, computational methods for identifying musical structure, repetition and variation along different musical dimensions, can be employed.

Enhancing the creative potential of patients for composing. Automatic pattern discovery has been successfully employed in MIR for the automatic generation of music [75, 76]. For supporting the composition method as part of active MT, pattern discovery could assist in enhancing the creativity of the patient in the iterative process of generating a composition from improvisations. For discovering appropriate motifs in the patient's improvisations, the methods need to be able to find non-exact matches that might be perceptually meaningful. Appropriate visualization methods for displaying identified matches could support patients in choosing which matches they consider meaningful to work with. Automatic music generation systems such as [77] could be used to explore whether they might support enhancing the patient's creativity in MT (see the discussions in [78] on AI and musical creativity, and [8] on automatic music generation specifically for MT).

We conclude the following steps for establishing the collaboration between MIR and MT on developing computational methods for music analysis of improvisations:

- Investigate music analysis methods of therapists and create data sets with clinical improvisations and music analytic annotations from therapists. Using simulated therapy sessions as in [12, 45] letting therapists imitate typical playing styles of patients is a first step, yet its usefulness is limited by providing only stereotypical examples.
- Establish a catalogue of typical intervention methods of therapists in clinical improvisations as a first step on finding appropriate musical features for developing analytical methods to musical interactions in MTI.
- Explore computational approaches from MIR for the automatic detection of musical structures in MTI; start with assessing existing MIR features, and determine suitable adaptations for supporting therapists' analytical concepts. Taking into account the granularity of music information required, determine in which contexts symbolic and/or audio formats are appropriate.
- Initiate case studies: For exploring the emergence of musical structures in MTI, identifying different playing styles of patients with psychosis as in Section 3.1.1 can provide a particular interesting starting point for collaboration once datasets are established, since these playing styles are well described in the MT literature.
- Assess how much therapists (dis)agree in their individual intervention and analysis styles. Investigating the agreement between therapists requires the building of dedicated annotation tools and methods to measure agreement between annotators, such as in [79–82].
- Assess how the intuitive knowledge of the music therapist on their subjective experience in the improvisation can be combined with and enhanced by the objective analysis of the musical material through computation, to support therapists in their daily work, and to further establish evidence-based treatments in MT.
- Consider aspects of usability, including considerations from HCI, for developing appropriate tools for the daily practice of therapists; see [8, 42] for some general considerations on developing tools for MT.

5. CONCLUSION

Clinical improvisation from music therapy provide interesting and novel musical data for developing MIR methods for music structure analysis and pattern discovery, aiming to support therapists in their daily work. Interdisciplinary efforts between MIR and MT researchers need to be invested to close the gap between high-level concepts of music analysis used by therapists, and low-level features of current computational approaches to analyse MTI. Investigating music analysis methods employed by therapists, including the explication of therapists' implicit musical knowledge and the assessment of the agreement between different therapists, constitutes a crucial step for developing computational tools for MT. Specifically important herein is the establishing of data sets with clinical improvisations for different application areas, such as for initial clinical assessment, including different patient groups and catalogues of typical interventions of therapists. Creating these data sets through interdisciplinary efforts will not only prepare the ground for the appropriate computational modeling of music structures in MTI, but also to a better understanding of music analysis methods in MT, ultimately contributing to ongoing research on establishing evidence-based methods in MT.

6. REFERENCES

[1] K. S. Moore, "A systematic review on the neural effects of music on emotion regulation: Implications for music therapy practice," *Journal of Music Therapy*, vol. 50, no. 3, p. 198–242, 2013.

[2] S. Dalla Bella, C. E. Benoit, N. Farrugia, M. Schwartze, and S. A. Kotz, "Effects of musically cued gait training in Parkinson's disease: Beyond a motor benefit," *Annals of the New York Academy of Sciences*, vol. 1337, pp. 77–85, 2015.

[3] W. L. Magee and C. Bowen, "Using music in leisure to enhance social relationships with patients with complex disabilities," *NeuroRehabilitation*, vol. 23, pp. 305–311, 2008.

[4] P. Kantan, E. G. Spaich, and S. Dahl, "A Technical Framework for Musical Biofeedback in Stroke Rehabilitation," *IEEE Transactions on Human-Machine Systems*, vol. 52, no. 2, pp. 220–231, 2022.

[5] V. Bégel, A. Seilles, and S. Dalla Bella, "Rhythm workers: A music-based serious game for training rhythmic skills," *Music & Science*, vol. 1, pp. 1–16, 2018.

[6] K. Agres and D. Herremans, "Music and motion-detection: A game prototype for rehabilitation and strengthening in the elderly," in *International Conference on Orange Technologies*, 2017, pp. 95–98.

[7] Z. Vamvakousis and R. Ramirez, "The eyeharp: A gaze-controlled digital musical instrument," *Frontiers in Psychology*, vol. 7, 2016.

[8] K. R. Agres, R. S. Schaefer, A. Volk, S. van Hooren, A. Holzapfel, S. Dalla Bella, M. Müller, M. de Witte, D. Herremans, R. Ramirez Melendez, M. Neerincx, S. Ruiz, D. Meredith, T. Dimitriadis, and W. L. Magee, "Music, computing, and health: a roadmap for the current and future roles of music technology for health care and well-being," *Music & Science*, vol. 4, pp. 1–32, 2021.

[9] X. Serra, M. Magas, E. Benetos, M. Chudy, S. Dixon, A. Flexer, E. Gómez, F. Gouyon, P. Herrera, S. Jordà, O. Paytuvi, G. Peeters, J. Schlüter, H. Vinet, and G. Widmer, *Roadmap for music information research*. Creative Commons BY-NC-ND 3.0 License, 2013.

[10] N. Letulė, "Musical Profiling: Towards a Computer-Based Analysis of Clinical Improvisations," Master's thesis, Dept. of Music, Univ. of Jyväskylä, Jyväskylä, Finland, 2014.

[11] G. Luck, K. Riikkilä, O. Lartillot, J. Erkkilä, P. Toiviainen, A. Mäkelä, K. Pyhäluoto, H. Raine, L. Verkila, and J. Värri, "Exploring relationships between level of mental retardation and features of music therapy improvisations: A computational approach," *Nordic Journal of Music Therapy*, vol. 15, no. 1, pp. 30–48, 2006.

[12] E. Streeter, M. E. Davies, J. D. Reiss, A. Hunt, R. Caley, and C. Roberts, "Computer aided music therapy evaluation: Testing the Music Therapy Logbook prototype 1 system," *The Arts in Psychotherapy*, vol. 39, no. 1, pp. 1–10, Feb. 2012.

[13] C. Anagnostopoulou, A. Alexakis, and A. Triantafyllaki, "A Computational Method for the Analysis of Musical Improvisations by Young Children and Psychiatric Patients with No Musical Background," in *Proceedings of the 12th International Conference on Music Perception and Cognition and the 8th Triennial Conference of the European Society for the Cognitive Sciences of Music*, E. Cambouropoulos, C. Tsougras, P. Mavromatis, and K. Pastiadis, Eds., Thessaloniki, 2012.

[14] 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, Apr. 2017, Art. no. 537.

[15] J. Erkkilä, "Music Therapy Toolbox (MTTB) – An Improvisation Analysis Tool for Clinicians and Researchers," in *Microanalysis in Music Therapy: Methods, Techniques and Applications for Clinicians, Researchers, Educators and Students*. Jessica Kingsley Publishers, 2021, ch. 10, pp. 134–148.

[16] American Music Therapy Association. "Setting the record straight: What music therapy is and is not." Musictherapy.org. Accessed at Aug. 23, 2021. [Online]. Available: https://www.musictherapy.org/about/musictherapy/

[17] K. E. Bruscia, *Defining Music Therapy*, 3rd ed. Gilsum, NH, USA: Barcelona Publishers, 2014.

[18] H. Smeijsters, *Sounding the self: analogy in improvisational music therapy*. Gilsum, NH, USA: Barcelona Publishers, 2005.

[19] B. L. Wheeler, *Music Therapy Handbook*. New York, NY, USA: The Guilford Press, 2015.

[20] H. L. Bonny. "Music Psychotherapy: Guided Imagery and Music". Voices.no. Accessed at Okt. 22, 2021. [Online]. Available: https://voices.no/index.php/voices/article/view/1876/1640

[21] J. Fachner, "Music, moments and healing processes: Music therapy," in *The Routledge Companion to Music Cognition*, R. Ashley and R. Timmers, Eds. Abingdon, UK: Routledge, 2017, ch. 1.8, pp. 89–100.

[22] J. De Backer, B. Sebreghts, and K. Foubert, "Composition Plus. A process-compositional approach in music therapy to empower creative potential." *Journal of Urban Culture Research*, vol. 25, pp. 161–175, 2022.

[23] P. Nordoff and C. Robbins, *Creative Music Therapy*. New York: John Day, 1977.

[24] K. Aigen, *Being in music: Foundations of Nordoff-Robbins music therapy*. Gilsum, NH, USA: Barcelona Publishers, 1996.

[25] J. De Backer and J. Sutton, *The Music in Music Therapy: Psychodynamic Music Therapy in Europe : Clinical, Theoretical and Research Approaches*. Jessica Kingsley Publishers, 2014.

[26] K. E. Bruscia, *Improvisational models of music therapy*. Springfield, IL, USA: Charles C Thomas Publisher, 1987.

[27] J. Erkkilä and T. Wosch, "The Music Therapy Toolbox," in *Music Therapy Assessment: Theory, Research, and Application*. London, UK: Jessica Kingsley Publishers, 2019, ch. 15, pp. 293–314.

[28] Nordic Journal of Music Therapy. "IAP Revisited". Njmt.w.uib.no. Accessed at Apr. 12, 2022. [Online]. Available: https://njmt.w.uib.no/nordic-journal-of-music-therapy/forum-online-discussions-1998-2006/iap-revisited/

[29] A. Gilboa, "Nordic Journal of Music Therapy Developments in the MAP: A method for describing and analyzing music therapy sessions," *Nordic Journal of Music Therapy*, vol. 21, no. 1, pp. 57–79, 2012.

[30] T. Wigram and T. Wosch, *Microanalysis in Music Therapy: Methods, Techniques and Applications for Clinicians, Researchers, Educators and Students*. London, UK: Jessica Kingsley Publishers, 2007.

[31] H. T. Baxter, J. Berghofer, and L. MacEwan, *The individualized music therapy assessment profile: IMTAP*. London, UK: Jessica Kingsley Publishers, 2007.

[32] M. Forinash and D. Gonzalez, "A Phenomenological Perspective of Music Therapy," *Music Therapy*, vol. 8, no. 1, pp. 35–46, 1989.

[33] M. Langenberg, J. Frommer, and W. Tress, "A Qualitative Research Approach to Analytical Music Therapy," *Music Therapy*, vol. 12, no. 1, pp. 59–84, 1993.

[34] N. Letulė, E. Ala-Ruona, and J. Erkkilä, "Professional freedom: A grounded theory on the use of music analysis in psychodynamic music therapy," *Nordic Journal of Music Therapy*, vol. 27, no. 5, pp. 448–466, 2018.

[35] Washington State University. "Module 3: Clinical Assessment, Diagnosis, and Treatment". opentext.wsu.edu. Accessed at Dec. 18, 2021. [Online]. Available: https://opentext.wsu.edu/abnormal-psych/chapter/module-3-clinical-assessment-diagnosis-and-treatment/

[36] J. Fachner, "Communicating change-meaningful moments, situated cognition and music therapy: A response to North (2014)," *Psychology of Music*, vol. 42, no. 6, pp. 791–799, 2014.

[37] M. Gavrielidou and H. Odell-Miller, "An investigation of pivotal moments in music therapy in adult mental health," *The Arts in Psychotherapy*, vol. 52, pp. 50–62, Feb. 2017.

[38] G. Ansdell, J. Davidson, W. L. Magee, J. Meehan, and S. Procter, "From "This F***ing life" to "that's better" … in four minutes: an interdisciplinary study of music therapy's "present moments" and their potential for affect modulation," *Nordic Journal of Music Therapy*, vol. 19, no. 1, pp. 3–28, 2010.

[39] J. De Backer and T. Wigram, "Analysis of Notated Music Examples Selected from Improvisations of Psychotic Patients," in *Microanalysis in Music Therapy: Methods, Techniques and Applications for Clinicians, Researchers, Educators and Students*, London, UK, ch. 9, pp. 120–133.

[40] K. Foubert, B. Sebreghts, J. Sutton, and J. De Backer, "Musical encounters on the borderline. Patterns of mutuality in musical improvisations with Borderline Personality Disorder," *Arts in Psychotherapy*, vol. 67, 2020, Art. no. 101599.

[41] J. De Backer, "Music and psychosis: A research report detailing the transition from sensorial play to musical form by psychotic patients." *Nordic Journal of Music Therapy*, vol. 17, no. 2, pp. 89–104, 2008.

[42] T. Veldhuis, "Repeated musical patterns in music therapy analysis," Small Project Report, Utrecht University, 2022.

[43] J. Erkkilä, O. Lartillot, G. Luck, K. Riikkilä, and P. Toiviainen, "Intelligent music systems in music therapy," *Music Therapy Today*, vol. 5, no. 5, pp. 1–21, 2004.

[44] G. Husain, W. F. Thompson, and E. G. Schellenberg, "Effects of musical tempo and mode on arousal, mood, and spatial abilities," *Music perception*, vol. 20, no. 2, pp. 151–171, 2002.

[45] A. Kurkjian, K. Skinner, and H. Ahonen, "Using music-adapted technology to explore Bruscia's clinical techniques introduced in autism research: Pilot study," *Approaches: An Interdisciplinary Journal of Music Therapy*, vol. 13, no. 2, pp. 178–204, 2021.

[46] G. Luck, P. Toiviainen, J. Erkkilä, O. Lartillot, K. Riikkilä, A. Mäkelä, K. Pyhäluoto, H. Raine, L. Varkila, and J. Värri, "Modelling the relationships between emotional responses to, and musical content of, music therapy improvisations," *Psychology of Music*, vol. 36, no. 1, pp. 25–45, 2008.

[47] 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*, vol. 3, no. 1, 2020.

[48] F. Bimbot, O. Le Blouch, G. Sargent, and E. Vincent, "Decomposition into autonomous and comparable blocks: a structural description of music pieces," in *Proc. of the 11th International Society for Music Information Retrieval Conference*, Utrecht, The Netherlands, 2010, pp. 189–194.

[49] J. Paulus, M. Müller, and A. Klapuri, "Audiobased music structure analysis," in *Proc. of the 11th International Society for Music Information Retrieval Conference*, Utrecht, The Netherlands, 2010, pp. 625–636.

[50] J. Pauwels, F. Kaiser, and G. Peeters, "Combining Harmony-Based and Novelty-Based Approaches for Structural Segmentation," in *Proc. of the 14th International Society for Music Information Retrieval Conference*, Curitiba, Brazil, 2013, pp. 601–606.

[51] G. Shibata, R. Nishikimi, E. Nakamura, and K. Yoshii, "Statistical music structure analysis based on a homogeneity-, repetitiveness-, and regularityaware hierarchical hidden semi-Markov model," in *Proc. of the 20th International Society for Music Information Retrieval Conference*, Delft, The Netherlands, 2019, pp. 268–275.

[52] E. Cambouropoulos, "Musical parallelism and melodic segmentation:: A computational approach," *Music Perception*, vol. 23, no. 3, pp. 249–268, 2006.

[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 *Proc. of the 14th International Society for Music Information Retrieval Conference*, 2013, pp. 549–554.

[54] D. Conklin, "Discovery of distinctive patterns in music," *Intelligent Data Analysis*, vol. 14, no. 5, pp. 547–554, 2010.

[55] D. Conklin and C. Anagnostopoulou, "Comparative pattern analysis of Cretan folk songs," *Journal of New Music Research*, vol. 40, no. 2, pp. 119–125, 2011.

[56] R. B. Dannenberg and N. Hu, "Pattern discovery techniques for music audio," *Journal of New Music Research*, vol. 32, no. 2, pp. 153–163, 2003.

[57] J. Forth, "Cognitively-motivated geometric methods of pattern discovery and models of similarity in music," Ph.D. dissertation, Goldsmiths, Univ. of London, London, UK, 2012.

[58] B. Janssen, W. Haas, A. Volk, and P. v. Kranenburg, "Finding repeated patterns in music: State of knowledge, challenges, perspectives," in *International Symposium on Computer Music Multidisciplinary Research*. Springer, 2013, pp. 277–297.

[59] O. Lartillot, "Automated motivic analysis: An exhaustive approach based on closed and cyclic pattern mining in multidimensional parametric spaces," in *Computational Music Analysis*, D. Meredith, Ed. Springer, 2016, ch. 11, pp. 273–302.

[60] D. Meredith, K. Lemström, and G. A. Wiggins, "Algorithms for discovering repeated patterns in multidimensional representations of polyphonic music," *Journal of New Music Research*, vol. 31, no. 4, pp. 321–345, 2002.

[61] O. Nieto and M. M. Farbood, "Identifying polyphonic patterns from audio recordings using music segmentation techniques," in *Proc. of the 15th International Society for Music Information Retrieval Conference*, 2014, pp. 411–416.

[62] M. Pesek, A. Leonardis, and M. Marolt, "Sym-CHM—An Unsupervised Approach for Pattern Discovery in Symbolic Music with a Compositional Hierarchical Model," *Applied Sciences*, vol. 7, no. 11, p. 1135, 2017.

[63] I. Y. Ren, H. V. Koops, A. Volk, and W. Swierstra, "In search of the consensus among musical pattern discovery algorithms," in *Proc. of the 18th International Society for Music Information Retrieval Conference*, 2017, pp. 671–678.

[64] G. Velarde, D. Meredith, and T. Weyde, "A wavelet-based approach to pattern discovery in melodies," in *Computational Music Analysis*. Springer, 2016, ch. 12, pp. 303–333.

[65] M. Krause, M. Müller, and C. Weiß, "Towards Leitmotif Activity Detection in Opera Recordings," *Transactions of the International Society for Music Information Retrieval*, vol. 4, no. 1, p. 127–140, 2021.

[66] K. R. Page, T. Nurmikko-Fuller, C. Rindfleisch, Weigl, R. D. M., Lewis, L. Dreyfus, and D. De Roure, "A toolkit for live annotation of opera performance: Experiences capturing Wagner's Ring cycle," in *Proc. of the 16th International Society for Music Information Retrieval Conference*, 2015, pp. 211–217.

[67] L. Dreyfus and C. Rindfleisch, "Using digital libraries in the research of the reception and interpretation of Richard Wagner's leitmotifs," in *Proceedings of the International Workshop on Digital Libraries for Musicology*, 2014, pp. 1–3.

[68] M. Rohrmeier and M. Neuwirth, "Towards a Syntax of the Classical Cadence," in *What Is a Cadence? Theoretical and Analytical Perspectives on Cadences in the Classical Repertoire.* Leuven University Press, 2015, pp. 287–338.

[69] P. van Kranenburg and F. Karsdorp, "Cadence detection in western traditional stanzaic songs using melodic and textual features," in *Proc. of the 15th International*

Society for Music Information Retrieval Conference, 2014, pp. 391–396.

[70] L. Bigo, L. Feisthauer, M. Giraud, and F. Levé, "Relevance of musical features for cadence detection," in *Proc. of the 19th International Society for Music Information Retrieval Conference*, 2018, pp. 671–678.

[71] 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.

[72] P. Boot, A. Volk, and W. B. de Haas, "Evaluating the role of repeated patterns in folk song classification and compression," *Journal of New Music Research*, vol. 45, no. 3, pp. 223–238, 2016.

[73] R. Caro Repetto, R. Gong, N. Kroher, and X. Serra, "Comparision of the singing style of two Jingju schools," in *Proc. of the 16th International Society for Music Information Retrieval Conference*, Malaga, Spain, 2015, pp. 507–513.

[74] V. I. S. Gulati, J. Serrà and X. Serra., "Mining Melodic Patterns in Large Audio Collections of Indian Art Music," in *Proc. of the Tenth International Conference on Signal-Image Technology and Internet-Based Systems*, 2014, pp. 264–271.

[75] D. Herremans and E. Chew, "MorpheuS: generating structured music with constrained patterns and tension," *IEEE Transactions on Affective Computing*, vol. 10, no. 4, pp. 510–523, 2019.

[76] D. Conklin and G. Maessen, "Generation of melodies for the lost chant of the mozarabic rite," *Applied Sciences*, vol. 9, no. 20, 2019. [Online]. Available: https://www.mdpi.com/2076-3417/9/20/4285

[77] F. Pachet, "The Continuator: Musical Interaction With Style," *Journal of New Music Research*, vol. 32, no. 3, pp. 333–341, 2003.

[78] B. L. T. Sturm, A. L. Uitdenbogerd, H. V. Koops, and A. Huang, "Editorial for TISMIR special collection: AI and musical creativity," *Transactions of the International Society for Music Information Retrieval*, vol. 5, no. 1, pp. 67–70, 2022.

[79] D. Tomašević, S. Wells, I. Y. Ren, A. Volk, and M. Pesek, "Exploring annotations for musical pattern discovery gathered with digital annotation tools," *Journal of Mathematics and Music*, vol. 15, no. 2, pp. 194–207, 2021.

[80] 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.

[81] 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, no. 1, pp. 182–194, 2021.

[82] H. V. Koops, W. B. de Haas, J. A. Burgoyne, J. Bransen, A. Kent-Muller, and A. Volk, "Annotator subjectivity in harmony annotations of popular music," *Journal of New Music Research*, vol. 48, no. 3, pp. 232–252, 2019.

TIMBRE TRANSFER USING IMAGE-TO-IMAGE DENOISING DIFFUSION IMPLICIT MODELS

Luca Comanducci **Fabio Antonacci** **Augusto Sarti**
Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Italy
luca.comanducci@polimi.it, fabio.antonacci@polimi.it, augusto.sarti@polimi.it

ABSTRACT

Timbre transfer techniques aim at converting the sound of a musical piece generated by one instrument into the same one as if it was played by another instrument, while maintaining as much as possible the content in terms of musical characteristics such as melody and dynamics. Following their recent breakthroughs in deep learning-based generation, we apply Denoising Diffusion Models (DDMs) to perform timbre transfer. Specifically, we apply the recently proposed Denoising Diffusion Implicit Models (DDIMs) that enable to accelerate the sampling procedure. Inspired by the recent application of DDMs to image translation problems we formulate the timbre transfer task similarly, by first converting the audio tracks into log mel spectrograms and by conditioning the generation of the desired timbre spectrogram through the input timbre spectrogram. We perform both one-to-one and many-to-many timbre transfer, by converting audio waveforms containing only single instruments and multiple instruments, respectively. We compare the proposed technique with existing state-of-the-art methods both through listening tests and objective measures in order to demonstrate the effectiveness of the proposed model.

1. INTRODUCTION

Timbre is an extremely important perceptual aspect of music, yet it is hard to both model and define. The concept of musical timbre can be defined as the perceived characteristics of a musical sound that are different from pitch and amplitude contours [1].

Timbre Transfer concerns the task of converting a musical piece from one timbre to another while preserving the other music-related characteristics. While this operation is not trivial, it is of extreme interest for several applications, from the development of plugins to be used in Digital Audio Workstations (DAW) to enabling the possibility of playing sounds corresponding to not widely available musical instruments.

© L. Comanducci, F. Antonacci, and A. Sarti. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** L. Comanducci, F. Antonacci, and A. Sarti, "Timbre Transfer using Image-to-Image Denoising Diffusion Implicit Models", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.*, Milan, Italy, 2023.

In this paper, we present DiffTransfer, a technique for timbre transfer which is tested both between single and multiple instruments and is based on a continuous Denoising Diffusion Implicit Model (DDIM) with deterministic sampling [2], a modified version of Denoising Diffusion Probabilistic Models (DDPMs) that are trained using the same procedure, but allow for faster sampling times. Specifically, in [2] it was empirically shown that DDIMs allow for $10\times - 50\times$ faster wall-clock time performances with respect to DDPMs.

In order to be able to convert one timbre into another, we use a procedure similar to the recently proposed image-to-image technique Palette [3]. Specifically, we use as input to the diffusion model the noise and condition it with the chosen input timbre spectrogram, then, through the denoising procedure, the model learns to reconstruct spectrograms of the desired timbre. We consider the scenario where the timbre-transfer task is *paired*, which means that the desired and input spectrograms have the same melodic/harmonic content, but differ in terms of timbre.

We experiment both with the possibility of converting between tracks containing only single instruments and also mixtures of instruments, with no prior separation step, while making no modifications to the model in order to take into account both configurations.

In order to demonstrate the effectiveness of the proposed model, we compare DiffTransfer with state-of-the-art techniques, both through objective measures and by performing a user-based listening test. The source code and audio excerpts can be found at `https://lucacoma.github.io/DiffTransfer/`.

2. RELATED WORK

Several types of timbre Transfer techniques have been proposed in the literature. In [4] a CycleGAN [5] is applied in order to perform an unpaired transfer using the Constant-Q transform and the audio is then recovered through a WaveNet [6] model. In [7] an attention-based architecture is applied in order to convert mel spectrograms, which are then inverted through a MelGAN architecture [8]. Gaussian mixture-based variational autoencoders are applied [9] in order to learn a latent space where pitch and timbre representations are disentangled.

Another class of methods, instead, extracts musical parameters such as pitch and loudness from the input audio tracks and performs the transfer by resynthesizing sound

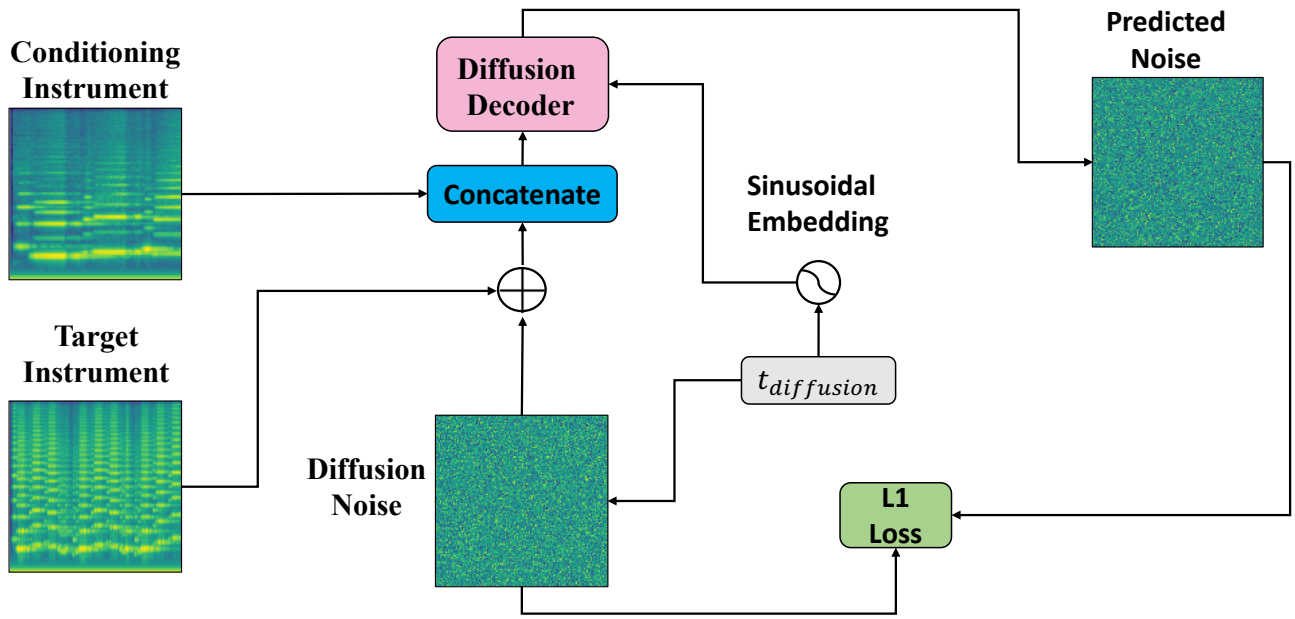


Figure 1: Training scheme of the proposed DiffTransfer technique. The target instrument spectrogram is summed with noise following a simplified cosine schedule. The decoder, conditioned on the conditioning instrument spectrogram and on the sinusoidal embedding representing the current time instant estimates the added noise. The decoder parameters are estimated by computing the L1 loss between the ground truth and the estimated diffusion noise.

through a network that has learned to generate tracks with the desired timbre. The most known example of these techniques is the Differentiable Digital Signal Processing (DDSP) [10] model. Other similar techniques were proposed such as [11], where a hierarchical model is used in order to reconstruct the signal at increasing resolutions. Recently there have been proposed also models that directly work on the audio waveform such as [12], where music pieces are translated to specific timbre domains. The only model that, to the best of our knowledge and except for the one proposed in this paper, is tested on multi-instrument timbre transfer without any source separation pre-processing is the Music-STAR network, presented in [13]. In Music-STAR a WaveNet autoencoder [14] is trained by applying teacher-forcing [15] to the decoders in order to recover the desired timbre.

Denoising Diffusion Probabilistic Models (DDPMs) [16] have recently become the latest state-of-the-art for what concerns deep learning-based generation fastly replacing Generative Adversarial Networks (GANs) [17] and Variational Autoencoders [18], due to their easier training procedure and increased quality of the produced results.

DDPMs have been successfully applied to a wide variety of image-related tasks such as generation [19] and translation [3].

More recently, DDPMs have been also used for audio-related tasks. In [20] a diffusion model is applied in order to convert midi tracks to spectrograms, while in [21] a text-to-music diffusion model is proposed. DDPMs have also been applied to symbolic music generation [22], speech synthesis [23] and singing voice extraction [24].

While DDPMs have extremely powerful generation capabilities they suffer from slow sampling times. To ameliorate this issue, recently Denoising Diffusion Implicit Models (DDIMs) [2], which allow for faster sampling times and were recently applied to image inpainting [25].

3. PROPOSED MODEL

In this section, we describe the proposed DiffTransfer technique for timbre transfer. Instead of working directly with raw audio signals, we convert them into log mel-scaled spectrograms, due to their easier handling by deep learning models. We then propose a model that, given as input the spectrogram corresponding to the conditioning instrument, generates the corresponding target spectrogram that would have been obtained by playing the same piece of music with the target instrument. Operatively we achieve this through a conditional continuous-time DDIM, which learns to denoise the target instrument spectrogram, while conditioned on the input instrument spectrogram, as depicted in Fig. 1. At inference time, the model is fed with the input conditioning instrument concatenated with Gaussian noise and generates the corresponding target spectrogram. We retrieve the audio signal by applying to the log mel spectrograms the SoundStream [1] model [26], provided by [20] where it was trained on a custom music dataset.

In the following, we'll provide a brief overview of the DDIM framework and notation used in this paper, in order to make the tractation as compact as possible, for additional and more thorough formulations, we refer the reader to [2] and [3]. We aim at giving a general overview of the process and we'll use a slight abuse of notation to describe the diffusion process using the continuous time framework,

[1] https://tfhub.dev/google/soundstream/mel/decoder/music/1

in order to make it more similar to the more common literature regarding DDPMs and DDIMs.

3.1 Diffusion Decoder

We adopt a procedure similar to the Palette [3] image-to-image translation technique in order to train the timbre transfer decoder as a Denoising Diffusion Implicit Model (DDIM) [2]. Broadly speaking, DDIMs work by learning how to generate data from noise in a two-part procedure. The first part is denoted as the *forward process*, where Gaussian noise $\gamma \sim \mathcal{N}(0,1)$ is subsequently added to the input until it is indistinguishable from the former. The second part consists of the *reverse process* where a decoder learns how to invert the forward process, effectively reconstructing data from the noise. DDIMs can be seen as a generalization of DDPMs that shares the same training procedure, however, they differ in the modeling of the reverse process, by using a non-markovian diffusion process, which allows for faster generation times.

3.1.1 Forward Process

Let us define $\mathbf{X}$ and $\mathbf{Y}$ as the log mel spectrograms corresponding to the conditioning and target instruments, respectively. We choose a continuous diffusion time [27–29]in order to be able to change the number of desired sampling steps. If we consider T steps, then the diffusion time can be defined as $t \in \{0,1\}$, where consecutive times are separated by $\Delta_t = 1/T$. Then, the forward process is defined similarly to the case of DDPMs by subsequently adding noise to the target spectrogram for T steps

$$\begin{aligned} q(\mathbf{Y}_t|\mathbf{Y}_{t-\Delta_t}) &= \mathcal{N}(\mathbf{Y}_t, \sqrt{(\alpha_t)}\mathbf{Y}_{t-\Delta_t}, \beta_t\mathbf{I}), \\ q(\mathbf{Y}_{1:T}|\mathbf{Y}_0) &= \prod_{t=1}^{T} q(\mathbf{Y}_{t-\Delta_t}) \end{aligned} \tag{1}$$

where α and β are parameters defined by a simplified cosine schedule [30].

3.1.2 Reverse Process

In the case of DDIMs, the reverse diffusion process is operated by introducing an additional distribution p_θ, where a sample $\mathbf{Y}_{t-\Delta t}$ can be generated from a sample $\mathbf{Y}_t$ as

$$\begin{aligned} \mathbf{Y}_{t-\Delta t} =& \sqrt{\beta_{t-\Delta t}} \left(\frac{c - \sqrt{\beta_t}\gamma_\theta^{(t)}(\mathbf{Y}_t, \mathbf{X})}{\sqrt{(\alpha_t)}} \right) + \\ & \sqrt{1-\alpha_{t-\Delta_t}} \cdot \gamma_\theta^{(t)}(\mathbf{Y}_t, \mathbf{X}), \end{aligned} \tag{2}$$

, where γ is the noise estimated by a network with parameters θ. The noise at time t $\gamma_\theta^{(t)}$ is estimated by a network that is conditioned also on the input timbre spectrogram $\mathbf{X}$, similarly to the formulation proposed in Palette [3].

3.1.3 Training Procedure

The denoising process is operated through a U-Net architecture which is conditioned on $\mathbf{X}$ and trained to predict the added noise in order to minimize the L1 loss

$$\mathbb{E} = ||\gamma_\theta^{(t)}(\mathbf{Y}_t, \mathbf{X}) - \gamma||_1^1, \tag{3}$$

where γ is the true perturbation, while $\gamma_\theta^{(t)}(\mathbf{Y}_t, \mathbf{X})$ is the estimate of the noise added to the target spectrogram at time t, conditioned on the input spectrogram $\mathbf{X}$.

3.2 Architecture

The decoder architecture is based on a U-Net model. The building element is made of residual blocks, in each of these the input is processed by (i) a 2D convolutional layer with swish activation, followed by batch normalization and by (ii) a convolutional layer with no activation. Both convolutional layers have kernel size 3. The output of this procedure is then summed with the residual, which is obtained by processing the input with a convolutional layer with kernel size 1.

The encoder part of the network consists of 3 downsampling blocks, each consisting of 4 residual blocks having filter sizes $64, 128, 256$. The output of each downsampling block is followed by average pooling, with pool size 2 in order to compress the dimension of the spectrograms. The last block of the encoder is followed a self-attention block.

The bottleneck obtained through the encoder is processed by a residual block with 512 filters and is then processed by the decoder, which is a specular version of the encoder. The only difference lies in the use of transposed convolutions in order to create upsampling layers needed to increase the dimension of the features.

The last downsampling layer of the encoder, the bottleneck and the first upsampling layer of the decoder are followed by self-attention.

3.3 Deployment

The proposed model takes as input spectrograms of a fixed size, therefore audio tracks longer than the ones used for training need to be sliced accordingly.

The decoder takes as input the conditioning spectrogram $\mathbf{X}$ and the diffusion noise and retrieves an estimate of the latter, which can then be subtracted in order to obtain an estimate of the desired output timbre spectrogram $\hat{\mathbf{Y}}$. The output waveform y can then be obtained by feeding the pre-trained SoundStream model with $\hat{\mathbf{Y}}$.

4. EXPERIMENTS

In this section, we describe experiments performed with the aim of demonstrating the capabilities of the proposed DiffTransfer technique both in the single-instrument and multi-instrument application scenarios.

In Fig. 3 we show an example of input, generated and ground-truth spectrograms, obtained via the DiffTransfer model when converting from a Clarinet to Strings.

4.1 Dataset

In order to train the model we considered the StarNet dataset [31], which contains a set of tracks that are played with two timbre-domains, namely strings-piano and vibraphone-clarinet. The dataset consists of roughly 22 hours of audio. We used the reduced version of the dataset,

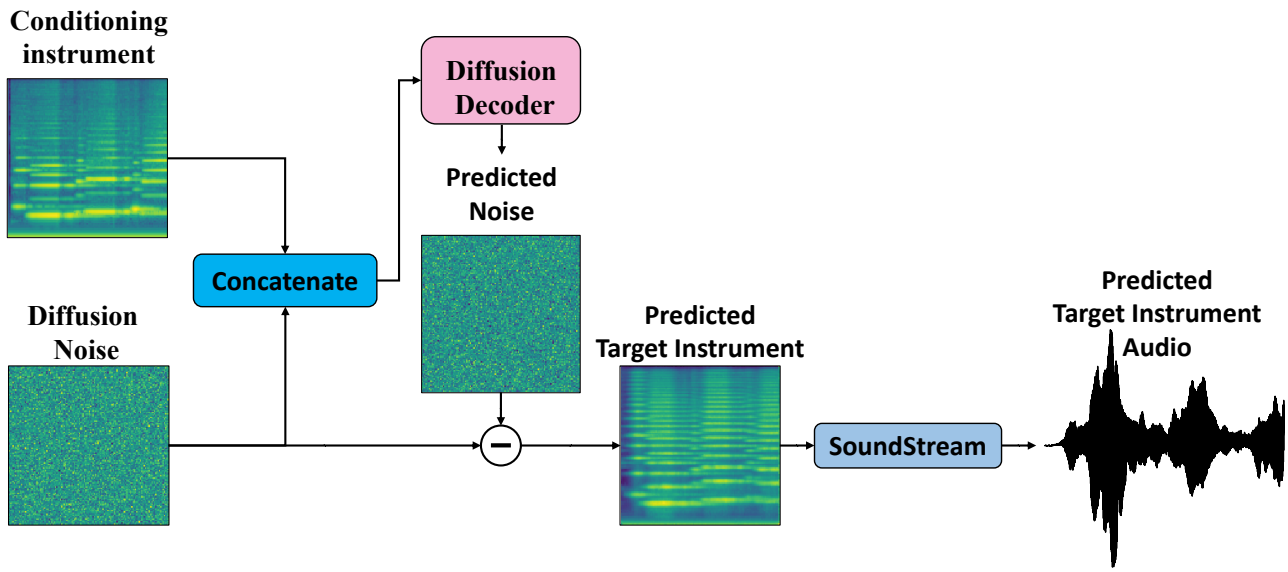


Figure 2: Deployment scheme of the proposed DiffTransfer technique. The decoder is fed with Gaussian noise and with the conditioning instrument spectrogram. The noise estimate provided by the decoder is then subtracted from the input noise in order to provide an estimate of the desired target spectrogram, from which the audio is estimated via the SoundStream model [20, 26].

where tracks are resampled to 16000 Hz and converted them to mono. In order to perform the evaluation, we use the same ten tracks considered in [13], in order to ease the comparison with their model.

4.2 Techniques Under Comparison

We consider two baselines in order to compare the performances of the proposed DiffTransfer architecture. For what concerns the single-instrument timbre transfer task, we consider the Universal Network [12] fine-tuned on the StarNet dataset as done in [13]. For what concerns the multi-timbre task, we consider the mixture-supervised version of the Music-STAR network proposed in [13]. We perform three different types of timbre transfer tasks: *single*, where only single instruments are converted, *single/mixed* where the separate conversions of single instruments are mixed in order to create the desired mixture track and *mixture*, where the mixture is directly converted. These nomenclatures are used just to ease the presentation of the results, we would like to point out that, for what concerns the DiffTransfer architecture, no specific changes are required for the various types of applications, except for the choice of desired input data.

4.3 Experiment Setup

The Universal Network and Music-STAR architectures are trained with the procedure described in [13]. The DiffTransfer network is trained for 5000 epochs using a batch size of 16, with the AdamW optimizer [32] with learning rate $2e-5$ and weight decay $1e-4$. The epoch that minimizes the $L1$ noise prediction loss is chosen in order to retain the model used to compute the results. We train a total of six models, performing the following timbre transfer conversions: vibraphone to piano, piano to vibraphone, clarinet to strings, strings to clarinet vibraphone/clarinet to piano/strings and piano/strings to vibraphone/clarinet.

The network input features are computed by first applying the Short-Time Fourier Transform (STFT) with a Hann window of size 0.020 s and 50% overlap to normalized audio tracks. Then the log mel spectrogram is computed over 128 bins corresponding to the range of $0-16000$ Hz. We do not feed the entire audio tracks as input to the network, instead, during each epoch we extract 128 frames from the log mel spectrogram, corresponding to $\approx$ 2 s. Each spectrogram slice is normalized between -1 and 1 before being given as input to the network and the output spectrograms are denormalized before being fed to the SoundStream model in order to recover the audio waveform. Since the tracks considered for the test are of length 10 s and the model gets as input a fixed 128 frames spectrogram we slice the conditioning spectrogram before feeding into the model and we keep the input noise fixed for all slices, in order to ensure consistency in the generation. All spectrogram slices are normalized in the range $[-1, 1]$ and denormalized before being fed to the SoundStream decoder.

4.4 Objective Evaluation

We evaluate the model objectively in order to analyze the perceptual similarity and content preservation capabilities of the generated tracks with respect to the ground truth audio.

In order to evaluate the perceptual similarity, we compute the Fréchet Audio Distance (FAD) [33] using the VGGish embeddings [34], through a PyTorch implementation [2]. FAD is a reference-free metric for music enhance-

[2] https://pypi.org/project/frechet-audio-distance/

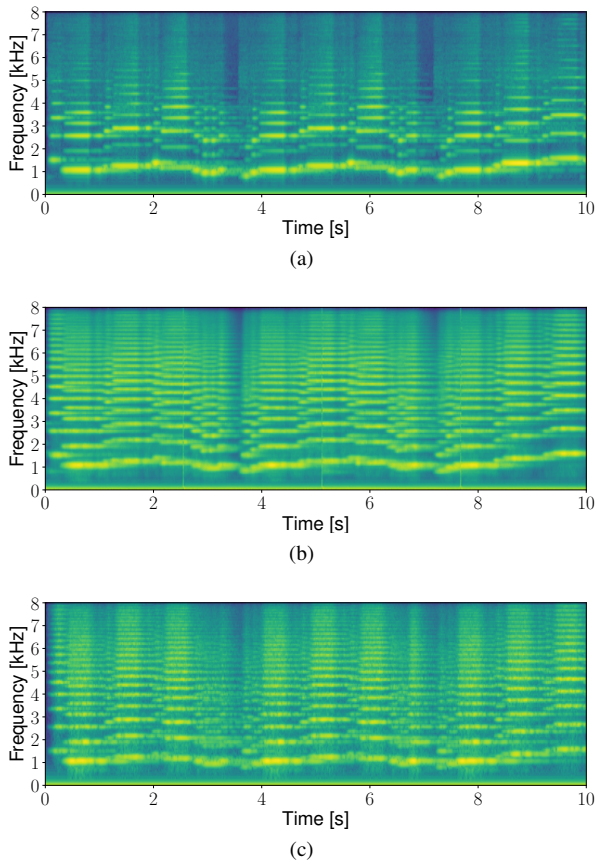


Figure 3: Example of Timbre Conversion log mel Spectrograms using the DiffTransfer architecture, obtained when converting Clarinet (a) to Strings (b). The ground truth Strings spectrogram is shown in (c).

ment algorithms, which views the embeddings as a continous multivariate Gaussian and is computed between the real and generated data as

$$\text{FAD} = ||\mu_r - \mu_g||^2 + \text{tr}(\Sigma_r + \mu_g - 2\sqrt{\Sigma_r \Sigma_g}), \quad (4)$$

where (μ_r, Σ_r) and (μ_g, Σ_g) are the mean and covariances of the embeddings corresponding to the real and generated data, respectively. Similarly to [20], we compute FAD in order to analyze the perceptual similarity between the generated audios with respect to the ground truth one, corresponding to the original StarNet dataset.

To understand the content-preservation capabilities of the model, following [35], we compute how the pitch contours of generated ground truth audio tracks are dissimilar, by calculating the mismatch between two sets of pitches A and B through the Jaccard Distance

$$JD(A, B) = 1 - \frac{|A \cap B|}{|A \cup B|}, \quad (5)$$

where a lower value corresponds to a lower mismatch and thus to a higher degree of similarity between the generated pitch contours. Pitch contours are computed using a multi-pitch version of the MELODIA [36] as implemented in the Essentia library [37], rounding pitches to the nearest semitone. We report the values obtained by computing the metrics on the test dataset in Table 1.

Objective Evaluation		
Method	**FAD ↓**	**JD ↓**
Universal Network (single)	7.09	0.53
DiffTransfer (single)	2.58	0.28
Universal Network (single/mixed)	10.47	0.64
DiffTransfer (single/mixed)	4.73	0.46
Music-STAR (mixture)	8.93	0.57
DiffTransfer (mixture)	4.37	0.38

Table 1: Objective Evaluation of the proposed DiffTransfer Method compared to the baselines, in terms of Fréchet Audio Distance (FAD) and Jaccard Distance (JD). Results are averaged over all participants and over all the tracks considered for each part of the test.

4.5 Subjective Evaluation

In order to evaluate subjectively the timbre transfer capabilities, we perform a listening test with 18 human participants. The web page of the test is available at [3]. The test was split into two parts corresponding to the single and multiple instrument application scenarios, respectively.

During the single instrument part of the test, the users listened to four tracks, corresponding to the four types of conversions performed, namely: clarinet to strings, strings to clarinet, piano to vibraphone, vibraphone to piano. Each example consisted of two conditions, one obtained via the DiffTransfer model and the other through the Universal Network.

In the second part of the test, concerning multiple instrument timbre transfer, a total of four tracks were considered, two for the conversion from vibraphone/strings to piano/strings waveforms and two for the reverse conversion. Each example consisted of four conditions, namely DiffStar (single/mix), Universal Network (single/mix), DiffStar (mixture) and Music-STAR (mixture).

Both the order of conditions and the order of examples in each separate part of the test were randomized.

The participants were asked to rate the conditions in terms of similarity with respect to the reference track on a 5 elements Likert scale where 1 corresponds to bad and 5 to excellent. We report the results obtained through the listening test in Table 2.

4.6 Discussion

By briefly inspecting both the objective and subjective results, reported in Table 1 and 2, respectively, it is clear how the proposed DiffTransfer model outperforms the Universal Network and Music-STAR baselines both for what concerns the single and multiple timbre transfer tasks.

When considering single timbre results, DiffTransfer is able to achieve significantly better performances in terms of FAD, Jaccard Distance and Perceived Similarity, with respect to the Universal network. The gap between the two methods becomes even more evident when considering the

[3] https://listening-test-ismir-ttd.000webhostapp.com/

Subjective Evaluation	
Method	**Similarity**
Universal Network (single)	1.82
DiffTransfer (single)	3.68
Universal Network (single/mixed)	1.69
DiffTransfer (single/mixed)	3.78
Music-STAR (mixture)	2.89
DiffTransfer (mixture)	3.80

Table 2: Objective Evaluation of the proposed DiffTransfer Method compared to the baselines, in terms of perceived similarity with respect to the ground truth on a Likert scale from 1 (Bad) to 5 (Excellent). Results are averaged over all test tracks.

single/mixed case, i.e. when single timbre transfer tracks are mixed in order to form the desired mixture audio.

For what concerns the Music-STAR method, the gap with respect to DiffTransfer remains high in terms of FAD, but becomes less noticeable when considering JD and the perceived subjective similarity.

5. CONCLUSION

In this paper, we have presented DiffTransfer a technique for both single- and multi-instrument timbre transfer using Denoising Diffusion Implicit models. The novelty of the proposed approach lies in the fact that in addition to being, to the best of our knowledge, the first application of diffusion models to timbre transfer, it is the first model to be tested in order to perform single and multi-timbre transfer, without varying the architecture depending on which application is chosen. We compared the proposed model with state-of-the-art Universal Network and Music-STAR baselines through both objective evaluation measures and a listening test, demonstrating the better capabilities of the proposed DiffTransfer approach.

Future works will involve increasing the audio quality of the generated audio, by taking into account the consistency of subsequent generated spectrograms. Furthermore, we plan on modifying the model in order to be able to perform unpaired timbre transfer, which greatly eases the dataset requirements and applicability of the technique.

6. REFERENCES

[1] J. T. Colonel and S. Keene, "Conditioning autoencoder latent spaces for real-time timbre interpolation and synthesis," in *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2020, pp. 1–7.

[2] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," in *International Conference on Learning Representations*, 2021.

[3] C. Saharia, W. Chan, H. Chang, C. Lee, J. Ho, T. Salimans, D. Fleet, and M. Norouzi, "Palette: Image-to-image diffusion models," in *ACM SIGGRAPH 2022 Conference Proceedings*, 2022, pp. 1–10.

[4] S. Huang, Q. Li, C. Anil, X. Bao, S. Oore, and R. B. Grosse, "Timbretron: A wavenet(cycleGAN(CQT(audio))) pipeline for musical timbre transfer," in *International Conference on Learning Representations*, 2019.

[5] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. of the IEEE international conference on computer vision*, 2017, pp. 2223–2232.

[6] 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.

[7] D. K. Jain, A. Kumar, L. Cai, S. Singhal, and V. Kumar, "Att: Attention-based timbre transfer," in *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2020, pp. 1–6.

[8] K. Kumar, R. Kumar, T. De Boissiere, L. Gestin, W. Z. Teoh, J. Sotelo, A. de Brébisson, Y. Bengio, and A. C. Courville, "Melgan: Generative adversarial networks for conditional waveform synthesis," *Advances in neural information processing systems*, vol. 32, 2019.

[9] Y.-J. Luo, K. Agres, and D. Herremans, "Learning disentangled representations of timbre and pitch for musical instrument sounds using gaussian mixture variational autoencoders," in *20th International Society for Music Information Retrieval (ISMIR2019)*, 2019.

[10] J. Engel, L. H. Hantrakul, C. Gu, and A. Roberts, "Ddsp: Differentiable digital signal processing," in *International Conference on Learning Representations*, 2020.

[11] M. Michelashvili and L. Wolf, "Hierarchical timbre-painting and articulation generation," in *21st International Society for Music Information Retrieval (ISMIR2020)*, 2020.

[12] A. P. Noam Mor, Lior Wold and Y. Taigman, "A universal music translation network," in *International Conference on Learning Representations (ICLR)*, 2019.

[13] M. Alinoori and V. Tzerpos, "Music-star: a style translation system for audio-based re-instrumentation," in *21st International Society for Music Information Retrieval (ISMIR2022)*, 2022.

[14] 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.

[15] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural computation*, vol. 1, no. 2, pp. 270–280, 1989.

[16] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in Neural Information Processing Systems*, vol. 33, pp. 6840–6851, 2020.

[17] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[18] A. Roberts, J. Engel, and D. Eck, "Hierarchical variational autoencoders for music," in *NIPS Workshop on Machine Learning for Creativity and Design*, vol. 3, 2017.

[19] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. Denton, S. K. S. Ghasemipour, R. Gontijo-Lopes, B. K. Ayan, T. Salimans, J. Ho, D. J. Fleet, and M. Norouzi, "Photorealistic text-to-image diffusion models with deep language understanding," in *Advances in Neural Information Processing Systems*, A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, Eds., 2022.

[20] C. Hawthorne, I. Simon, A. Roberts, N. Zeghidour, J. Gardner, E. Manilow, and J. Engel, "Multi-instrument music synthesis with spectrogram diffusion," in *Ismir 2022 Hybrid Conference*, 2022.

[21] D. Yang, J. Yu, H. Wang, W. Wang, C. Weng, Y. Zou, and D. Yu, "Diffsound: Discrete diffusion model for text-to-sound generation," 2022.

[22] G. Mittal, J. Engel, C. Hawthorne, and I. Simon, "Symbolic music generation with diffusion models," in *Proc. of the 22nd International Society for Music Information Retrieval Conference*, 2021.

[23] Z. Kong, W. Ping, J. Huang, K. Zhao, and B. Catanzaro, "Diffwave: A versatile diffusion model for audio synthesis," in *International Conference on Learning Representations*, 2021.

[24] 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 (ISMIR) Conference*, 2022.

[25] G. Zhang, J. Ji, Y. Zhang, M. Yu, T. Jaakkola, and S. Chang, "Towards coherent image inpainting using denoising diffusion implicit models," *arXiv preprint arXiv:2304.03322*, 2023.

[26] 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.

[27] A. Campbell, J. Benton, V. De Bortoli, T. Rainforth, G. Deligiannidis, and A. Doucet, "A continuous time framework for discrete denoising models," *Advances in Neural Information Processing Systems*, vol. 35, pp. 28 266–28 279, 2022.

[28] 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.

[29] S. Rouard and G. Hadjeres, "Crash: raw audio score-based generative modeling for controllable high-resolution drum sound synthesis," in *22nd International Society for Music Information Retrieval (ISMIR2021)*, 2021.

[30] A. Q. Nichol and P. Dhariwal, "Improved denoising diffusion probabilistic models," in *International Conference on Machine Learning*. PMLR, 2021, pp. 8162–8171.

[31] M. Alinoori and V. Tzerpos, "Starnet," Aug. 2022. [Online]. Available: https://doi.org/10.5281/zenodo.6917099

[32] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations*, 2019.

[33] K. Kilgour, M. Zuluaga, D. Roblek, and M. Sharifi, "Fréchet audio distance: A reference-free metric for evaluating music enhancement algorithms." in *INTERSPEECH*, 2019, pp. 2350–2354.

[34] 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.

[35] 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.

[36] 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.

[37] D. Bogdanov, N. Wack, E. Gómez Gutiérrez, S. Gulati, H. Boyer, O. Mayor, G. Roma Trepat, 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.

CORRELATION OF EEG RESPONSES REFLECTS STRUCTURAL SIMILARITY OF CHORUSES IN POPULAR MUSIC

Neha Rajagopalan
Stanford University
neharaj@stanford.edu

Blair Kaneshiro
Stanford University
blairbo@stanford.edu

ABSTRACT

Music structure analysis is a core topic in Music Information Retrieval and could be advanced through the inclusion of new data modalities. In this study we consider neural correlates of music structure processing using popular music—specifically choruses of Bollywood songs—and the NMED-H electroencephalographic (EEG) dataset. Motivated by recent findings that listeners' EEG responses correlate when hearing a shared music stimulus, we investigate whether responses correlate not only within single choruses but across pairs of chorus instances as well. We find statistically significant correlations within and across several chorus instances, suggesting that brain responses synchronize across structurally matched music segments even if they are not contextually or acoustically identical. Correlations were only occasionally higher within than across choruses. Our findings advance the state of the art of naturalistic music neuroscience, while also highlighting a novel approach for further studies of music structure analysis and audio understanding more broadly.

1. INTRODUCTION

Music structure analysis (MSA)—the task of dividing and labelling songs into perceptually salient segments [1]—is a core topic of Music Information Retrieval (MIR) and has been approached through a variety of data types including audio representations, lyrics, and perceptual annotations. For example, choruses of popular songs are often easily recognizable by music listeners, and can be detected from audio due to both their placement throughout a song and their intrinsic features [2]. While much progress has been made in this area, there may be new approaches and data modalities that could advance it even further.

MIR studies have come to involve brain data, particularly electroencephalography (EEG) [3]. EEG has been used to predict stimulus labels, decode musical attributes such as beat and tempo, and even reconstruct music. EEG inter-subject correlation (ISC), which captures neural synchronization of audience members experiencing a complex, real-world stimulus [4], has also advanced music neuroscience research. We leverage and extend this approach to investigate MSA.

We focus on responses to four Bollywood songs written in the popular form—specifically their choruses, due to their salience and tendency to repeat with a high degree of similarity. Importantly, while past EEG-ISC studies have considered responses among listeners experiencing the same stimulus (e.g., one chorus instance), we ask for the first time whether neural responses also synchronize *across* instances of structurally similar content (i.e., pairs of choruses). Moreover, by using a dataset containing two response trials from each participant, we can investigate correlations both across and within participants. In sum, we address the following research questions:

RQ 1 *Does music structure similarity translate to measurable similarity among responses?* In other words, do brain responses synchronize across structurally matched musical segments, even when those segments are contextually unique (in their placement within the song) and also often acoustically unique from one another? Here we expect structural similarity to produce statistically significant EEG correlations both within and across a song's choruses.

RQ 2 *Even if responses are similar across chorus instances, are individual choruses still uniquely experienced?* This question extends RQ1 to investigate whether EEG responses are more correlated within, versus across, chorus instances. We predict that within-chorus EEG correlations will be higher than across-chorus correlations.

RQ 3 *Are a listener's neural responses more similar to themselves than to responses from other listeners?* Understanding whether reliable measures of music structure similarity can be obtained from single listeners can motivate the design of future studies. We expect EEG correlation with one's own data will be higher than correlation with the data of other listeners, due to individual differences in perception and EEG characteristics.

We report small but often significant correlations that align with previous published research. Moreover, within-chorus correlations do not systematically outperform across-chorus correlations. While preliminary, our findings suggest that this novel application of EEG correlation may capture structural similarity during music listening, which may motivate future MSA studies.

© N. Rajagopalan and B. Kaneshiro. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** N. Rajagopalan and B. Kaneshiro, "Correlation of EEG Responses Reflects Structural Similarity of Choruses in Popular Music", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.*, Milan, Italy, 2023.

2. RELATED WORK

2.1 MSA and Chorus Analysis

MSA involves recognizing and labelling non-overlapping musical segments based on musical similarity [1]. Over the years, MSA has come to involve specific features, similarity representations, and algorithms [5]. One sub-topic of MSA is chorus identification; here, choruses have often been identified based on repetition and contextual cues using measures of similarity [6] and Markov models [7], as well as chroma features and image processing filters [8]. Some systems have also used segment length and positioning to identify choruses [6, 8]. Independently of context, Van Balen et al. looked at intrinsic content features that might distinguish choruses [2]. Their "Chorusness" variable, a probability measure of how likely a segment may be labelled as a chorus by an independent annotator, highlights audio features (e.g., higher loudness and roughness) that qualify the particular salience of choruses.

MSA remains a challenging task due, for example, to ambiguities around defining similarity as well as subjectivity and interpretation of annotations [1, 9]. In their 2020 overview article, Nieto et al. called for "richer human labels in upcoming MSA datasets" [1]; we propose that brain data may fit this call.

2.2 MIR and EEG

The growing use of decoding and signal-based approaches and complex, naturalistic (real-world) stimuli in neuroscience has increased that field's relevance to the more applied field of MIR. Kaneshiro & Dmochowski have suggested that MIR and neuroscience researchers might augment their gains through collaboration, highlighting EEG as a particularly relevant response type for MIR due to its high temporal resolution, non-invasiveness, whole-brain coverage, and relative portability and low cost [3].

EEG studies addressing MIR topics include using classification to predict which stimulus elicited an EEG response [10, 11] or which stream a listener attended to in a polyphonic stimulus [12–15]. Other tasks include EEG-based tempo detection/classification [16–18], onset detection [19], and music reconstruction [20]. EEG has been mapped to time-varying music or audio features using Canonical Correlation Analysis (CCA) [21] or deep-CCA [22]; by correlating EEG with semantic music vectors [23]; or using MEG—the magnetic analogue of EEG—and temporal response functions to decode surprisal [24]. In recent studies, Ofner and Stober examined EEG responses at automated segmentation boundaries [25], and Sangnark et al. performed music preference classification on EEG responses to choruses with and without lyrics [26]. However, we know of no study to date that has assessed *similarity* among EEG responses to repeated structural segments.

2.3 Neural Correlation

A particularly relevant approach for the current study involves the correlation of neural responses to a shared stimulus, often termed inter-subject correlation (ISC). Hasson et al.'s 2004 seminal functional magnetic resonance imaging study showed that real-world stimuli (e.g., movie excerpts) can synchronize neural responses across audience members, and that the timing and location of synchronized activity identifies stimulus-evoked brain activity [27]. This data-driven approach, reducing the need for controlled stimuli and a priori event markers, facilitated the use of complex stimuli in neuroscience. In 2012, Dmochowski et al. introduced an EEG implementation which first optimizes the data for ISC [4]. Often referred to as "Correlated Components Analysis (CorrCA)" [4] or "Reliable Components Analysis (RCA)" [28], this optimization applies a relative eigenvalue decomposition to compute multiple spatial filters in which across-trials variance relative to within-trials variance (i.e., ISC) is maximized.

Recent studies involving music have shown that EEG-ISC is modulated by listener expertise [29], musical tempo [30], temporal stimulus manipulations [30, 31], and salient musical events [31]. Auditory studies have reported small but significant group-mean ISC ($0.01 < r < 0.02$) in RC1, the maximally reliable spatial component. Repetition, explored through repeated listens of full excerpts, sometimes but not always results in lower ISC on repeated listens [29, 30]. However, the topic of repeating structural elements *within* a song has not yet been addressed.

2.4 Music-EEG Datasets

The acquisition and preparation of EEG data for analysis requires specialized expertise and sizeable investments in recording apparatus [3]. A key factor supporting MIR-EEG research is the growing number of open EEG datasets released with the intent for re-use by other researchers. Datasets vary in stimuli, stimulus manipulations, participant samples, listening tasks, additional response types, and EEG platforms used. Shorter stimuli are used in the MIIR dataset, comprising perceived and imagined responses to 12 excerpts 6.9–16.0 seconds in length [32] and the MAD-EEG dataset involving 78 solo, duet, or trio stimuli, each around six seconds long [14]. Datasets involving slightly longer excerpts include the DEAP dataset, with 40 one-minute excerpts from music videos [33]; MUSIN-G, with 12 excerpts, 100–132 seconds in length, from various genres [34]; and NMED-M, containing five-minute excerpts of various versions of a minimalist work [31]. Finally, a few datasets use complete musical works as stimuli: NMED-H includes four Bollywood songs [35], NMED-T uses 10 EDM-style songs [36], and NMED-E includes a cello concerto movement [37].

3. METHODS

3.1 EEG Dataset and Stimuli

Among the available datasets, we chose to work with NMED-H (Naturalistic Music EEG Dataset—Hindi) [35] as it used full-length pop (Bollywood) songs with repeating choruses as stimuli. Specifically, we work with the four "Intact" songs of the dataset: "Ainvayi Ainvayi", "Daaru

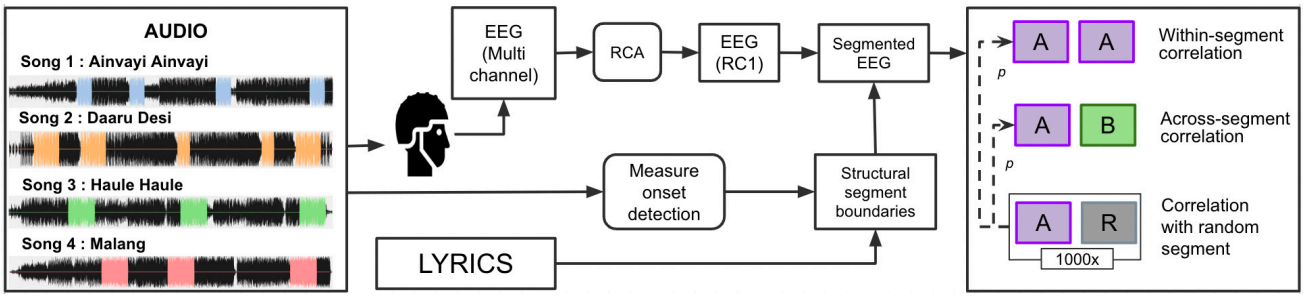


Figure 1. Analysis overview. The NMED-H dataset contains EEG responses recorded while 48 participants listened to four full-length Bollywood songs. We used RCA to compute a spatial EEG component in which ISC was maximized, and used the stimulus audio and lyrics to identify chorus segmentation boundaries to further epoch the EEG. For each song, correlations were performed within and across choruses, as well as between choruses and segments epoched at random.

Desi", "Haule Haule", and "Malang". Each song is around 4 min 30 sec in length and contains between 3 and 5 choruses as illustrated in color in Fig. 1. The stimuli were assumed to be new to the participants, who did not understand the Hindi-dialect song lyrics. We used the preprocessed, 125-channel EEG data sampled at 125 Hz with average reference; each song contained 24 trials from 12 unique participants (48 participants total) as each participant had listened to their assigned song twice.

3.2 EEG Analyses

To analyze the EEG, we followed an established procedure of spatial filtering followed by correlation calculations (Fig. 1). We used a publicly available RCA implementation [1] to compute a single spatial filter across all four songs. We computed RCA across entire song durations and not just chorus segments, as our permutation testing procedure involved segments sampled from throughout each song (see § 3.3). We then analyzed the vectorized form of single EEG trials from only the maximally reliable component RC1, as previous studies have shown that that component explains most of the ISC in EEG responses to music [30, 31]. Thus, the response data for each song was a time-by-trial matrix, with 24 trials from 12 participants for each song and a variable number of time samples per song.

To identify and segment song choruses, we first identified structural segment boundaries at the measure level using lyrics. [2] Next, we used a publicly available beat-tracking algorithm [38] to identify audio sample indices of the boundaries and converted those time stamps to the sampling rate of the EEG to segment the EEG accordingly.

Correlations were performed on a per-song basis, in two broad categories. *Within-chorus correlations* involved pairwise correlations among response trials from a single chorus instance, producing a symmetric matrix whose diagonal (being 1) was excluded from further analysis. *Across-chorus correlations* involved the cross-correlation of two matrices, each representing a different chorus instance. These correlations produced asymmetric matrices, since no response vector was ever correlated with itself. Each correlation also involved both *intra*-subject correlations (IaSC) of non-identical trials from the same participant and *inter*-subject correlations (ISC) of trials from different participants. As illustrated in Fig. 2, with 24 trials per song comprising two listens from each of 12 participants, within-chorus correlations produced for each participant one IaSC value (first listen and second listen) and 22 ISC values, excluding the diagonal. Across-chorus correlations produced for each participant four IaSC values (reflecting two distinct chorus instances $\times$ two distinct listens) and 88 ISC values. For each calculation, we computed mean correlations at the participant as well as the group level.

3.3 Statistical Analyses

We assessed statistical significance over distributions of per-participant results ($N = 12$). For RQ1 we used permutation testing: Each analysis was performed over 1000 pairs of segments of the same length as the true chorus segments, but with one segment epoched from a random start time in the song. The 1000 results served as the null distribution against which we compared the true result to compute the p-value. For RQ2 and RQ3 we used nonparametric Wilcoxon signed-rank tests to account for variable standard deviations of the sampling distributions caused by the discrepancy in the number of samples in each group (i.e., IaSC versus ISC; within- versus across-chorus). We performed one-sided tests in accordance with our expected results (RQ2 H_1: $within > across$; RQ3 H_1: $I_aSC > ISC$). We corrected for multiple comparisons using False Discovery Rate [39] on a per-song basis for RQ1 and RQ3 and on a per-song, per-condition basis for RQ2. We report statistically significant results ('***', '**') and also indicate but do not summarize marginally significant results ('*') for this first exploratory analysis.

4. RESULTS

4.1 Individual Correlations

We correlated vectors of spatially filtered, single-trial EEG on a per-song basis, both among responses to single choruses as well as across pairs of different choruses. The re-

[1] https://github.com/dmochow/rca
[2] https://gaana.com/, https://www.jiosaavn.com/

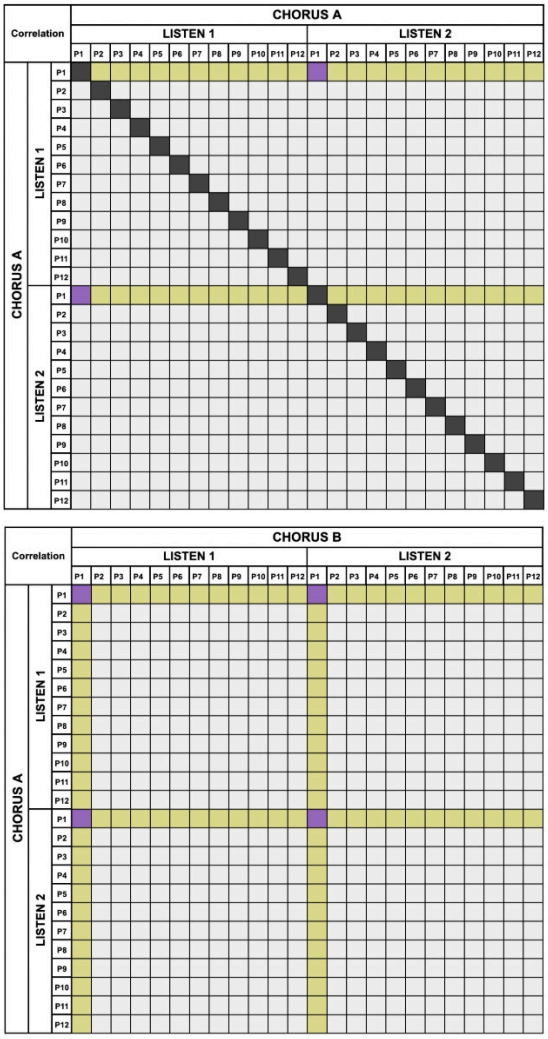


Figure 2. Illustration of IaSC and ISC matrix elements for Participant 1 of 12; each participant heard their assigned stimulus twice. **Top**: Within-chorus correlation produces a symmetric 24×24 matrix. The same IaSC correlation appears twice (purple), along with 22 unique ISC correlations (yellow). **Bottom**: Across-chorus correlation produces an asymmetric matrix with four unique IaSC correlations (2 choruses × 2 listens) and 88 unique ISC correlations.

sulting correlation matrices could then be partitioned into correlations from the same participant (IaSC) and different participants (ISC). Results are visualized in Fig. 3 and provided numerically in Tab. 1. After multiple comparison correction, 10 of 15 within-chorus IaSC and 2 of 22 across-chorus IaSC were statistically significant ('**' or '***'). For ISC, 14 of 15 within-chorus calculations and 12 of 22 across-chorus correlations were significant. IaSC distributions tended to have larger variance than ISC distributions, both at the participant level for single analyses (Fig. 3) and across the group means (Tab. 1).

4.2 Within- versus Across-Section Correlations

We assessed whether within-chorus correlations—involving identical musical content and context—were higher than across-chorus correlations, which are structurally similar but not identical. Tab. 2 summarizes the statistical significance of each comparison. After correcting for multiple comparisons, within-chorus IaSC was found to exceed across-chorus IaSC 7 times, while within-chorus ISC was higher than across-chorus ISC 4 times. Significant (and marginally significant) results most often implicated the first chorus of a song.

4.3 Intra- versus Inter-Subject Correlation

For our last analysis, we assessed whether IaSC—being computed from the same listener's data—would exceed ISC. Contrary to our expectations, one-sided Wilcoxon signed-rank tests revealed that after multiple comparison correction, IaSC did not exceed ISC for any within- or across-chorus correlation.

5. DISCUSSION

MSA has leveraged various representations—e.g., audio, lyrics, human annotations—to model human perception of musical structure. In this study we have answered the call for new forms of human response data to inform this task [1] and explored perception of repeated structure segments using brain data. Specifically, we assessed whether EEG responses to repeating choruses of four Bollywood songs were significantly correlated.

We found that EEG responses within and across choruses of a song were often significantly correlated, particularly for ISC. While small, these ISCs are on par with those reported in previous auditory EEG studies [30,31,40]. Correlating across choruses contrasts with past ISC research, which considered correlation only among responses to a single stimulus. That precedent may be due to those studies using predominantly narrative stimuli, such as movies or speeches, which generally do not include repeated segments. But for music, repetition is often integral to structure, from brief melodic motifs to large-scale elements [41]. The present use of ISC to assess music structure similarity is also a departure from its previous application to index brain states of attention and "engagement" in relation to attributes of stimuli (e.g., narrative tension, temporal coherence) [4, 30, 31] or participants (e.g., trained versus untrained musicians) [29]. Future research could consider data from spatial components beyond RC1 and further explore relationships between EEG correlation, music structure, and repetition to index both content similarity and listener engagement with repeated content.

We found that within-chorus correlation occasionally but not consistently exceeded across-chorus correlation; future research is needed to elucidate the role of acoustical or contextual differences across chorus instances in this result. Notably, within-chorus correlation most often exceeded across-chorus correlation in a song's first chorus. Past studies have shown that EEG-ISC often drops upon repeated exposures to full stimuli [4, 29, 30], and music-discovery engagement has been shown to be highest for first choruses compared to subsequent instances [42]. While this might lead one to expect higher ISC during

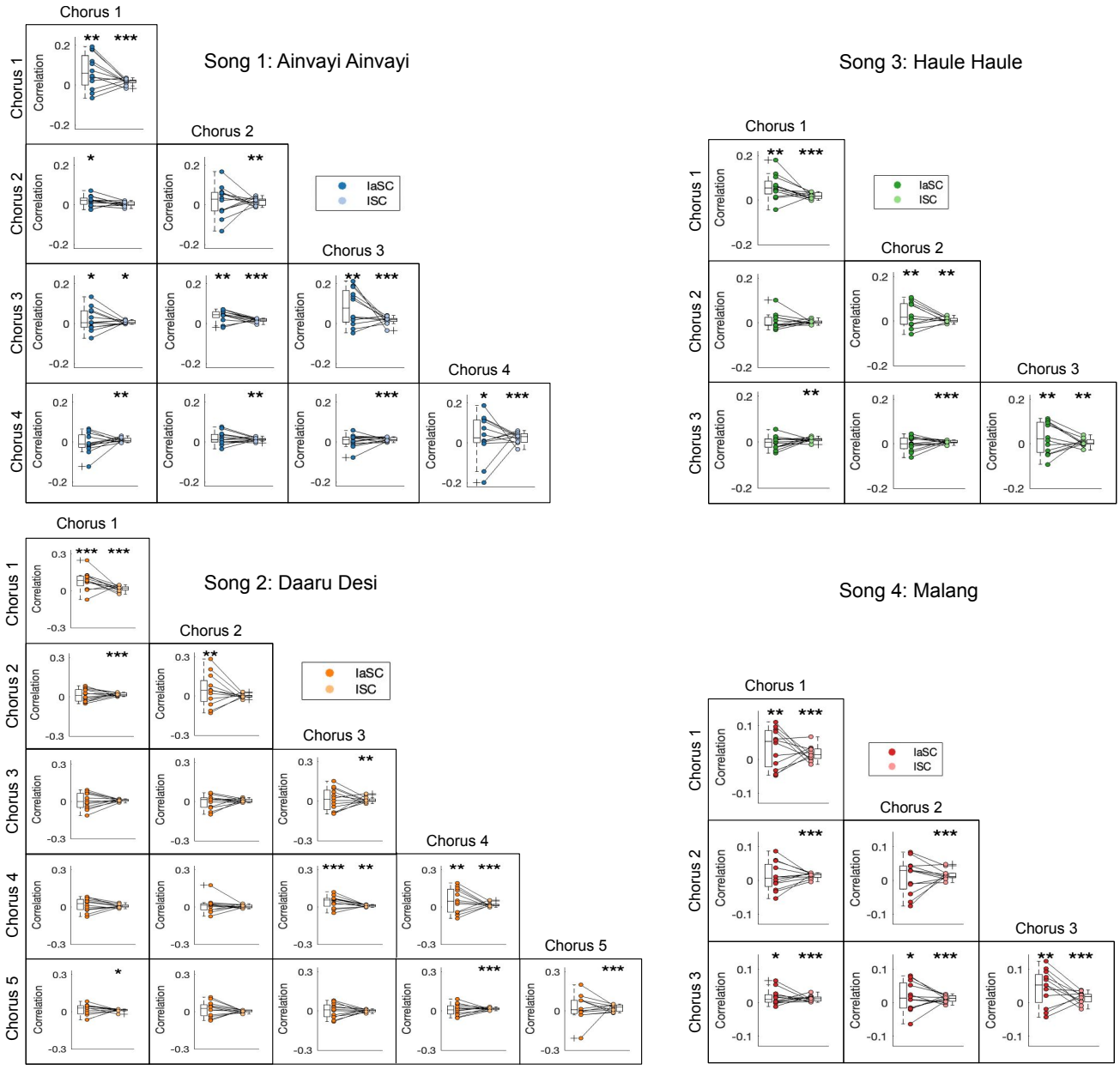


Figure 3. EEG correlations within and across choruses of four Bollywood songs. Each plot shows a distribution of intra- (IaSC) and inter- (ISC) subject correlation values across the 12 participants assigned to that song. Statistical significance of each correlation is denoted as p = 0 *** 0.01 ** 0.05 * 0.10 for FDR-corrected p-values.

the first chorus, current results do not suggest that within-chorus correlation drops as a song progresses. However, it may be that listeners have a unique perceptual experience of first choruses relative to other choruses.

Our expectation that IaSC would exceed ISC was not supported by the data. The large variance of IaSC relative to ISC, and greater number of significant ISC results despite lower group means, suggests that ISC ultimately provided a more stable estimate of neural correlation. Whether this is due to IaSC comprising fewer correlations, or an advantage of correlating across a heterogeneous sample of listeners, can be further investigated to inform future study designs.

This study contributes a first step toward using EEG data for MSA. While we focused on establishing similarity of neural responses among pre-identified repeating segments, and not detection of repeated segments or segment boundaries, our findings lay a foundation for multiple avenues of future work. For instance, a multimodal MSA framework could incorporate EEG measures of similarity alongside music content representations and human annotations. Other EEG-ISC analysis configurations may also prove useful for MSA: For instance, Dauer et al.'s finding that ISC computed over short time windows peaked during salient musical events including structural segment boundaries [31] is worth exploring further. Returning, too, to established connections between ISC and engagement, using ISC to identify highly engaging portions of songs could inform audio thumbnailing. Finally, while the present work leveraged an existing dataset, future studies could be de-

		IaSC					ISC				
		C1	C2	C3	C4	C5	C1	C2	C3	C4	C5
Song 1	C1	0.069**					0.017***				
	C2	0.020*	0.016				0.004	0.016**			
	C3	0.021*	0.041**	0.083**			0.008*	0.017***	0.017***		
	C4	-0.006	0.019	0.005	0.028*		0.010**	0.014**	0.013***	0.026***	
Song 2	C1	0.082***					0.019***				
	C2	0.013	0.046**				0.019***	0.001			
	C3	0.004	-0.003	0.013			0.010	0.007	0.010**		
	C4	0.018	0.014	0.044***	0.053**		0.010	0.006	0.011**	0.020***	
	C5	0.015	0.021	0.001	0.010	0.024	0.013*	0.006	0.001	0.016***	0.023***
Song 3	C1	0.059**					0.020***				
	C2	0.007	0.030**				0.003	0.005**			
	C3	-0.003	0.000	0.026**			0.010**	0.008***	0.007**		
Song 4	C1	0.036**					0.017***				
	C2	0.012	0.012				0.015***	0.017***			
	C3	0.017*	0.019*	0.045**			0.013***	0.012***	0.014***		

Table 1. Intra- and inter-subject correlation coefficients within and across choruses of four Bollywood songs. Statistical significance of correlations (FDR-corrected p-values) is denoted as p = 0 *** 0.01 ** 0.05 * 0.10.

		IaSC					ISC				
		C1	C2	C3	C4	C5	C1	C2	C3	C4	C5
Song 1	C1	-	*	*	*		-	**	**	*	
	C2	ns	-	ns	ns		ns	-	ns	ns	
	C3	ns	ns	-	**		ns	ns	-	ns	
	C4	ns	ns	ns	-		*	*	ns	-	
Song 2	C1	-	**	**	**	**	-	ns	ns	ns	ns
	C2	ns	-	ns	ns	ns	ns	-	ns	ns	ns
	C3	ns	ns	-	ns	ns	ns	ns	-	ns	ns
	C4	ns	ns	ns	-	ns	ns	ns	ns	-	ns
	C5	ns	ns	ns	ns	-	ns	*	*	ns	-
Song 3	C1	-	***	***			-	***	**		
	C2	ns	-	*			ns	-	ns		
	C3	ns	ns	-			ns	ns	-		
Song 4	C1	-	ns	ns			-	ns	ns		
	C2	ns	-	ns			ns	-	ns		
	C3	ns	ns	-			ns	ns	-		

Table 2. Results of one-sided Wilcoxon signed-rank tests assessing whether within-chorus correlation exceeds across-chorus correlation. Statistical significance of correlations (FDR-corrected p-values) is denoted as p = 0 *** 0.01 ** 0.05 * 0.10; 'ns' denotes non-significance.

signed to address specific MSA questions with newly collected EEG data. In all, we do not propose that EEG should or could replace existing data modalities for MSA, but rather highlight potential insights from EEG that may complement other existing approaches and inputs.

5.1 Limitations

We acknowledge limitations of this work. First, while we report multiple significant results, they do not imply generalizability: The correlations are small, and our findings—while promising—are not conclusive across all calculations. Next, we chose NMED-H as a ready-to-use EEG dataset of responses to popular songs containing repeated choruses. But the small stimulus set of four songs also hinders generalization, and future confirmatory studies should utilize a larger song set. We note that the original design of NMED-H specified that participants not be familiar with the songs or the language of their lyrics [35]. This too may limit generalizability, as more familiar or lyrically understandable songs may result in different EEG correlations.

Another main limitation is that while the song choruses crucially elicited the EEG data, they were only treated as repeating segments, and we did not consider nuances of placement or content attributes of individual choruses. Yet such features are known to impact perceptual and neural responses to choruses [26]. Thus, future research should consider finer-grained characterizations of music segments treated as structurally similar. One concrete next step could involve cross-modal comparisons of music similarity—for instance, whether similarity measures derived from audio, lyrics, or human annotations predict neural similarity.

Lastly, we trained RCA once over all available trials. Future work should incorporate cross-validation—iteratively optimizing the RCA spatial filter on training data and then applying it to holdout test trials—into the analysis pipeline to avoid overfitting.

6. CONCLUSION

MSA is an MIR topic with rich applications in audio thumbnailing, motif-finding, music summarization, music recommendation, and automatic music generation. Aiming to expand the scope of data modalities that may inform this task, we have contributed a first look at structural repetition using brain data. We used a publicly available EEG dataset and analyzed single-trial responses to choruses from four Bollywood pop songs by computing intra- and inter-subject correlations within and across choruses. We find that neural responses do often synchronize to a significant extent, which suggests that similarity among repeated choruses may translate to neural similarity. These findings motivate future studies of music similarity perception and highlight EEG data as a promising input to multimodal MSA systems.

7. ACKNOWLEDGMENTS

The authors thank Jacek Dmochowski for helpful advice on the statistical analyses.

8. 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*, vol. 3, no. 1, 2020.

[2] J. Van Balen, J. A. Burgoyne, F. Wiering, R. C. Veltkamp *et al.*, "An analysis of chorus features in popular song," in *Proceedings of the 14th Society of Music Information Retrieval Conference*, 2013.

[3] 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*, 2015, pp. 538–544.

[4] J. P. Dmochowski, P. Sajda, J. Dias, and L. Parra, "Correlated components of ongoing EEG point to emotionally laden attention—a possible marker of engagement?" *Frontiers in Human Neuroscience*, vol. 6, 2012.

[5] R. B. Dannenberg and M. Goto, *Music Structure Analysis from Acoustic Signals*. Springer, 2008, pp. 305–331.

[6] M. Goto, "A chorus-section detecting method for musical audio signals," in *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03).*, vol. 5. IEEE, 2003, pp. V–437.

[7] J. Paulus and A. Klapuri, "Labelling the structural parts of a music piece with Markov models," in *Computer Music Modeling and Retrieval. Genesis of Meaning in Sound and Music: 5th International Symposium, CMMR*. Springer, 2009, pp. 166–176.

[8] A. Eronen and F. Tampere, "Chorus detection with combined use of MFCC and chroma features and image processing filters," in *Proc. of 10th International Conference on Digital Audio Effects*, 2007, pp. 229–236.

[9] B. McFee, O. Nieto, M. M. Farbood, and J. P. Bello, "Evaluating hierarchical structure in music annotations," *Frontiers in psychology*, vol. 8, p. 1337, 2017.

[10] R. S. Schaefer, J. Farquhar, Y. Blokland, M. Sadakata, and P. Desain, "Name that tune: Decoding music from the listening brain," *NeuroImage*, vol. 56, no. 2, pp. 843–849, 2011.

[11] S. Stober, D. J. Cameron, and J. A. Grahn, "Classifying EEG recordings of rhythm perception." in *Proceedings of the 15th International Society for Music Information Retrieval Conference*, 2014, pp. 649–654.

[12] M. S. Treder, H. Purwins, D. Miklody, I. Sturm, and B. Blankertz, "Decoding auditory attention to instruments in polyphonic music using single-trial EEG classification," *Journal of neural engineering*, vol. 11, no. 2, p. 026009, 2014.

[13] G. Cantisani, S. Essid, and G. Richard, "EEG-based decoding of auditory attention to a target instrument in polyphonic music," in *2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2019, pp. 80–84.

[14] G. Cantisani, G. Trégoat, S. Essid, and G. Richard, "MAD-EEG: An EEG dataset for decoding auditory attention to a target instrument in polyphonic music," in *Speech, Music and Mind (SMM), Satellite Workshop of Interspeech 2019*, 2019.

[15] G. Cantisani, S. Essid, and G. Richard, "Neuro-steered music source separation with EEG-based auditory attention decoding and contrastive-NMF," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 36–40.

[16] S. Stober, T. Prätzlich, and M. Müller, "Brain beats: Tempo extraction from EEG data," in *Proceedings of the 17th International Society for Music Information Retrieval Conference*, 2016, pp. 276–282.

[17] M.-S. Kim, G. Y. Lee, and H.-G. Kim, "Multi-channel EEG classification method according to music tempo stimuli using 3D convolutional bidirectional gated recurrent neural network," *The Journal of the Acoustical Society of Korea*, vol. 40, no. 3, pp. 228–233, 2021.

[18] G. Y. Lee, M.-S. Kim, and H.-G. Kim, "Extraction and classification of tempo stimuli from electroencephalography recordings using convolutional recurrent attention model," *ETRI Journal*, vol. 43, no. 6, pp. 1081–1092, 2021.

[19] A. Vinay, A. Lerch, and G. Leslie, "Mind the beat: Detecting audio onsets from EEG recordings of music listening," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 231–235.

[20] 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, 2018, pp. 392–399.

[21] N. Gang, B. Kaneshiro, J. Berger, and J. P. Dmochowski, "Decoding neurally relevant musical features using Canonical Correlation Analysis," in *Proceedings*

of the 18th International Society for Music Information Retrieval Conference, 2017, pp. 131–138.

[22] J. R. Katthi and S. Ganapathy, "Deep multiway Canonical Correlation Analysis for multi-subject EEG normalization," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 1245–1249.

[23] C. Foster, D. Dharmaretnam, H. Xu, A. Fyshe, and G. Tzanetakis, "Decoding music in the human brain using EEG data," in *IEEE 20th International Workshop on Multimedia Signal Processing (MMSP)*, 2018, pp. 1–6.

[24] E. B. Abrams, E. Muñoz 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*, 2022.

[25] 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*, 2020, pp. 566–573.

[26] S. Sangnark, P. Autthasan, P. Ponglertnapakorn, P. Chalekarn, T. Sudhawiyangkul, M. Trakulruangroj, S. Songsermsawad, R. Assabumrungrat, S. Amplod, K. Ounjai, and T. Wilaiprasitporn, "Revealing preference in popular music through familiarity and brain response," *IEEE Sensors Journal*, vol. 21, no. 13, pp. 14 931–14 940, 2021.

[27] U. Hasson, Y. Nir, I. Levy, G. Fuhrmann, and R. Malach, "Intersubject synchronization of cortical activity during natural vision," *Science*, vol. 303, no. 5664, pp. 1634–1640, 2004.

[28] J. P. Dmochowski, A. S. Greaves, and A. M. Norcia, "Maximally reliable spatial filtering of steady state visual evoked potentials," *NeuroImage*, vol. 109, pp. 63–72, 2015.

[29] J. Madsen, E. H. Margulis, R. Simchy-Gross, and L. C. Parra, "Music synchronizes brainwaves across listeners with strong effects of repetition, familiarity and training," *Scientific reports*, vol. 9, no. 1, pp. 1–8, 2019.

[30] 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.

[31] T. Dauer, D. T. Nguyen, N. Gang, J. P. Dmochowski, J. Berger, and B. Kaneshiro, "Inter-subject correlation while listening to minimalist music: A study of electrophysiological and behavioral responses to Steve Reich's Piano Phase," *Frontiers in Neuroscience*, vol. 15, 2021.

[32] S. Stober, A. Sternin, A. M. Owen, and J. A. Grahn, "Towards music imagery information retrieval: Introducing the OpenMIIR dataset of EEG recordings from music perception and imagination." in *Proceedings of the 16th International Society for Music Information Retrieval Conference*, 2015, pp. 763–769.

[33] 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, 2012.

[34] K. P. Miyapuram, N. Ahmad, P. Pandey, and J. D. Lomas, "Electroencephalography (EEG) dataset during naturalistic music listening comprising different genres with familiarity and enjoyment ratings," *Data in Brief*, vol. 45, p. 108663, 2022.

[35] B. Kaneshiro, D. T. Nguyen, J. P. Dmochowski, A. M. Norcia, and J. Berger, "Naturalistic music EEG dataset—Hindi (NMED-H)," in *Stanford Digital Repository*, 2016. [Online]. Available: http://purl.stanford.edu/sd922db3535

[36] S. Losorelli, D. T. Nguyen, J. P. Dmochowski, and B. Kaneshiro, "NMED-T: A tempo-focused dataset of cortical and behavioral responses to naturalistic music," in *Proceedings of the 18th International Society for Music Information Retrieval Conference*, 2017, pp. 339–346.

[37] B. Kaneshiro, D. T. Nguyen, J. P. Dmochowski, A. M. Norcia, and J. Berger, "Naturalistic music EEG dataset—Elgar (NMED-E)," in *Stanford Digital Repository*, 2021. [Online]. Available: https://purl.stanford.edu/pp371jh5722

[38] D. P. W. Ellis, "Beat tracking by dynamic programming," *Journal of New Music Research*, vol. 36, no. 1, pp. 51–60, 2007.

[39] Y. Benjamini and D. Yekutieli, "The control of the false discovery rate in multiple testing under dependency," *The Annals of Statistics*, vol. 29, no. 4, pp. 1165–1188, 2001.

[40] J. J. Ki, S. P. Kelly, and L. C. Parra, "Attention strongly modulates reliability of neural responses to naturalistic narrative stimuli," *Journal of Neuroscience*, vol. 36, no. 10, pp. 3092–3101, 2016.

[41] E. H. Margulis, *On repeat: How music plays the mind*. Oxford University Press, 2014.

[42] B. Kaneshiro, F. Ruan, C. W. Baker, and J. Berger, "Characterizing listener engagement with popular songs using large-scale music discovery data," *Frontiers in Psychology*, vol. 8, p. 416, 2017.

CHROMATIC CHORDS IN THEORY AND PRACTICE

Mark R. H. Gotham
Durham University

ABSTRACT

'Chromatic harmony' is seen as a fundamental part of (extended) tonal music in the Western classical tradition (c.1700–1900). It routinely features in core curricula. Yet even in this globalised and data-driven age, 1) there are significant gaps between how different national 'schools' identify important chords and progressions, label them, and shape the corresponding curricula; 2) even many common terms lack robust definition; and 3) empirical evidence rarely features, even in discussions about 'typical', 'representative' practice. This paper addresses those three considerations by: 1) comparing English- and German-speaking traditions as an example of this divergence; 2) proposing a framework for defining common terms where that is lacking; and 3) surveying the actual usage of these chromatic chord categories using a *computational* corpus study of *human* harmonic analyses.

1. INTRODUCTION

Different traditions for teaching music theory come with divergent terminology. These gaps often correspond to national trends (or 'schools') and to the different languages used. As always with language, these gaps can take several forms. Some terms may be *shared* by the two languages, so no translation is needed. Other times, a term is present *in one language only*; this inclusion may indicate an importance for the term/concept on one side of this divide and not the other. More complex still, two languages may have some terms with *partially overlapping* meaning.

There are significant gaps between English- and German-speaking terminology for chromatic harmony, despite so much shared historical heritage. Even the distinction of 'chromatic' from 'diatonic' betrays an English-language stance. Section §1.1 introduces something of a frame for this comparison and §2 discusses three interesting case-studies of 'canonical' terms. The focus is on chords that are either *intrinsically* chromatic (Augmented Sixths, §2.2), or chromatic against their diatonic *context* (Neapolitan Sixths, §2.1; Modal Mixture, §2.3). We leave to one side what is sometimes called 'functional chromaticism' (the 'secondary'/'applied' chords involved in tonicisation/modulation – see [1, Part 5]) though the final section (§5) briefly considers some relevant chord *progressions*.

© M. Gotham. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** M. Gotham, "Chromatic Chords in Theory and Practice", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

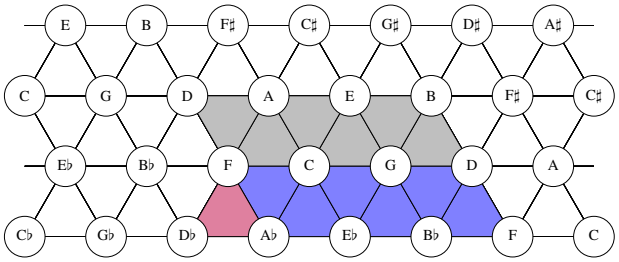


Figure 1. A 'Tonnetz' diagram of tonal space. Major and minor triads in the key of C *major* are grey; those in C (natural) *minor* are in blue, and the 'Neapolitan' is purple.

Moreover, a closer look reveals that even some of the apparently core concepts in chromatic harmony are only vaguely defined. For example, although 'modal mixture' is common (at least in US-English music theory), no source sets out comprehensive criteria for inclusion in this category. Section §3 addresses this, seeking to establish not so much a single, definitive answer, but a *framework* to deal with the various issues involved.

Finally, having established the (range) of terms that German- and English- speaking scholars elevate as important, and clarified the meaning of some, §4 provides an initial overview of the relative *usage* of these chords in the 'When in Rome' repository: a meta-corpus of all Roman numeral analyses that human annotators have encoded in computational formats [2]. In doing so, we gain insight into how common these chords are, at least in the repertoires covered and the view of those human analysts.

Clearly, sheer *usage* is not the only relevant consideration for the *significance* of a chordal category — many subjects are interesting partly because of their rarity. In any case, all such discussions, and any claims about 'general practice', need a basis in this kind of empirical evidence. The clarity that evidence brings may prompt a review of our existing practice (how we categorise these chords, and/or how much time we devote to them in our curricula and wider musical practice). It may also clarify the extent to which that attention is based on the frequency of occurrence as opposed to some other factor, like how *explainable* the concept is in terms of a particular theory.

1.1 Textbooks, Terminology, and Tradition

We begin with that slippery notion of a 'tradition'. While it is hard to pin down exactly what this means in practice, [1] the contents of widely circulated textbooks provide

[1] For more on the question of 'representativeness', see [3].

one kind of insight into what is 'typically taught and commonly known' in a given context. Among the issues here is the privileging of contexts in which textbooks are common (broadly speaking, the US), and lack of sensitivity to more flexibly amassed materials, particularly in a changing world with ever more materials shared ever more accessibly online. [2] Then again, many of these online materials and apps continue to reflect what is described here in terms of textbooks. And I implicate myself in this: see, for example, the chapter listings and content of the 'Open Music Theory' (OMT) textbook [5] which (incidentally) serves throughout this paper as a go-to resource for further reading, with links to relevant chapters provided.

1.1.1 English-language (hereafter 'Anglophone')

On the Anglophone side we benefit from two surveys of the 'core curriculum' in American music theory teaching, including information about the textbooks typically used [6, 7]. The more recent of these surveys finds that 91.89% (238/259) respondents include 'Chromatic harmony' in their core curriculum (see Table IV-1), with 1 or 2 semesters being the 'most commonly reported lengths of time for teaching' this content (p.202, Table IV-9), and that 79.92% use textbooks/anthologies (Table IV-10).

These surveys also appear to indicate that the preference for *which* textbook to use changes quickly, [3] but that *what* those textbooks cover remains largely the same: they consistently cover the same canonical collection including at least the so-called 'Neapolitan sixths', 'Augmented sixths', and 'Modal Mixture'. [4]

Click on those terms above for OMT chapters about them, and click here for a summary of these chords in a musical score that you can view, play and more online (no login required). That rendering is relatively typical of the simple, purportedly 'prototypical' ways these chords are set out in textbooks. (Naturally, we will discuss here just how prototypical they really are.)

1.1.2 German-language (hereafter 'DACH')

As no equivalent survey existed for the DACH side, [5] we conducted one anew in mid-2022. [6] Specifically, we asked anyone teaching chromatic harmony at a German-speaking tertiary education institution to answer basic questions about the textbooks and terminology they know and use. Please refer to that study for a thorough report on the method and results of the survey. This paper refers to only the most salient results as relevant for present purposes, as discussed in the following sections.

[2] On the growing adoption of technological alternatives see the 'What Do Musicologists Do All Day' (WDMDAD) surveys (2014-15, [4] and 2021-22, forthcoming) which investigate 'the use of technology in the work of music researchers in the widest sense' (including teaching).

[3] I.e., there is little overlap between the 2000 and 2017 results.

[4] Increasingly, many also refer to the common-tone diminished sevenths, (for which see §5) though they often package this more deeply e.g., within the 'Rise of Symmetrical Harmony in Tonal Music' [1].

[5] 'DACH' is an abbreviation/acronym for Germany, Austria and Switzerland. These are the main areas of German-speaking today and where all the institutions approached for the survey are situated.

[6] The written report is forthcoming (Feilen, Schnauss and Gotham).

2. THREE CANONICAL CATEGORIES

2.1 Similar usage: the 'Neapolitan sixth'

The 'Neapolitan sixth' appears routinely in both languages. It is interesting to note the status of this chord in relation to the *Funktions*- and *Stufentheorie* approaches to harmony which capture much of the core divide between DACH and Anglophone approaches (respectively).

The Neapolitan can be seen as a simple, one-semi-tone modification to the minor subdominant. [7] In *Funktionstheorie*, such small transformations typically indicate close harmonic relations, leading to maps of tonal space like the *Tonnetz* of figure 1 which shows how the Neapolitan sits alongside diatonic chords, especially in minor. [8] (We will return to the minor-specific aspect *in practice* in §4). *Stufentheorie*, by contrast, typically describes the Neapolitan in terms of a modification of the second degree ($\flat\mathrm{II}^6$). This is clearly relegated to a subsidiary position, a 'chromatic' chord outside the main, 'diatonic' set. [9]

Notwithstanding the different theoretical frames, the Neapolitan presents relative close Anglophone-DACH agreement: not only is there agreement on which pitches are involved, but both typically relate this chord to the 'subdominant' (both), or more loosely to 'predominant' function (Anglophone). Despite the Anglophone notion of $\flat\mathrm{II}^6$, the close relation to 'iv' ('s') is often emphasised, and likewise it is common in DACH to eschew the possible *Funktions*-only explanation in favour of the symbol s^n that further emphasises the proximity to the subdominant.

Anglophone and DACH traditions also share most of the definitional incompleteness, notably terms of whether to admit: other *inversions* (e.g., 53) and other *tones* (e.g., seventh chords such as 653). DACH theory often *does* admit the 53 configuration of this chord, and reserves a special name for it: the *verselbständigter*. It is noteworthy that, despite being rather sparing in its use of special terms for individual chords, DACH considers the Neapolitan worthy not only of one term, but two. [10] Both Anglophone and DACH theories lack an explicit consensus on whether the Neapolitan may have a seventh.

2.2 Divergent terms: Augmented-sixth chords

The Anglophone convention for teaching Augmented-sixth chords identifies (at least) three forms that have been given spurious national labels: the 'Italian sixth' (63), the 'French' (643), and the 'German' (653). Those labels seem

[7] This can be viewed as the *Mollsubdominantgegenklang* (sG), though see the following text on s^n. The *-gegenklang* transformation is the same as the *-gegenparallel* and better known in Anglophone contexts as the *leittonwechsel* or 'leading-tone exchange'.

[8] Although this common visual analogy for tonal 'space' is familiar to Anglophone music theory, is much more closely related to the *Funktions* mentality. The earliest, recognisable form seems to be from Euler [8] (yes, the mathematician) but the best known exposition of this idea and 'space' is that of Riemann [9] (no, not the mathematician).

[9] DACH can also express this chromatic alteration (*hoch-* and *tiefalterierte*), but usually does so a last resort where other theory fails.

[10] One of the earliest recorded Anglophone uses of this term treads a middle line in which the chord is explicitly built on the subdominant scale degree ('Fa', i.e., $\hat{4}$) and 'is never inverted', apparently meaning that, unusually, this 63 form is not to be considered 'inverted' [10].

to have originally been proposed (c.1800) based on their usage in the repertoire. For example, [11] explicitly links these chords to the music of those nations. [11]

Leaving until §4 the question of whether these national labels have anything to do with repertoire usage, there is an Anglophone/DACH division in the terms themselves which may perhaps be telling. DACH emphasises a single category for which the recognised term is *übermäßiger Quintsextakkord*. This explicitly refers to the 653 form — the one that Anglophone theory calls the 'German' sixth.

This also indicates opposing ways of handling augmented sixth chord categories: Anglophone traditions not only use 3 categories, but tend to start with the 'Italian' (63) as the prototype (at least in the pedagogical sense) and then *add* tones to build the French (6*4*3) and German (6*5*3); DACH, by contrast, starts with the 653 and would need to *remove or modify* from there. [12]

These differences aside, there is broad Anglophone-DACH agreement on the composition of the chords. The eponymous augmented sixth interval is needed (and spelt as such), and there is a strong focus on both the *inversion* that sees the lower note of that interval in the bass and the *voice-leading* whereby this interval expands 'out' to a perfect octave on the dominant ($\hat{5}$).

2.3 Anglophone only: 'Modal Mixture'

Most Anglophone textbooks offer a short definition of 'modal mixture' (a.k.a. 'borrowing') as the use of a chord that is not diatonic in the key specified, but would be in the parallel (German: *variant*) major / minor and can therefore be thought of as a 'mixture' of major and minor modes, or a 'borrowing' from the one to the other. Some coverage of this topic is present in all the Anglophone textbooks surveyed, usually with a dedicated chapter.

No DACH equivalent appears in German textbooks. Equivalents do sporadically appear in DACH analysis scholarship with terms such as *Dur-moll-Austauschbarkeit*, (or simply *Austauschbarkeit*, literally 'exchange'), but this term cannot be assumed knowledge in the classroom or beyond. [13]

Despite the ubiquity of the term 'mixture' in Anglophone textbooks, it is particularly under-defined and never fully unpacked to account for all in-/exclusions. This is perhaps understandable in a pedagogical context where the increased clarity must be weighted against the corresponding complexity, but as a field, we clearly need a framework for robust definition. The following, dedicated section §3 provides such a framework.

[11] Callcott appears to have inherited the term 'Italian', noting that it 'has been termed' the Italian. There's no direct reference, though nearby mention of Rousseau suggests that may be at least one of his sources. Callcott seems to introduce the other two 'nationalities'.

[12] Click here for a modern, online example of this DACH pattern, and see also Biamonte's account of this chord, including DACH sources dating back to Marpurg 1755 [12].

[13] Incidentally, it is not self-evident that this 'mixture' is indeed a mixture of distinct parts, as opposed to a unified entity. For instance, another school of thought (historically of German-origin, now more common in Russian music theory) sees the major mode with $\flat\hat{6}$ as a single 'harmonic major' scale. See [13] for the progress of this idea from Hauptmann, via Iogansen, Liadov, and Rimsky-Korsakov to modern Russian theory.

3. DEFINING MODAL MIXTURE

In a major context, the subdominant is also major ('IV' or 'S'). Probably the most common chord identified in terms of modal mixture is the minor variant of this subdominant ('iv' or 's'). So in C major, for example, we would have <F-A♭-C> in place of <F-A-C>.

But what if this mixture chord had a seventh, so not simply <F-A♭-C>, but <F-A♭-C-E♭>, or <F-A♭-C-E>? The first case, <F-A♭-C-E♭> seems like a very good candidate: the additional borrowing from the minor of E♭ further strengthens the case for mixture. The same can't be said for <F-A♭-C-E> as E belongs to C major exclusively and arguably counts *against* the notion of mixture. [14] So **should cases of clear *non-mixture* be excluded**?

If we admit the <F-A♭-C-E> as a case of modal mixture, then what do we have to say about the case of <F-A-C-E♭>? Is that equivalent? Now the E♭ is borrowed, but the A is arguably not depending on the type of minor mode. **What minor form are we talking about when we speak of mixture?** Some accounts seem to hint at the natural minor, but then every raised leading-tone chord (V, V7, viio, . . .) would count as cases of mixture in minor.

Should the case of <F-A-C-E♭> depend on whether it is cast as IV♭7 or as V/♭VII? That is, **should secondary/applied chords be handled differently** as a case of 'functional' chromaticism or (put another way) as diatonic elements in a new key area? Does this depend on whether that secondary tonality is realised by a subsequent **tonicisation or modulation**? This question opens a second set of possible criteria: in addition to questions about the chord's *content*, we now must also consider its *context*.

Speaking of context, **does the so-called 'Picardy Third' count?** [15] And arguably related to both content and context, (and certainly relevant to applied chords) is the question: **does pitch spelling matter?** Were our <F-A-C-E♭> chord spelt as <F-A-C-D♯>, apart from potentially leading analysts to describe it differently, should that spelling itself have a bearing on the status of mixture? Is the minor third mixed only when spelt as such, or is it to be handled as a pitch class, and thus admitting the enharmonic equivalent of a raised second degree ($\sharp\hat{2}$)?

Altogether, these musical questions capture something of the ambiguity in defining modal mixture, and the need for greater clarity in what 'counts'. They also suggest the need to create a *framework* for category membership, rather than clear-cut rules applicable in all contexts.

Realising this, functionality at 'When in Rome' enables user-defined answers to any of the questions raised above, while also providing default settings and proposing a system for grading the *relative* strength of mixture, both in terms of the chord *content* and of the surrounding chord *context*.

[14] Note we are talking specifically about how relatively *mixed* these chords are, not how *chromatic*.

[15] This term stands for the practice of ending a minor key passage with a major tonic as the final chord. (Click here for the modal mixture chapter of OMT, including an example of the 'Picardy Third'.) It is extremely common, at least in some repertoire contexts.

3.1 Which pitches, which minor?

In working towards a relative gradation of mixture (which may be paired with strict requirements/exclusions), we begin with an account of how *each pitch* can add to or detract from the mixture status. This necessarily also involves the question of 'which minor', a conundrum which often complicates matters of definition in tonal music.

Many definitions of modal mixture restrict themselves to natural minor specifically (minor $\hat{6}$ and $\hat{7}$), yet they do not describe V in minor as a mixture, despite the raised $\hat{7}$ clearly belonging to major and not the chosen minor form. Tones' mixture status can be organised in a few categories: *clearly* indicative of one mode and not the other; *possible* indication of mixture; and *neutral/shared*. The following categorisation is logically guided, but only one (set of) opinion(s). Users of this framework are free to re-allocate the status of these pitches (within reason).

3.1.1 Clear (non-)mixture: m3, M3, m6

Tones strongly indicate (non-)mixture when they clearly belong in either the host mode or the parallel mode but not both. The clearest example is scale degree $\hat{3}$. The minor third (m3) is a clear case of mixture when it appears in major (hereafter min→maj mixture) and non-mixture when in minor. Likewise, vice versa, for the major M3: this is a clear case of mixture in minor (hereafter maj→min) and non-mixture in major. (Again, these comments are separate from the *context* caveats discussed elsewhere, e.g., concerning the 'Picardy Third'.)

The minor sixth (m6) in major is almost as clear: it is not in the major scale and does belong to both natural and harmonic minors, as well as the descending melodic minor form. Only the ascending melodic minor misses this pitch. When in Rome defaults suggest the inclusion of m6 as a case of *clear* mixture, in the definition framework, while enabling theorists to categorise it instead as a case of *possible* mixture if they prefer for a specific repertoire/task.

3.1.2 Possible mixture: M6, m7, M7

Some tones offer a lower level of *possible* mixture due to their considerably more ambiguous status. When in Rome proposes the major sixth degree (M6) for this category as it is more strongly associated with major, though it can be reached in one melodic minor form (ascending). M6 may therefore indicate *possible* maj→min mixture.

Likewise, the minor seventh degree (m7) may indicate min→maj mixture: it does not feature in major, but it is also not as strongly indicative of minor mode as m6 is, appearing only in natural and descending melodic forms.

Finally, as discussed, the major seventh degree (M7) arguably indicates maj→min mixture, though raised leading-tones are too common in tonal music to support this as a chromatic category.

3.1.3 Neutral (1, 2, 4, 5) and 'chromatic' (♯1, ♯4)

Neutral tones belong to both major and minor forms equally. This group includes scale degrees 1, 2, 4, and 5 along with any tones excluded from the above categories.

That leaves tones which may be called 'chromatic' in the sense that they do not belong to either mode. We can confidently populate this category with degrees $\sharp\hat{1}$ and $\sharp\hat{4}$. If the user asserts that spelling matters, then the chromatic category also hosts enharmonics (like $\sharp\hat{2}$, discussed above).

3.2 Metrics and/or Categories

If we accept the notion some chords are more strongly indicative of mixture than others, largely because of the relative status of the tones, then we may wish to explicitly weight that relative strength, note by note. For example, *clearly* mixed tones might attract twice the weight (2) of *possible* mixture (1), with *neutral* values at 0. *Chromatic* tones are perhaps the most ambiguous. When in Rome defaults to a value of -1, because their clearly chromatic status often detracts from their candidacy for mixture.

For instance, to return to the above example cases of min→maj mixture: the strength of mixtures like ♭VI, ♭VI7, and iv7 derives from that fact that they all feature the m6 and m3, and all avoid any detractions. The weighting values above would grade each of these at 4, twice the strength of chords like iv with only the m6 (no m3, but also no detractions) at 2. The pros and cons of an ambiguous chords combining m6 and M3 would effectively balance out.

One asset of this weighting-by-tone metric is its flexibility: it enables any chord to be assessed, including modifications like added/altered tones, and it can handle the enharmonic question separately. Context can be handled either categorically (e.g., excluding all secondaries) and/or with further weightings. For instance, the status of mixture may be enhanced when it is bookended by clearly *non-mixed* chords as in I-iv-I (T-s-T). Again, see 'When in Rome' for a demonstration.

4. IN PRACTICE (CORPUS STUDY)

All of the above discussion – 'national' category variants, graded definitions, and more – would benefit from comparison with the actual usage in practice. For instance, if a chord is *not* commonly used in a particular style but *is* commonly taught in courses purporting to represent that style, then we need to be clear on the reasons why.

Part of the difficulty of establishing robust definitions of the chords above comes from the fact that a robust definition of the 'chord' itself is challenging. Western classical notation includes information about which *pitches* to play, and when, but has no explicit statements on how they connect as *chords*. [16] It differs in this (and other) respects from leadsheets, for example, where it *is* typical to include chord symbols. [17] While many explicit algorithms for automatic harmonic analysis have been proposed, none really approaches the quality of a human expert. And arguably the best automatic analysis systems to have emerged in recent years are those based on machine learning, which derive, in turn, learns from the *computer* encodings of *human* expert analyses discussed here [14].

[16] Baroque figured bass is arguably a partial exception: given the bass note and figuring, you have something like a chordal analysis.

[17] Though they are not key-relative like Roman numerals.

The assessment of chordal usage 'in practice' here is based on that data, and specifically the 'When in Rome' repository, which provides a synthesis of all those *computer* encodings of *human* analyses for Western classical music using Roman numerals.

As with all analysis, this is inherently subjective; while the score source material may have editorial ambiguities that evade the notion of 'ground truth', this is all the more so in analytical commentary on that source. Then again, the harmonic analyses are our subject of interest, and so this subjectivity is not only inevitable, but also desirable.

Once the analyses have been encoded in a suitable format (legible to human and machine alike), although there are still operational decisions to make, the process of extracting them is readily implemented and interpreted. The operational decisions include filters for more or less detailed versions of the chords used as best befits the research question at hand. For example, it is sometimes necessary to *retain* inversion information, while at other times it is best to report on aggregated data *excluding* inversion.

Every such option is fully, openly implemented in extensively documented and tested code at 'When in Rome' to allow maximum re-use and adaptation for future research. Moreover, that repository presents the percentage usage per basic chord type in dedicated files, separated both by sub-corpus and for major-*versus*-minor contexts.

From this alone, we can assess the relative usage of our 'canonical' chords. Any such survey highlights the extreme predominance of basic tonic and dominant function chords (c.75% of the total). Chromatic chords are certainly marginal in relation to this, but we are more concerned here with how common the chromatic chords are *relative to each other*. Figure 2 provides an example of the summative data and visualisations provided on 'When in Rome', in this case for the example of Augmented Sixth chords in the lieder sub-corpus, divided (as discussed) into separate data for major-*versus*-minor tonal contexts.

In addition to the source repo, anyone interested can interact with this data on OMT's chromatic harmony anthology (click here) where instances of these chromatic chord types can be browsed in sortable tables, in their full score context, and in few-bar excerpts.

4.1 Results for the Three Chromatic Categories

For each of the three 'canonical' chromatic categories discussed above, this section provides some high-level observations from the evidence of the corpus, and it considers the implications these observations might have for reviewing our attitudes to those chords.

The **Neapolitan sixth** is used relatively little. The main use cases in the lieder sub-corpus are '♭II6' and '♭II65' in minor (c.0.5%). Another c.0.4% accounts for the other Neapolitan candidates in minor, and there is very little use in major contexts at all. Other corpora broadly bear out this trend, and with even less usage of the seventh chords. Even here in the lieder, many of the '♭II65' sevenths cases occur in progressions against an inverted pedal, potentially suggesting a possible sub-category for this specific device.

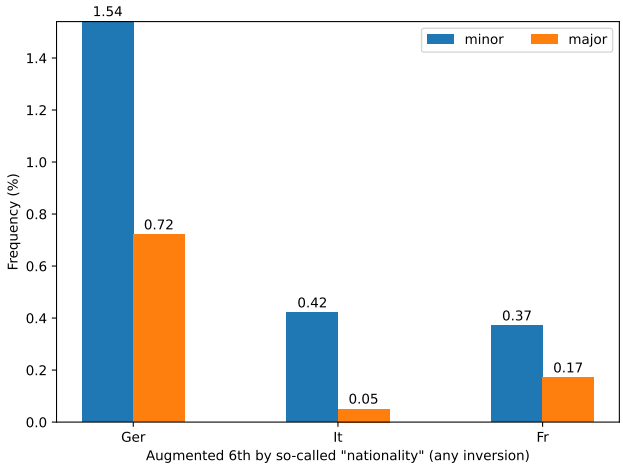


Figure 2. Augmented sixths chords in the Lieder corpus.

The fact that Neapolitans are so commonly taught is somewhat is contrary to the evidence of usage, perhaps prompting a review of the importance attributed to them, especially in the 'category light' DACH tradition.

Augmented sixth chords are much more common. For instance, in the lieder the 'German' (653) in minor alone accounts for over 1.5%, and thus more than all the possible Neapolitans in both modes. Within the augmented sixth category, it is notable that the 'German' (653) is so much more common than the other forms, and that all forms are much more common in the minor mode context. The DACH practice of concentrating teaching and terminology of *augmented sixths* on the 653 form arguably receives support from this usage-in-practice evidence.

Modal mixture is much more common still, but very unevenly so. The kinds of strong candidates for mixture described above occur relatively infrequently, for instance, with only '♭VI' making a short-list of top-10 cases (c.0.1%). Much (c.10x) more common are moderate mixtures like i (c.1.2%) and iv (c.1.0%). This extremely varied extent of usage reinforces the need for a distinction between types or grades of *relative* mixture strength.

It is perhaps also notable that the 'other' chromatic chord categories discussed (Augmented and Neapolitan sixths) feature among the most common cases of possible 'mixture'. They all pose a strong case for mixture, (especially the 'German' which features both of the main mixed tones), but they also have the detraction of chromatic notes (at least $\sharp\hat{4}$ for the Augmented Sixths; $\flat\hat{2}$ for Neapolitans). This may prompt a review or clarification of categories which, in turn, speaks to wider issues such as the 'French' sixth's status in relation to tritone substitution (again, see [12]), the 'bebop' dominant seventh with diminished fifth, and even some secondary dominants.

5. PROGRESSING TO CHORD PROGRESSIONS

This brief paper has set out some of the musical, computational, and even national/institutional issues at stake in defining chromatic chords and commenting on their use in practice, focussing on three individual chromatic chords.

Clearly this is only an initial step towards developing recommendations for how to define chords and describe 'general' practice in a given repertoire.

There is plenty of opportunity for future work, not least in growing the datasets (their sheer scale, repertoire coverage, and range of analytical perspectives), and in widening the range of both chordal categories and the languages/'schools' considered. Another clear next step is to expand the remit from individual *chords* to chord *progressions*. This is not so clear-cut a distinction as it may seem.

We close with some examples. Once again, all of the logic discussed here is implemented in the When in Rome repository, and examples are presented in both the 'Anthology' section of that repo, and in the more browsing-friendly format of the OMT harmony anthology.

5.1 Chord or progression? The Case of the 'Cto7'

Some chromatic cases sit ambiguously between 'chord' and 'progression'. As discussed, mixture is arguably an example: we certainly have to take account of the modal context (iv is diatonic in minor but mixture in major) and we may also chose to have additional contextual requirements such as the elimination of secondary dominants that resolve, and/or of the 'Picardy Third' endings.

The common-tone diminished seventh chord ('Cto7') presents an example that nudges further into the realm of progressions. Once again, we describe a single chord, though certainly need a wider contextual view. Here the chord's construction as a fully diminished seventh is required, but only a small part of the definition which otherwise relies on the context of at least the following chord. Almost certainly required is a common-tone with the *following* chord ...which is not a suspension.[18] Not usually required, (though potentially strengthening the case) is use of a common-tone with the *preceding* chord. And the case is arguably stronger still if the preceding and following chords are the same, indicating more of a *prolongation*.

5.2 Anglophone/DACH Progressions

The comparison of Anglophone and DACH traditions can, of course, continue to chord progressions. The 'Cto7' does not feature in DACH traditions, though a related form known in Anglophone circles as the 'Omnibus' does have a relative in the DACH concept of the *Teufelsmühle* [15].

Not yet at the textbook level, Lewandowki [16] recently proposed a category pair for *fallender Quintanstieg* (hereafter, fQ) and *aufsteigender Quintfall* (hereafter, aQ) both of which see pairs of fifth in the same direction, separated by a step in the opposite direction. For example, D-A-C-G would be an instance of the fQ, while G-C-A-D would be a case of the aQ.

Instances of these progressions can be extracted by any corpus, functional or otherwise.[19] Filtering When in

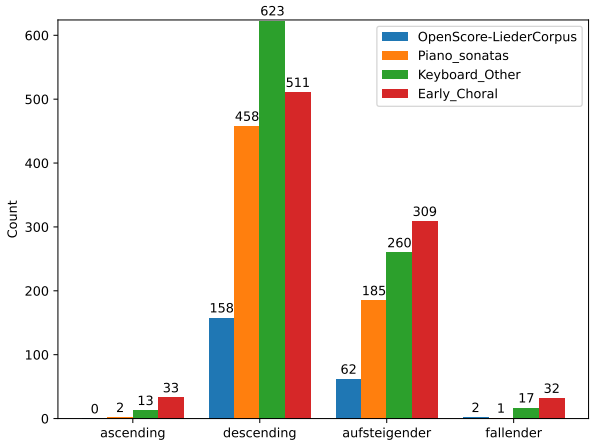


Figure 3. Fifth progressions by category and corpus.

Rome for the minimal, 4-chord instances of each shows that the aQ is much (c.20x) more common than the fQ across all corpora (RHS of fig.3). The better known pattern of rising/falling cycles of fifths are related in that they also feature pairs of fifths a step apart. Filtering for 4-grams of these progressions reveals a similar, and even more extreme (c.50x) preference for one direction (falling) over the other (rising). So once again, while we may seem to have a class of equal schema *in theory*, the usage *in practice* highlights an imbalance that arguably needs including at the outset of teaching these materials.

5.3 Beyond the Anglophone-DACH Comparison

We close with an example progression originating in another language, and with the additional constraint of having an expected *position* for its usage, thus further expanding the *context* we need to assess.

'Partimenti' treatises originating in 18th-century Italy have enjoyed a renewed interest from music theorists in recent years [17]. This method centres on prototypical, schematic patterns that can serve as the basis of composition (including improvisation). The schema are typically defined by their bass and melodic lines, their harmony, and their position both in relation to the metre and the large-scale form.[20] Harmonic analysis data captures all of this except the melodic line. For example, most aspects of the *Quiescenza* are captured by progressions like I-V7/IV-IV-V-I,[21] and by the expected position at the *end* (coda) of a work. These textbooks provide repertoire examples, and there are certainly cases in the meta-corpus (which includes a sub-corpus of Corelli Trio Sonatas) that fit.

However, counter-examples are also easy to find and an initial survey of overall usage finds no tendency towards end-section emphasis in any sub-corpus, including the Corelli.[22] Once again, the data suggests that it is time for a thorough re-evaluation of schematic associations passed pedagogically from one generation to the next.

[18] The progression of viio7/V to the cadential 64 is common, but a weak candidate for the Cto. It is excluded by most definitions (e.g., on OMT), though it may be significant as an historical origin for this progression.

[19] As the labels are not dependent on key-context or RN labelling, it is reasonable to include pop examples here (as Lewandowki does). For example, the fQ (D-A-C-G) is the chord progression of TLC's *Waterfalls*.

[20] Click here for examples in the relevant section of OMT.

[21] Again, the code sets out how to catch all and only the relevant cases.

[22] The code includes functionality for plotting usage-by-position.

6. ACKNOWLEDGEMENTS

Thanks to all who have contributed to the (rather long) gestation of these ideas. Clearly the international aspect reflects my time working in several countries. Thanks to all colleagues and students for the exchanges! For mixture especially, I have canvassed opinions from the (computational) music theory community since around 2020, on music21 (as part of creating the `.isMixture()` method here), on Twitter (here), and elsewhere (several public talks). Thanks to all who engaged with this . . . or even simply heard me out!

7. REFERENCES

[1] S. G. Laitz, *The Complete Musician: An Integrated Approach to Theory, Analysis, and Listening*, 4th ed. New York: Oxford University Press, 2016.

[2] M. Gotham, G. Micchi, N. Nápoles-López, and M. Sailor, "When in Rome: a meta-corpus of functional harmony," *Transactions of the International Society for Music Information Retrieval*, expected 2023.

[3] J. London, "Building a Representative Corpus of Classical Music," *Music Perception: An Interdisciplinary Journal*, vol. 31, no. 1, pp. 68–90, 2013, publisher: University of California Press. [Online]. Available: http://www.jstor.org/stable/10.1525/mp.2013.31.1.68

[4] C. Inskip and F. Wiering, "In Their Own Words: Using Text Analysis to Identify Musicologists' Attitudes Towards Technology," in *Proceedings of the 16th International Society for Music Information Retrieval*, 2015.

[5] M. Gotham, K. Gullings, C. Hamm, B. Hughes, B. Jarvis, M. Lavengood, and J. Peterson, *Open Music Theory*, 2nd ed. VIVA Pressbooks, 2021. [Online]. Available: https://viva.pressbooks.pub/openmusictheory/

[6] R. B. Nelson, "The College Music Society Music Theory Undergraduate Core Curriculum Survey - 2000," *College Music Symposium*, vol. 42, pp. 60–75, 2002, publisher: College Music Society. [Online]. Available: https://www.jstor.org/stable/40374423

[7] B. Murphy and B. McConville, "Music Theory Undergraduate Core Curriculum Survey: a 2017 Update," *Journal of Music Theory Pedagogy*, vol. 31, no. 1, Jan. 2017. [Online]. Available: https://digitalcollections.lipscomb.edu/jmtp/vol31/iss1/9

[8] L. Euler, *Tentamen novae theoriae musicae ex certissismis harmoniae principiis dilucide expositae*. Saint Petersburg Academy, 1739.

[9] H. Riemann, *Vereinfachte Harmonielehre; oder, Die Lehre von den tonalen Funktionen der Akkorde [Harmony Simplified: Or, the Theory of the Tonal Functions of Chords]*. London: Augener, 1893.

[10] W. Crotch, *Elements of musical composition : comprehending the rules of thorough bass, and the theory of tuning*. London : Printed for Longman, Hurst, Rees, Orme, & Brown by Nathaniel Bliss, Oxford, 1812. [Online]. Available: http://archive.org/details/elementsofmusica00crot

[11] J. W. Callcott, *A musical grammar, in four parts: 1. Notation; 2. Melody; 3. Harmony; 4. Rhythm.* London: Printed by B. Macmillan for R. Birchall, 1806, open Library ID: OL14794362M.

[12] N. Biamonte, "Augmented-Sixth Chords vs. Tritone Substitutes," *Music Theory Online*, vol. 14, no. 2, Jun. 2008. [Online]. Available: https://mtosmt.org/issues/mto.08.14.2/mto.08.14.2.biamonte.html

[13] P. Ewell, "Harmonic Functionalism in Russian Music Theory: A Primer," *Theoria*, vol. 26, 2020.

[14] N. Nápoles López, M. Gotham, and I. Fujinaga, "AugmentedNet: A Roman Numeral Analysis Network with Synthetic Training Examples and Additional Tonal Tasks," in *Proceedings of the 22nd International Society for Music Information Retrieval Conference*. Online: ISMIR, Nov. 2021, pp. 404–411. [Online]. Available: https://doi.org/10.5281/zenodo.5624533

[15] M.-A. Dittrich, "›Teufelsmühle‹ und ›Omnibus‹," *Zeitschrift der Gesellschaft für Musiktheorie [Journal of the German-speaking Society of Music Theory]*, vol. 4, no. 1–2, pp. 107–121, 2007, publisher: Gesellschaft für Musiktheorie. [Online]. Available: https://www.gmth.de/zeitschrift/artikel/247.aspx#abstract

[16] S. Lewandowski, "›Fallende Quintanstiege‹," *ZGMTH*, vol. 7, no. 1, pp. 85–97, 2010. [Online]. Available: https://www.gmth.de/zeitschrift/artikel/508.aspx

[17] R. O. Gjerdingen, *Music in the Galant Style*. Oxford ; New York: Oxford University Press, 2007.

Papers – Session III

BPS-MOTIF: A DATASET FOR REPEATED PATTERN DISCOVERY OF POLYPHONIC SYMBOLIC MUSIC

Yo-Wei Hsiao[1] **Tzu-Yun Hung**[1,2] **Tsung-Ping Chen**[1] **Li Su**[1]
[1]Academia Sinica, Taiwan [2]National Taiwan Normal University, Taiwan
lisu@iis.sinica.edu.tw

ABSTRACT

Intra-opus repeated pattern discovery in polyphonic symbolic music data has challenges in both algorithm design and data annotation. To solve these challenges, we propose BPS-motif, a new symbolic music dataset containing the note-level annotation of motives and occurrences in Beethoven's piano sonatas. The size of the proposed dataset is larger than previous symbolic datasets for repeated pattern discovery. We report the process of dataset annotation, specifically a peer review process and discussion phase to improve the annotation quality. Finally, we propose a motif discovery method which is shown outperforming baseline methods on repeated pattern discovery.

1. INTRODUCTION

Repetition is ubiquitous in music. Computational discovery of repeated patterns in music data has been long discussed in the field of music information retrieval (MIR). Aside from its importance in music analysis [1], the role of repeated pattern discovery has also been noticed in music classification [2, 3] and generation [4, 5]. The definition of a pattern is multi-fold. Generally speaking, a pattern refers to a group of notes that serves a musically important role and occurs multiple times in a piece of music. Repeated patterns are known by various names, such as motifs, themes, phrases and sections, depending on their specific musical function. The goal of the *repeated pattern discovery* problem is then to find the relevant patterns (depending on the intended task) and all of their occurrences within the provided musical data.

Compared to other music analysis tasks (e.g., harmony analysis) on polyphonic symbolic music data, repeated pattern discovery is relatively less discussed due to mainly two challenges. First, searching for all the possible candidates of repeated patterns is costly and redundant [6]. The computational complexity of the algorithm is high, while the discovered patterns often have little musical significance [7]. Second, repetition is a non-exact attribute of music. A large pattern can be potentially divided into small ones; whether a note group constitutes a meaningful repeated pattern also depends on the subjective views regarding repetition, similarity, and musical importance. As a result, human-annotated datasets that comprehensively identify all the available patterns and all of their occurrences remains in a quite limited scale.

In this paper, we propose a new dataset, BPS-motif, to improve the scalability of music pattern discovery research. The BPS-motif dataset contains the note-level annotation of motives and their occurrences in the first movements of Beethoven's Piano Sonatas (BPS). This is an extension of the many previous musical annotations on BPS, such as the functional harmony, phrase and section annotation provided in the Beethoven Piano Sonata Functional Harmony (BPS-FH) dataset [8]. We are specifically interested in annotating the *motivic* units in the melody parts of each piece of music, which could be complementary to the more *thematic* annotation (e.g., phrases and sections) provided in the BPS-FH dataset. We expect that the proposed dataset can enrich not only multi-task MIR research but also novel computational music analysis tasks.

Besides, as another contribution of this paper, we also propose a simple yet effective algorithm for repeated pattern discovery. Different from previous works which emphasized the equal translations among notes, we emphasize the contextual relationships among short segments of notes. We demonstrate that the proposed algorithm not only outperform several baselines on the BPS-motif dataset, but also on the JKU-PDD dataset [9], the most widely used dataset for the discovery of repeated themes and sections. In other words, the proposed algorithm is competitive for finding both motivic and thematic patterns.

The rest of this paper is organized as follows. Section 2 gives a background introduction and a survey of previous works on the datasets and methods for repeated pattern discovery. In Section 3, we introduce the dataset and our proposed annotation process. In Section 4, we introduce the proposed motif discovery algorithm and demonstrate its evaluation results. Conclusions are made in Section 5.

2. RELATED WORK

2.1 Repeated pattern discovery datasets

The datasets for repeated pattern discovery are built mostly for the interest in computational music analysis research. Complete annotation of repeated patterns should incorporate all the note groups (each note in pitch-onset part) that

© Y.-W. Hsiao, T.-Y. Hung, T.-P. Chen and L Su. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Y.-W. Hsiao, T.-Y. Hung, T.-P. Chen and L Su, "BPS-Motif: A Dataset for Repeated Pattern Discovery of Polyphonic Symbolic Music", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

constitute 1) the patterns of interest and 2) the occurrences of each pattern. Usually, a music piece contains more than one pattern, and each pattern should repeat (i.e., occur more than twice). The occurrences of a pattern may not be the same; one occurrence can be an exact copy of, or a variation from another occurrence that belongs to the same pattern. The music data can be either monophonic or polyphonic, and can be in either symbolic or audio format. The annotation can be either *intra-opus* or *inter-opus* [10]. In the former case, the analysis focuses on how a piece of music is broken down into pattern occurrences by having occurrences of a pattern within one music piece [7, 10]. In the latter case, the analysis focuses on the evolution of common elements in a corpus, by having occurrences of a pattern in different pieces of music. It should be noted that the annotation in inter-opus datasets can be limited to only a small set of patterns, while intra-opus datasets need a comprehensive set of patterns and occurrences and is hard to build; see Table 1 and the discussion below.

Table 1 presents the datasets for both inter-opus and intra-opus pattern discovery. In the Saraga dataset, Srinivasamurthy *et al.* annotated 4,571 temporal occurrences from 1,067 *characteristic melodic phrases*, a musical unit related to the *rāga*, over 170 audio recordings [11]. Krause *et al.* performed large-scale leitmotif classification in audio recordings by annotating the time intervals of 10 leitmotifs in Richard Wagner's four-opera cycle *Der Ring des Nibelungen* and achieve a large scale of occurrence over 16 versions of recordings [12]. [1] In the MTC-ANN dataset, Kranenburg *et al.* categorized 93 patterns in 360 monophonic folk tunes and annotated 1,657 occurrences [13]. Finkensiep *et al.* considered 20 types of *schemata* and annotated 244 events in Mozart's piano sonatas [14]. In the MIREX campaign of *Discovery of Repeated Patterns and Themes*, Collins *et al.* firstly compiled an organized, open-source intra-opus pattern discovery dataset containing 165 occurrences in five pieces and this dataset has been widely discussed in the follow-up research works. It should be noted that, among these datasets, only the JKU-PDD dataset is for intra-opus pattern discovery but its size is smallest among all (only five pieces of music).

Aside from the above-mentioned datasets, it is still worth mentioning the datasets for *pattern matching* [15], such as the Dig That Lick dataset for Jazz music [16] and the Theme Finder for Classical music [17]. These datasets support pattern retrieval tasks with known query, but they neither support pattern discovery research nor provide the annotation of pattern occurrences explicitly.

2.2 Repeated pattern discovery methods

For symbolic music data, there are three major approaches to implementing the repeated pattern discovery algorithms: 1) string-based approach which represents music data as one-dimensional pitch sequence and finds repeated patterns with sub-string matching [18, 19]; 2) geometry-based approach which represents music data as multi-dimensional point sets (usually onset-pitch pairs in two-dimensional space) and retrieves the translatable subsets (see discussion below) as repeated patterns [20–22]; 3) feature-based approach which extracts or learns features from music data, and retrieves patterns with clustering or classification of the features [14, 23–25].

[1] There are in total 38,448 occurrences if counting the 16 versions.

	format	usage	#ps	#ptns	#ocrs
[11]	poly audio	inter	170	1,067	4,571
[12]	poly audio	inter	11	10	2,403
[13]	mono symbolic	inter	360	93	1,657
[14]	poly symbolic	inter	54	20	244
[9]	poly symbolic	intra	5	32	165
Ours	poly symbolic	intra	32	263	4,944

Table 1: Comparison of several open-source musical repeated pattern datasets including the saraga dataset [11], The *Ring* (one performance version) [12], MTC-ANN [13], Schemata [14], JKU-PDD [9], and BPS-motif (ours). The number of pieces (#ps), the number of individual patterns (#ptns), and the number of occurrences (#ocrs) are listed. The data formats can be monophonic (mono) or polyphonic (poly), audio or symbolic. The type of annotation can be inter-opus (inter) or intra-opus (intra).

While the string-based approach falls limited in representing polyphonic music [22], research efforts on pattern discovery have been more emphasized on the geometry-based approach. In the geometry-based approach, we consider a music piece $\mathbf{D}$ with N notes and $\mathbf{d}$ denotes a note. We have $\mathbf{D} := \{\mathbf{d}_i\}_{i=1}^{N}$, where $\mathbf{d}_i := (o_i, p_i)$ denotes the ith note, and o_i, p_i denote its onset and pitch value, respectively. In the discussion of the structure induction algorithm with translational equivalence classes (SIATEC) [20], two subsets (i.e., two patterns) $\mathbf{m}$ and $\mathbf{n}$ in $\mathbf{D}$ are translatable (denoted as $\mathbf{n} \equiv \mathbf{m}$) if there exists a vector $\mathbf{v}$ such that the translation function $f(\mathbf{d}, \mathbf{v}) : \mathbf{m} \to \mathbf{n}; \mathbf{d} \mapsto \mathbf{d} + \mathbf{v}$ is bijective. All the patterns translatable with respect to $\mathbf{m}$ form a translational equivalence class (TEC) of $\mathbf{m}$ in $\mathbf{D}$, that means

$$\mathrm{TEC}(\mathbf{m}, \mathbf{D}) := \{\mathbf{n} : \mathbf{n} \equiv \mathbf{m}, \mathbf{n} \subseteq \mathbf{D}\}\,. \quad (1)$$

A *maximal translatable pattern* (MTP) is the largest pattern translatable by a translatable vector $\mathbf{v}$ [20]:

$$\mathrm{MTP}(\mathbf{v}, \mathbf{D}) := \max_{|\mathbf{d}|}\{\mathbf{d} : \mathbf{d} \in \mathbf{D} \text{ and } \mathbf{d} + \mathbf{v} \in \mathbf{D}\}\,, \quad (2)$$

where $|\mathbf{d}|$ is the number of notes in $\mathbf{d}$. SIATEC is then an algorithm which finds all the TEC of the available MTPs in $\mathbf{D}$. A survey and comparative study can be found in [26].

In the feature-based approach, machine learning techniques are usually applied; features are processed by clustering for the pattern discovery task (when a query is not given), and by classification for the pattern matching task (when a query is given) [15]. For example, in [23], agglomerative clustering over the wavelet transform of the pitch sequence data was used for pattern discovery in melodies. In [14], music schema recognition was performed by extracting the schema candidates using a skip-gram model and then a binary classification on the rhythm

and pitch features over the candidates. It is also noted that the feature-based approach has also been widely discussed in the repeated pattern discovery of audio. In [24], Nuttall *et al.* adopted matrix profile, a time-series-based motif discovery method [27], on the predominant pitch contours to extract the characteristic melodic phrases from audio [11]. Krause *et al.* utilized recurrent neural networks (RNN) to classify over 30,000 leitmotifs over differnt performances of *Der Ring des Nibelungen* [25].

3. DATASET

3.1 Overview

The BPS-motif dataset contains the annotation of motives in the first movements of Beethoven's 32 piano sonatas. An annotation unit contains a group of motif notes and the corresponding motif label. The motif labels are sorted in alphabetical order: the motif that occurs first in the music piece is labeled as A, the secondly occurred motif is labeled as B, the thirdly occurred one is C, and so on. The group of notes which are the jth occurrence of the motif A in $\mathbf{D}$ is annotated as $\mathbf{m}_{A,j}$, $\mathbf{m}_{A,j} \subset \mathbf{D}$, $j \in \mathbb{Z}_{\geq 0}$. All the occurrences of this motif are annotated with A. Further information, such as the start time and end time of each motif occurrence, and the non-motif notes (i.e., the notes which do not belong to any motif) can be directly derived. The dataset is available at: `https://github.com/Wiilly07/Beethoven_motif`.

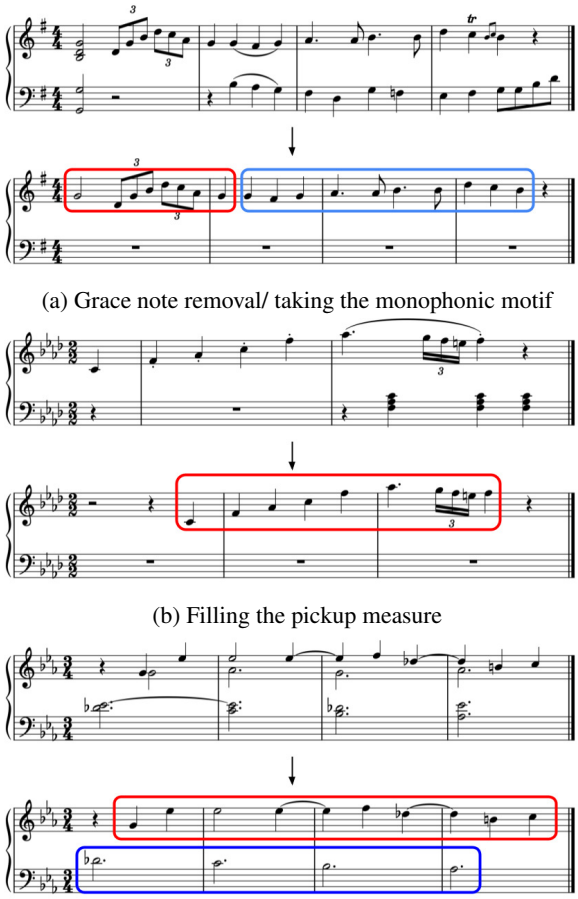
Over the 32 music pieces, we labeled 263 distinct motives with 4,944 occurrences in total (see Table 1). These occurrences contain 36,652 notes, which is 28.87% of the total number of 126,943 notes. For each piece of music, the number of motives ranges from 2 to 13 (average 8.22 motives), and the number of occurrences ranges from 41 to 290 (average 154.5 occurrences). On average, a motif contains 7.41 notes and spans 5.30 crochet beats. The pitch ranges of the motives are mostly within two octaves.

To facilitate the annotation process, we only consider the repeated patterns in melodic notes; that means, all the annotated motives are constrained to be a monophonic note sequence. For example, in Figure 1a, although the first beat of the first measure contains three notes (i.e., B3, D4, G4), only G4 is included in the the annotated motif `A0`. However, there can be multiple motives which are fully or partly overlapped in time; see the demonstration in Figure 1c (the red and blue boxes represent two overlapped motives).

3.2 Data format

We basically followed the data format adopted in the BPS-FH dataset. First, all the articulation symbols and grace notes were omitted (see Figure 1a). Second, pickup was filled when needed (see Figure 1b and the following discussion). Repeat signs are also unfolded when needed.

We take a crotchet beat as the unit time step (i.e., the duration of a crotchet is 1 in our note annotation and is 1 second in MIDI) to represent the data. Two types of timestamps are recorded. The *score time* takes the pickup measure as negative while the *MIDI time* fills the pickup



(a) Grace note removal/ taking the monophonic motif

(b) Filling the pickup measure

(c) Annotating overlapped motives

Figure 1: Examples of annotated motives. From (a) to (c), the three demonstrated excerpts are from Beethoven's Piano Sonata No. 20, No. 1, and No. 5, respectively. The notes bounded by a colored box form a motif occurrence.

measure and defines the beginning of the measure as 0. For example, in Figure 1b, the score time of the C4 note at the beginning is -1 while the MIDI time is 3. Both the score time and the MIDI time unfolds the repeat signs so the timestamps increase monotonically. Similarly, at the measure level, the *score measure number* is the measure number counted on the score sheet (the pickup measure is measure 0, with repeat signs), while the *MIDI measure number* takes the pickup measure (if there is) as measure 1 and unfolds the repeat signs. Two types of pitch number are recorded: the MIDI pitch (in MIDI number) and the morphetic pitch number [28].

For each piece of music, we provide annotation data in different formats for users to retrieve the motif events in different ways. The file formats include:

1. A multi-track MIDI file that records the motif notes. Temporally overlapped motives are recorded in different tracks. There are at most four tracks in our annotation of this dataset.
2. A list of all the notes. Each note has the labels of 1) onset time (in score time), 2) MIDI pitch number, 3) morphetic pitch number, 4) note duration (in

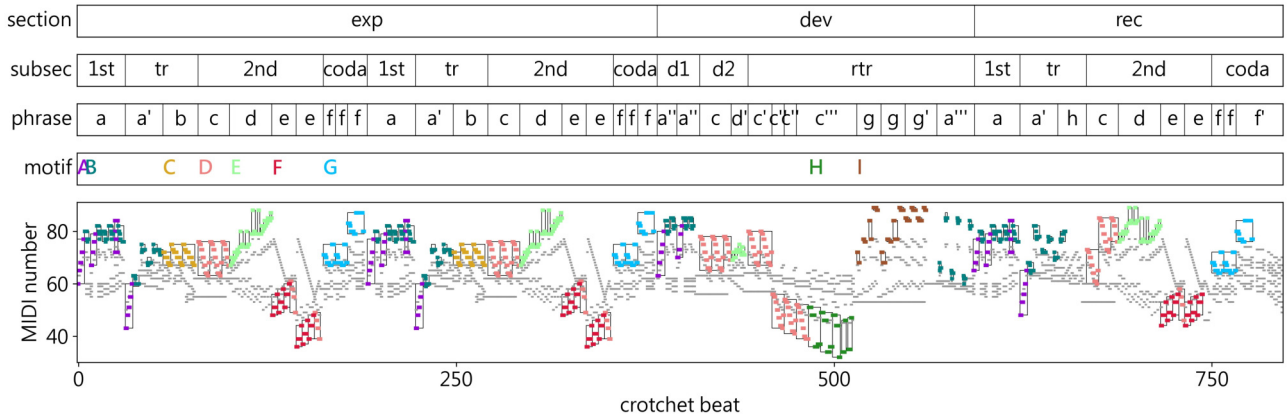


Figure 2: Motives and occurrences labeled in Beethoven's Piano Sonata No.1 in F minor. From top to bottom shows the annotation of section intervals, subsection intervals, phrase intervals, the time when a new motif occurs (with motif labels), and the piano roll of the music piece marked with motif and non-motif notes. In the bottom subfigure, different motives are specified by different colors. Motif occurrences are marked with a black bounding box. Non-motif notes are in gray color.

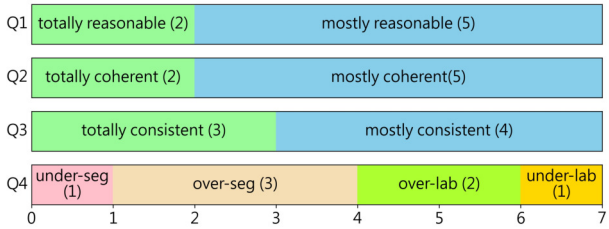


Figure 3: Assessment results (Q1 to Q4) regarding the data annotation from the seven reviewers. The results were collected before the discussion phase.

crotchet beats), 5) staff number (integers from zero for the top staff), 6) MIDI measure number, and 7) motif (e.g., a note is annotated as A if it is part of A). The notes without motif labels are non-motif notes.

3. Individual note lists of each motif occurrence. These lists are provided for users to better retrieve each occurrence. The labels in these lists are the same as the ones in the list of all notes.
4. A list describing the properties of motif occurrences. Each motif occurrence has the labels of 1) the start time and end time (in both score time and MIDI time), 2) the duration of the occurrence, 3) the measure number where the motif start (in both score measure and MIDI measure), 4) the "start beat" of the motif start, and 5) time signature.

We also provide the PDF scoresheets with the annotator's manual annotations and notes. These scoresheets are for reference only because they are the raw annotations and may not be the same as the annotation of our final version; see Section 3.3 for more details about the annotation process. The note lists and the score time are compatible with the BPS-FH dataset, therefore annotation of more thematic units (e.g., theme, sub-section and phrase) can be retrieved from the BPS-FH dataset. To better see our annotation result, Figure 2 illustrates the hierarchical musical structures with motives of Beethoven's Piano Sonata No. 1, combining the section, subsection and phrase labels in BPS-FH, and the motif labels in BPS-motif. [2]

3.3 Annotation process

There are a few challenges in the data annotation process. First, as mentioned, identifying of musical motifs and their repetitions or variations in a piece of music is not straightforward. Ambiguity arises from multiple factors. For example, some repeated patterns may not be considered as valid musical motifs, a motif may not always be the smallest unit of a repeated pattern, and the similarity or difference between two such sequences can also be subject to human interpretation. Besides, while experienced musicians can read the scoresheet and mark the motives directly by hand on it, converting such hand-drawing annotations into database formats still requires lots of efforts.

Our proposed approach to build the BPF-motif dataset incorporates three parts: annotation, review, and score typing. First, two annotators (the first and the second authors) manually annotate the motives on the scoresheet. Each piece is annotated by one annotator. Then, we invite external reviewers to review annotated scoresheets. Also, the reviewer helps us digitize the manual annotation. In the review process, we design a review form to let the reviewers assess the overall quality of annotation and also provide their suggested annotation if they hold different opinions. The review form contains the following questions:

1. (Q1) Are the annotations reasonable? (3: totally reasonable; 2: mostly reasonable; 1: unreasonable)
2. (Q2) Are the annotations coherent with your opinion? That means, if you were the annotator, will you

[2] It should be noted that there are still some annotation inconsistency between the BPS-FH and BPS-motif datasets. For example, in Figure 2, the phrase c is constructed only with the motif D, while the phrase c''' is constructed only with the motif H. This means that while the annotator of BPS-FH considered c''' as simply a variation of c, the annotator of BPS-motif considered them being different (and are thereon constructed with different motives).

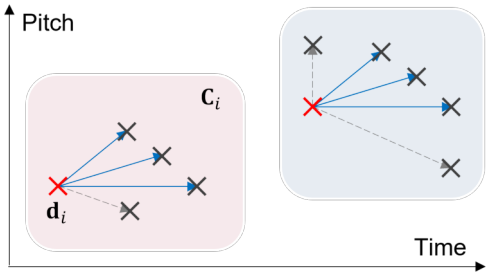


Figure 4: Two segments (light gray and light purple regions) and their common structure. The crosses indicate notes, and the set of vectors in a segment represents its structure. Blue vectors denote the common structure which exists in both segments. The dashed gray arrows represents non-motivic notes within the two segments.

also have the same annotations as ours? (3: totally coherent; 2: mostly coherent; 1: incoherent)

3. (Q3) Are the annotations consistent (i.e., did we hold consistent criteria annotating the data)? (3: totally consistent; 2: mostly consistent; 1: inconsistent)
4. (Q4) In which way your opinions are different from ours? (a: we took multiple motives into one (*undersegmentation*); b: we divided a motif into many (*oversegmentation*); c: we took some non-motif patterns as motives (*overlabeling*); d: we omitted some musically important motives (*underlabeling*)) Choose one even you totally agree to our annotation.
5. (Q5) If you hold different opinions on our annotation and think we should revise them, leave your comments explicitly. Your comments can be, for example, "the motif E in Sonata No. x should be further divided into F and G" (describe what F and G are); "the motif H in Sonata No. y can be considered as a variation of B and should be merged," etc.

Seven reviewers were invited to review the annotations. The reviewers are all from composition background and are good at using computer scorewriters. Each reviewer was assigned from 3 to 7 pieces (according to the length of the music piece) for review, then they answered the above questions and provided their suggested annotations on a co-edited document. During the review and discussion phase, the reviewers also need to convert the manual annotation on the score into the symbolic form using the scorewriter MuseScore. This confirms that they had carefully read the annotation, and also speed up the process of building the dataset. After the reviewers typed the scores of the annotated motives, we can directly convert it to MIDI and the final annotation data.

The reviewer's assessment results are shown in Fig. 3. From Q1 to Q3, it is shown that no reviewer reported our annotation as unreasonable, incoherent to their thoughts, or self-inconsistent. However, over half of the reviewers did point out a few annotation they considered problematic. We discussed with the reviewers regarding those issues and revised them such that all the annotations are acceptable for the reviewer. The result of Q4 shows that reviewers tend to say our annotations are oversegmented. This however fits our needs because doing this provides extra flexibility to the dataset; researchers who are interested in longer repeated patterns can simply merge our annotations. On the other hand, it is hard to retrieve short motivic patterns from undersegmented annotation.

Algorithm 1 Find Common Structure

1: **function** COMMON STRUCTURE($\mathbf{D}, \Delta t$)
2: $\quad \mathbf{S} \leftarrow \emptyset$
3: $\quad$ **for** $i \leftarrow 1$ to $N-1$ **do**
4: $\quad\quad \mathbf{C}_i \leftarrow \emptyset$
5: $\quad\quad$ **for** $j \leftarrow i+1$ to N **do**
6: $\quad\quad\quad$ **if** $\mathrm{o}_j - \mathrm{o}_i < \Delta t$ **then**
7: $\quad\quad\quad\quad$ add $\mathbf{d}_j$ to $\mathbf{C}_i$
8: $\quad\quad\quad$ **end if**
9: $\quad\quad$ **end for**
10: $\quad\quad \mathbf{S}_i \leftarrow \{\mathbf{d}_j - \mathbf{d}_i,\ \mathbf{d}_j \in \mathbf{C}_i\}$
11: $\quad\quad$ add $\mathbf{S}_i$ to $\mathbf{S}$
12: $\quad$ **end for**
13:
14: $\quad \mathbf{M} \leftarrow \emptyset$
15: $\quad$ **for** $i \leftarrow 1$ to $N-1$ **do**
16: $\quad\quad \hat{\mathbf{S}} \leftarrow \emptyset$
17: $\quad\quad$ **for** $j \leftarrow i+1$ to N **do**
18: $\quad\quad\quad$ add $\{\mathbf{S}_i \cap \mathbf{S}_j\}$ to $\hat{\mathbf{S}}$
19: $\quad\quad$ **end for**
20: $\quad\quad$ add MOST_COMMON($\hat{\mathbf{S}}$) to $\mathbf{M}$
21: $\quad$ **end for**
22: $\quad$ **return** $\mathbf{M}$
23: **end function**

4. MOTIF DISCOVERY

4.1 Algorithm

We regard a motif as a short pattern recurring with little change in its structure. In other words, the relative positions of the notes in a motif will be almost fixed. We therefore find motifs by detecting *common structures* in short musical segments. The idea of the proposed algorithm is presented in Figure 4 and Algorithm 1. Formally, let Δt denote a threshold of time interval, and $\mathbf{D} := \{\mathbf{d}_i\}_{i=1}^{N}$ a musical piece composed of N notes sorted in ascending order, with $\mathbf{d}_i = (\mathrm{o}_i, \mathrm{p}_i)$ being a two-dimensional vector indicating the onset and pitch number of the ith note. For $\mathbf{d}_i$, we first aggregate its context $\mathbf{C}_i$ and create a segment $\mathbf{S}_i$. The derived segments are then compared pairwisely to obtain common structures. By representing a segment as a set of *vectors* (see Figure 4 and Line 3–12 in Algorithm 1), the common structure of any two segments (i.e., the blue arrows in Figure 4) can be obtained by collecting vectors which exist in both segments (Line 18).

As the pairwise comparisons between segments (Line 15-19) will result in various types of common structures, we retrieve a representative pattern and all its occurrences by finding the "most common" structure (i.e., the com-

Algorithm	P_{est}	R_{est}	F_{est}	P_{occ}	R_{occ}	F_{occ}	P_{thr}	R_{thr}	F_{thr}	Runtime
SIATEC	0.1804	0.6444	0.2803	0.2102	0.2771	**0.2235**	0.0408	**0.2994**	0.0713	**28.5082**
COSIATEC	0.2118	0.4557	0.2863	**0.2769**	0.1282	0.1548	0.0489	0.1601	0.0743	208.4119
SIATECCompress	0.2136	0.4326	0.2835	0.1430	0.1121	0.1103	0.0579	0.1703	0.0856	636.6930
Proposed	**0.5709**	**0.8339**	**0.6733**	0.1491	**0.4174**	0.2002	**0.1222**	0.2644	**0.1646**	119.5330

(a) Motif discovery on the proposed dataset

Algorithm	P_{est}	R_{est}	F_{est}	P_{occ}	R_{occ}	F_{occ}	P_{thr}	R_{thr}	F_{thr}	Runtime
SIATEC	0.1238	0.4630	0.1920	**0.5248**	0.3970	**0.4437**	0.0706	**0.4006**	0.1176	**1.5099**
COSIATEC	0.1140	0.2530	0.1491	0.1305	0.0870	0.1044	0.0740	0.2042	0.1027	6.0167
SIATECCompress	0.1807	0.2849	0.2181	0.1778	0.0889	0.1185	**0.1117**	0.2202	0.1470	34.6371
Proposed	**0.2649**	**0.5002**	**0.3406**	0.4208	**0.5105**	0.3948	0.1096	0.3003	**0.1561**	4.0546

(b) Repeated pattern discovery on the JKU-PDD dataset

Table 2: Evaluation of pattern discovery algorithms. The subscripts *est*, *occ*, and *thr* indicate the *establishment*, *occurrence*, and *three-layer* measurements, respectively. The averaged runtime is in minutes.

mon structure that occurs the most times) in $\hat{\mathbf{S}}$ with the MOST_COMMON operation (Line 20). Finally, motifs are acquired by filtering out non-motivic patterns in $\mathbf{M}$ heuristically. In this work, we set $\Delta t = 12$ crotchet beats.

The proposed algorithm differs from the SIA family in two aspects. First, the SIA family aggregates notes of a pattern by detecting equal *translations* among notes, while our algorithm finds patterns by identifying common structures, or *contextual relationships*, among small segments. Second, the SIA family computes maximal translatable patterns (MTP) and subsequently find their occurrences, whereas our algorithm establishes a small pattern and all its occurrences at the same time. Our approach is promising in that the contextual comparisons between segments help identify motifs which are small and recurring. The code of the proposed algorithm is available at `https://github.com/Tsung-Ping/motif_discovery`.

4.2 Evaluation

We evaluate the motif discovery algorithm on the proposed dataset (with an averaged number of 3937 notes per piece) as well as the JKU-PDD dataset (1284 notes in average) [9] using standard metrics for pattern discovery. The *establishment measurement* (est) shows the capability of an algorithm to recognize patterns rather than to find all occurrences of a pattern. The *occurrence measurement* (occ), on the contrary, emphasizes the ability to find all occurrences of a pattern. The *three-layer measurement* (thr) is a comprehensive evaluation combining aspects of both the establishment and occurrence measurements. Each of the three measurements are specified in terms of precision, recall, and F1 score. [3] The *averaged runtime* on each dataset will also be measured to give a rough sketch of the time complexity. We compare the proposed algorithm with three methods from the SIA family, i.e., SIATEC [20], COSIATEC [21], and SIATECCompress [21]. [4] All algorithms were implemented in Python programming language.

The evaluation results are summarized in Table 2. Generally, our algorithm performs consistently across datasets despite that the two datasets are composed of distinct types of musical patterns, i.e., *motivic* (the proposed) versus *thematic* (the JKU-PDD), which differ with each other mainly in the size. Our algorithm is superior to the baselines in all the three establishment measures, indicating that our method can identify more existences of the ground-truth patterns than the other algorithms. Besides, our algorithm is competent in the other two measurements, with at least one best performance in each measurement. Specifically, the R_{occ} measure shows that the patterns retrieved by our algorithm are more complete (i.e., discovering more occurrences of a pattern) with respect to the ground-truth patterns, and the F_{thr} measure suggests that our method has better capability to recognize salient patterns in music, especially the motivic ones. Finally, the runtime measurement indicates that our algorithm can achieve a better performance on the pattern discovery tasks at a moderate computational cost, which is 4.2 (resp. 2.7) times slower than the SIATEC on the proposed (reps. JKU-PDD) dataset.

5. CONCLUSION

We have demonstrated a dataset for repeated pattern discovery of polyphonic symbolic data and a motif discovery algorithm. Our data annotation clearly demonstrates the hierarchical structure of music. The proposed motif discovery algorithm has been shown outperforming the baseline methods on various repeated pattern discovery problems. These findings suggest a direction for developing repeated pattern discovery algorithms, and also evoke further investigation on music structure analysis, novelty analysis, and repeated pattern discovery algorithms.

[3] For more detailed definitions of the three evaluation measurements, refer to `https://www.music-ir.org/mirex/wiki/2017:Discovery_of_Repeated_Themes_\%26_Sections`.

[4] For the three baseline algorithms, we use the implementations available at `https://github.com/wsgan001/repeated_pattern_discovery`.

6. ACKNOWLEDGEMENT

The contribution of each author is as follows. Yo-Wei Hsiao performed data annotation and compiled the whole dataset. Tzu-Yun Hung performed data annotation and the data review process. Tsung-Ping Chen developed the motif discovery algorithm. Lastly, Li Su contributed in project supervision and paper writing. Also, the authors would like to thank Li-Rong Huang, Hsing-Chen Lin, Yu-Fang Hsu, Po-Hsuan Huang, Joseph-On-King Lau, Pei-Ling Kuo, and Chia-Han Li from the Department of Music, National Taiwan Normal University, for their efforts in reviewing and digitizing our data annotation.

7. REFERENCES

[1] D. Meredith, Ed., *Computational music analysis*. Springer Cham, 2016.

[2] P. Boot, A. Volk, and W. B. de Haas, "Evaluating the role of repeated patterns in folk song classification and compression," *Journal of New Music Research*, vol. 45, no. 3, pp. 223–238, 2016.

[3] C. Louboutin and D. Meredith, "Using general-purpose compression algorithms for music analysis," *Journal of New Music Research*, vol. 45, no. 1, pp. 1–16, 2016.

[4] 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*, March 2022.

[5] Z. Hu, X. Ma, Y. Liu, G. Chen, and Y. Liu, "The beauty of repetition in machine composition scenarios," in *Proceedings of the 30th ACM International Conference on Multimedia*, Lisboa, Portugala, 2022, pp. 1223–1231.

[6] D. Meredith, "RECURSIA-RRT: Recursive translatable point-set pattern discovery with removal of redundant translators," in *International Workshops of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD) 2019*, Würzburg, Germany, 2019, pp. 485–493.

[7] O. Björklund, "Siatec-c: Computationally efficient repeated pattern discovery in polyphonic music," in *Proceedings of the 23rd International Society for Music Information Retrieval Conference (ISMIR)*, Bengaluru, India, 2022, pp. 59–66.

[8] T.-P. Chen and L. Su, "Functional harmony recognition of symbolic music data with multi-task recurrent neural networks," in *Proceedings of the 19th International Society for Music Information Retrieval Conference (ISMIR)*, Paris, France, 2018, pp. 90–97.

[9] T. Collins. 2013:Discovery of Repeated Themes & Sections. [Online]. Available: https://www.music-ir.org/mirex/wiki/2013:Discovery_of_Repeated_Themes_%26_Sections

[10] D. Conklin and C. Anagnostopoulou, "Representation and discovery of multiple viewpoint patterns," in *Proceedings of the 2001 International Computer Music Conference (ICMC)*, Havana, Cuba, 2001, pp. 479–485.

[11] 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, December 2021.

[12] M. Krause, F. Zalkow, J. Zalkow, C. Weiß, and M. Müller, "Classifying leitmotifs in recordings of operas by Richard Wagner," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Montréal, Canada, 2020, pp. 473–480.

[13] P. van Kranenburg, B. Janssen, and A. Volk, "The meertens tune collections: The annotated corpus (mtc-ann) versions 1.1 and 2.0.1," *Meertens Online Reports*, vol. 2016, no. 1, 2016.

[14] C. Finkensiep, K. Déguernel, M. Neuwirth, and M. Rohrmeier, "Voice-leading schema recognition using rhythm and pitch features," in *Proceedings of the 21st International Society for Music Information Retrieval Conference (ISMIR)*, Montreal, Canada, 2020, pp. 520–526.

[15] 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 on Computer Music Multidisciplinary Research (CMMR)*, Marseille, France, 2014, pp. 277–297.

[16] K. Frieler, F. Höger, M. Pfleiderer, and S. Dixon, "Two web applications for exploring melodic patterns in jazz solos," in *Proceedings of the 19th International Society for Music Information Retrieval Conference (ISMIR)*, Paris, France, 2018, pp. 777–783.

[17] D. Huron. Theme finder. [Online]. Available: http://www.themefinder.org/

[18] J.-L. Hsu, A. L. Chen, and C.-C. Liu, "Efficient repeating pattern finding in music databases," in *Proceedings of the 7th international conference on Information and knowledge management*, Maryland, USA, 1998, pp. 281–288.

[19] E. Cambouropoulos, M. Crochemore, C. Iliopoulos, L. Mouchard, and Y. Pinzon, "Algorithms for computing approximate repetitions in musical sequences," *International Journal of Computer Mathematics*, vol. 79, no. 11, pp. 1135–1148, 2002.

[20] D. Meredith, K. Lemström, and G. A. Wiggins, "Algorithms for discovering repeated patterns in multidimensional representations of polyphonic music," *Journal of New Music Research*, vol. 31, no. 4, pp. 321–345, 2002.

[21] D. Meredith, "COSIATEC and SIATECCompress: Pattern discovery by geometric compression," in *Music Information Retrieval Evaluation eXchange (MIREX)*, Curitiba, Brazil, 2013.

[22] ——, "Point-set algorithms for pattern discovery and pattern matching in music," in *Dagstuhl Seminar Proceedings on Content-Based Retrieval*, Schloss Dagstuhl, Germany, 2006.

[23] G. Velarde, D. Meredith, and T. Weyde, "A wavelet-based approach to pattern discovery in melodies," D. Meredith, Ed. Springer Cham, 2016, pp. 303–333.

[24] T. Nuttall, G. Plaja, L. Pearson, and X. Serra, "The matrix profile for motif discovery in audio-an example application in carnatic music," in *Proceedings of the 15th International Symposium on Computer Music Multidisciplinary Research (CMMR)*, Online, 2021, pp. 109–118.

[25] M. Krause, F. Zalkow, J. Zalkow, C. Weiß, and M. Müller, "Classifying leitmotifs in recordings of operas by richard wagner," in *Proceedings of the 21st International Society for Music Information Retrieval Conference (ISMIR)*, Montreal, Canada, 2020, pp. 473–480.

[26] I. Ren, A. Volk, W. Swierstra, and R. C. Veltkamp, "A computational evaluation of musical pattern discovery algorithms," *arXiv preprint arXiv:2010.12325*, 2020.

[27] C.-C. M. Yeh, Y. Zhu, L. Ulanova, N. Begum, Y. Ding, H. A. Dau, D. F. Silva, A. Mueen, and E. Keogh, "Matrix profile i: all pairs similarity joins for time series: a unifying view that includes motifs, discords and shapelets," in *2016 IEEE 16th international conference on data mining (ICDM)*, Barcelona, Spain, 2016, pp. 1317–1322.

[28] D. Meredith, "Computing pitch names in tonal music: a comparative analysis of pitch spelling algorithms," Ph.D. dissertation, St. Anne's College, University of Oxford, 2007.

WEAKLY SUPERVISED MULTI-PITCH ESTIMATION USING CROSS-VERSION ALIGNMENT

Michael Krause, Sebastian Strahl, Meinard Müller
International Audio Laboratories Erlangen, Germany
{michael.krause,sebastian.strahl,meinard.mueller}@audiolabs-erlangen.de

ABSTRACT

Multi-pitch estimation (MPE), the task of detecting active pitches within a polyphonic music recording, has garnered significant research interest in recent years. Most state-of-the-art approaches for MPE are based on deep networks trained using pitch annotations as targets. The success of current methods is therefore limited by the difficulty of obtaining large amounts of accurate annotations. In this paper, we propose a novel technique for learning MPE without any pitch annotations at all. Our approach exploits multiple recorded versions of a musical piece as surrogate targets. Given one version of a piece as input, we train a network to minimize the distance between its output and time–frequency representations of other versions of that piece. Since all versions are based on the same musical score, we hypothesize that the learned output corresponds to pitch estimates. To further ensure that this hypothesis holds, we incorporate domain knowledge about overtones and noise levels into the network. Overall, our method replaces strong pitch annotations with weaker and easier-to-obtain cross-version targets. In our experiments, we show that our proposed approach yields viable multi-pitch estimates and outperforms two baselines.

1. INTRODUCTION

Music transcription, i. e., converting music audio recordings into score representations, is a fundamental task in music information retrieval (MIR). As a subtask of transcription, one may estimate the pitches active at different points in time throughout a recording of polyphonic music, yielding a piano roll representation (without considering instrumentation, note values, or other score-based information). This goal is commonly referred to as multi-pitch estimation (MPE). Recent years have seen significant advances in MPE systems, mainly due to the use of deep learning models [1–6]. These models are typically trained with large amounts of aligned pitch annotations as targets, see also Figure 1a. Creating such annotations may involve an enormous effort. In particular, manually annotating pitch activity in every frame of an audio recording would be prohibitively time consuming. Many datasets are thus annotated using semi-automatic methods like score–audio synchronization (e. g., [7]), which introduces annotation errors. Because of this, systems that can learn pitch estimation without large amounts of pitch annotations are highly desirable.

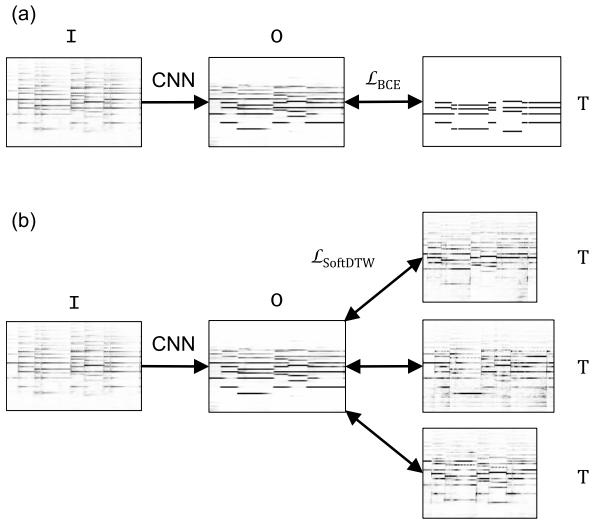


Figure 1: Systems for multi-pitch estimation are typically trained using pitch annotations (a), which are cumbersome to create. In this work, we propose to use different versions of a piece as surrogate targets (b), which are much easier to obtain. In both scenarios, a network input (I) is passed through convolutional layers, producing an output (O), which is compared to one or several targets (T) using some loss function ($\mathcal{L}$).

In this paper, we propose a novel approach for learning MPE without pitch annotations. As our key idea, we use different versions (i. e., recorded performances) of a musical piece as surrogate targets. To this end, we leverage cross-version music datasets, which contain several versions per piece. Such datasets are especially common for Western classical music, where the same compositions are regularly performed by different musicians. Each version exhibits unique timing, artistic expression, and varying acoustic conditions. All versions, however, are based on the same musical score and thus contain the same combinations of pitches. We therefore hypothesize that a deep network may produce pitch estimates by learning the commonalities between different versions of a piece.

© M. Krause, S. Strahl, and M. Müller. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).
Attribution: M. Krause, S. Strahl, and M. Müller, "Weakly Supervised Multi-Pitch Estimation Using Cross-Version Alignment", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

In our approach, we train a deep network that takes a time–frequency representation of one version as input, and whose output minimizes a certain distance to time–frequency representations of other versions. This core idea is illustrated in Figure 1b. Since versions vary in length and the timing of pitch events may be different, we require a distance measure that temporally aligns the network output to the representations of other versions. To do so within a deep learning setting, we use a differentiable variant of dynamic time warping called SoftDTW [8]. Apart from the fundamental frequencies of pitches played, all recorded versions of a piece contain overtone structures and ambient noise. To increase the validity of our hypothesis and to encourage the network to capture nothing but pitches, we incorporate knowledge about overtones and noise using additional fixed processing blocks.

Overall, our proposed approach replaces the need for strong pitch annotations (which are frame-wise, binary, and difficult-to-obtain) with weaker cross-version targets (not temporally aligned, real-valued, and easy-to-obtain).

In summary, we make the following contributions: We propose a novel approach for weakly supervised MPE that does not require pitch annotations, based on the hypothesis that pitch estimation can be learned from multiple versions. We further propose to incorporate extra layers for simulating overtones and noise levels to ensure that our hypothesis holds. Finally, as a proof of concept, we show qualitatively and quantitatively that our approach can be used for MPE and outperforms two baselines. To aid reproducibility, we release code and trained models for our approach. [1]

The remainder of this paper is structured as follows: In Section 2, we discuss related work on pitch estimation. In Section 3, we describe our proposed approach. Section 4 covers the experimental setup, while Section 5 contains our results. Section 6 concludes the paper with an overview of possible directions for future work.

2. RELATED WORK ON MULTI-PITCH ESTIMATION

The majority of work on MPE and music transcription in general has focused on supervised training schemes, where a dataset of music recordings with aligned pitch annotations is given. Most recent papers utilize deep learning models that are trained with pitch targets using standard cross-entropy loss functions [1–5]. Often, these works focus on piano music, where annotations can be obtained using MIDI recording technology built into certain types of pianos [9]. We refer to [6] for an overview of music transcription research.

Some works have explored pitch estimation from data without aligned pitch annotations. Weiß and Peeters [10] proposed to utilize weakly aligned annotations, where there may be temporal deviations between recorded performance and annotations. This scenario is also explored in [11]. However, in both cases, pitch annotations are required for the entire training dataset. Gfeller et al. [12] introduced a self-supervised approach for pitch estimation, where a network learns to predict the relative differences between pitch-shifted, monophonic recordings. Their approach requires only a small amount of data with pitch annotations, but does not deal with polyphonic scenarios. Berg-Kirkpatrick et al. [13] describe a system for MPE on piano recordings that does not use pitch annotations. Their approach solves an optimization problem, with constraints motivated by the sound production process in pianos. In contrast, the method we propose in this paper utilizes several versions of a musical piece and could be used for recordings with arbitrary instruments.

3. PROPOSED METHOD

We now describe our proposed approach for learning MPE using cross-version alignment. Here, we assume that we have multiple corresponding recorded versions for each musical piece in the training set. Let us denote the set of all corresponding versions for one piece by $\mathcal{V} = \{V_1, V_2, \dots\}$. Furthermore, given a version $V \in \mathcal{V}$, we write $\text{InputRep}(V)$ for the audio representation of V that our network takes as input. [1]

Given an input $\mathtt{I} = \text{InputRep}(V)$, we formulate MPE as the problem of producing a binary piano roll $\tilde{\mathtt{M}} \in \{0,1\}^{B \times N}$ that matches the pitch annotations $\mathtt{A} \in \{0,1\}^{B \times N}$ for that input. Here, B denotes the number of pitch bins, while N is the number of time frames in the input. In the supervised case, deep networks for MPE produce a real-valued output $\mathtt{O} \in [0,1]^{B \times N}$ that is optimized using the binary cross-entropy loss $\mathcal{L}_{\text{BCE}}$ with $\mathtt{T} = \mathtt{A}$ as targets (where the loss is averaged over all time–pitch bins). The final pitch predictions $\tilde{\mathtt{M}}$ are obtained from $\mathtt{O}$ by applying a threshold τ. This threshold is often set to a fixed value (e. g., $\tau = 0.4$ in [7]) or optimized on a validation dataset [14]. This supervised approach to MPE, which crucially relies on the aligned pitch annotations $\mathtt{A}$, is illustrated in Figure 1a. In the following, we will refer to it with the shorthand $\mathtt{Sup}$.

Our proposed approach, illustrated in Figure 1b, also takes an input representation $\mathtt{I} = \text{InputRep}(V)$ for some version $V \in \mathcal{V}$. As before, our network yields a real-valued output $\mathtt{O} \in [0,1]^{B \times N}$. However, rather than using pitch annotations $\mathtt{A}$, we utilize a surrogate target $\mathtt{T} = \text{TargetRep}(V')$ based on another version $V' \in \mathcal{V} \setminus \{V\}$. We choose a time–frequency representation $\text{TargetRep}(V') \in [0,1]^{B \times N'}$ as target that is normalized in the range $[0,1]$ and has the same number of bins B as $\mathtt{O}$, but a potentially different number of time frames N', due to the temporal differences between versions. [1] As explained in the introduction, $\mathtt{T}$ contains the same combinations of pitches as $\mathtt{I}$. [2] Intuitively, if $\mathtt{O}$ is close to the target representations of all versions $\mathcal{V} \setminus \{V\}$, we hypothesize that $\mathtt{O}$ must correspond to pitch estimates for $\mathtt{I}$. We

[1] https://www.audiolabs-erlangen.de/resources/MIR/2023-ISMIR-WeaklySupervisedMPE

[1] Details of InputRep and TargetRep are provided in Section 4.

[2] Here, we assume that there are no structural differences between versions, i. e., performers do not deviate from the score. Versions performed in different keys can be handled through pitch shifting, see Section 4.

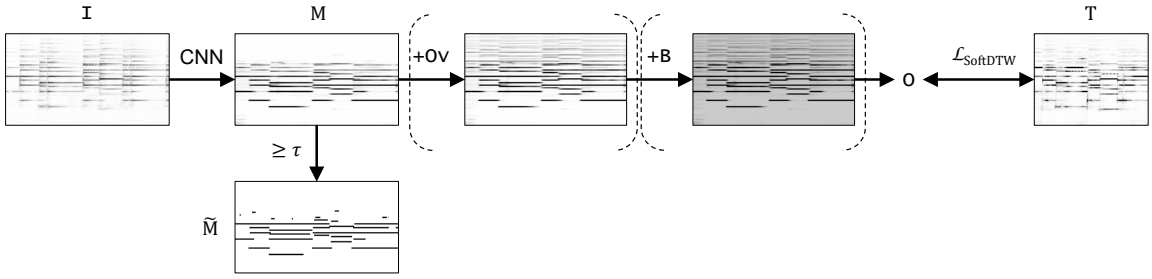


Figure 2: Detailed overview of the proposed cross-version alignment (CVA) method, see also Figure 1b. Before applying the alignment loss ($\mathcal{L}_{\text{SoftDTW}}$), the intermediate output of the network (M) is optionally extended using a simple overtone model (+Ov) and a bias value (+B) to address background noise. The final output O can thus arise from different configurations (e. g., O = M, O = M+Ov, . . .). Importantly, the MPE output of the system ($\tilde{\text{M}}$) is computed based on the intermediate representation M, rather than the output O.

refer to our proposed approach with the shorthand CVA (for "cross-version alignment").

Note that we cannot directly apply a loss on time–pitch bins here (as in the supervised case), since O and T are not temporally aligned. For this reason, we use the differentiable alignment loss $\mathcal{L}_{\text{SoftDTW}}$ in our approach, see Section 3.1. Furthermore, our hypothesis may fail to apply, since recorded versions of a piece contain overtone structures and background noise in addition to the pitches played. We thus extend our approach to account for these properties of music recordings in Section 3.2.

3.1 Differentiable Alignment

In order to perform temporal alignment between O and a target representation T in a differentiable fashion, we use the SoftDTW loss [8]. SoftDTW is a differentiable approximation of the classical dynamic time warping algorithm that is often used to align music sequences [15]. SoftDTW has originally been introduced for one-dimensional time series but has also been adopted for computer vision tasks like action recognition in video recordings [16, 17]. Within MIR, SoftDTW has previously been used in the context of music synchronization [18] and MPE [11]. In [11], the authors showed that SoftDTW can be used to replace strongly aligned (i. e., frame-wise) pitch annotations with weakly aligned pitches without a major impact on MPE performance. Nevertheless, their approach requires pitch annotations for training.

In our case, we crucially rely on the ability of SoftDTW to align real-valued sequences such as time–frequency representations of audio. In contrast, a commonly used alternative loss function called connectionist temporal classification (CTC) can only handle discrete target sequences. To compute $\mathcal{L}_{\text{SoftDTW}}$, one needs to choose a local cost function (for comparing individual frames of the time–frequency representations) and set a temperature hyperparameter called γ (which determines the approximation quality of SoftDTW). Here, we use the cosine distance for comparing frames, which exhibited high training stability in our experiments. We further set $\gamma = 0.1$, which corresponds to a good approximation of DTW.

As a drawback, the time and space complexity of SoftDTW is quadratic in the lengths of the input sequences. We thus train on short input excerpts (see Section 4).

3.2 Overtone and Noise Model

Aside from differentiable alignments, our proposed approach utilizes fixed processing layers that simulate overtone structures and background noise. In this way, our method follows the analysis-by-synthesis paradigm [19], where one estimates parameters from an audio recording (pitches, in our case) by re-synthesizing the input. Choi and Cho [20] utilized this idea for unsupervised drum transcription. Their network consists of a transcription stage and a fixed sample-based drum synthesizer. The transcription network is trained by minimizing a reconstruction loss on the synthesizer output. In recent years, such systems have become more popular due to the release of the differentiable digital signal processing (DDSP) library [21], which has been used, e. g., in the context of unsupervised monophonic pitch estimation [22]. In contrast to these works, our proposed approach utilizes cross-version data.

A full overview of our CVA approach is given in Figure 2. We explicitly add overtones (denoted by +Ov) and background noise (+B) to an intermediate output M of our network via dedicated layers. In this way, the network may learn a sparser and more piano roll-like representation M, since overtones and noise are added afterwards. Crucially, the final MPE results $\tilde{\text{M}}$ are obtained from M, before overtones and noise are applied. The output O, used for alignment with the cross-version targets, depends on the model configuration used. For example, O = M+Ov+B if all modules are used, O = M+Ov if only overtones are added, etc. In the basic system without extensions, O = M.

Here, we opt for very simple overtone and noise models that serve to indicate the potential of our core idea. We estimate the relative amplitudes of different harmonics from a small internal dataset of single-note piano recordings. The resulting estimates, used for our overtone model, are illustrated in Figure 3. We keep these values fixed for all subsequent experiments. To apply this fixed overtone model within our network in a differentiable fashion, we sum up pitch-shifted versions of M. For each harmonic h, we shift M along the vertical axis by a number of semitones corre-

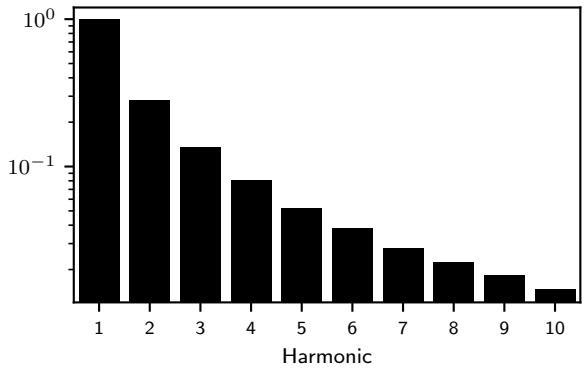


Figure 3: Amplitudes for the overtone model (+Ov) employed in our proposed approach.

sponding to h (e. g., 12 semitones for $h = 2$). We then weight the shifted representation with the amplitude estimated for h (see Figure 3). The final output is obtained by summing the resulting representations for all h.[3] To address the overall noise level in the target T, we add a fixed bias term of $\delta = 0.2$ after applying the overtone model. As a result of this additional processing, we may obtain outputs larger than 1. We therefore clip all values outside the interval $[0, 1]$ (corresponding to the value range of the target representations T) to get the final output O.

4. EXPERIMENTAL SETUP

4.1 Model, Representations, and Training

In this work, we focus on demonstrating the potential of our cross-version approach compared to traditional, fully supervised training for MPE. Thus, we do not propose complicated network architectures that require extensive tuning. Instead, we use a relatively small convolutional neural network for extracting the representation M from I. For InputRep and TargetRep, we use time–frequency representations based on the constant-Q transform (CQT), which provides a frequency axis corresponding to semitones. Note that we cannot train on entire (several minutes long) recordings in a single step. Instead, our training batches contain short input excerpts and we use state-of-the-art music synchronization techniques [23] to find the corresponding sections in other versions.

Concretely, we use the network architecture, input representation, and training setup from [10] (we refer to their paper for details). Their network consists of five convolutional layers with musically motivated kernel shapes and roughly 50 000 learnable parameters. The network takes a magnitude harmonic CQT (HCQT [24]) of an audio excerpt as InputRep, containing $N = 500$ frames computed with a hop size of 512 from waveforms at 22 050 Hz (i. e., an excerpt of 11.6 seconds length). The network produces outputs M of the same length, with a pitch axis containing $B = 72$ bins (corresponding to the semitones from C1 to B6). The final layer of the network contains a sigmoid activation, such that all values in M are restricted to the interval $[0, 1]$. For TargetRep, we use magnitude CQTs where the center frequencies of different bins correspond to the same $B = 72$ semitones. Column-wise max-normalization is applied on T, such that the target values are also in $[0, 1]$.

We train our network by minimizing the SoftDTW loss over all training excerpts until the validation loss has stopped improving for 12 epochs. In each training step, we compute the loss on a batch of 16 inputs. Each input excerpt is based on some version $V \in \mathcal{V}$ and aligned to the corresponding excerpt in one randomly selected target version $V' \in \mathcal{V} \setminus \{V\}$. We use the Adam optimizer with a learning rate of 0.001, which is reduced whenever the validation loss has not improved for three epochs. Finally, we employ an efficient CUDA implementation of the SoftDTW recursions by Maghumi et al. [25].[4]

4.2 Dataset and Split

To train our cross-version approach, we require a dataset containing multiple versions per piece. For testing, we additionally require aligned pitch annotations for the recordings. We opt for using the Schubert Winterreise Dataset (SWD, [26]) for training, which contains nine versions of the 24 songs in the cycle "Winterreise" composed by Franz Schubert (in total, roughly 11 h of audio). Each song constitutes one unique musical piece. The recordings consist of a tenor or baritone singer accompanied by piano. There are no structural differences between versions. Thus, all recordings for a piece contain the same combinations of pitches up to transposition (a global pitch shift), since some musicians chose to perform some songs in different keys. When training our CVA approach, we ensure that input and target version are in the same key by appropriately shifting the target CQT representation according to the key annotations given in the dataset.

We train and evaluate our model using a challenging split where the train and test sets contain both different versions and different songs. We choose songs 1–13 for training, 14–16 for validation, and 17–24 for testing. Furthermore, versions HU33 and SC06 are used for testing, while the remaining seven versions are used for training and validation. Such a split is also referred to as a "neither split", since neither the same versions nor songs appear during training and testing [27]. This split avoids over-optimistic evaluation due to confounders such as the "album effect" [28].

4.3 Baselines

Aside from the supervised baseline Sup, which is trained using strong pitch annotations, we compare our proposed CVA approach to two additional baselines. With these, we aim to evaluate our hypothesis that cross-version targets are useful for learning MPE-like representations (see

[3] Equivalently, the overtone model can be understood as a frame-wise convolution in pitch direction, with a kernel based on the amplitudes in Figure 3.

[4] Note that, within one batch, the targets T may have different lengths. In order to benefit from parallelization across the batch dimension, we therefore rescale the targets T to a common length $N' = 500$ (a trick referred to as W4 in [11]). This did not affect results negatively in early experiments. Note that rescaling is not equivalent to temporally aligning inputs and targets.

Scenario	CS	AP	$\tau = 0.4$		$\tau = \tau^*$	
			F	Acc.	F	Acc.
CQT	0.585	0.410	0.443	0.287	0.450	0.292
AE	0.588	0.500	0.336	0.203	0.511	0.345
CVA	0.632	0.589	0.585	0.416	0.592	0.423
CVA+Ov	0.664	0.639	0.553	0.384	0.623	0.455
CVA+B	0.633	0.563	0.560	0.392	0.592	0.424
CVA+Ov+B	0.682	0.646	0.625	0.458	0.627	0.460
Sup	0.748	0.753	0.700	0.543	0.703	0.546

Table 1: Results for multi-pitch estimation on the Schubert Winterreise Dataset for the baselines and different configurations of our proposed approach.

Section 3). For the CQT baseline, we take the target representations of our test recordings (which are normalized to have values in the range $[0, 1]$) and obtain multi-pitch estimates by directly thresholding these magnitude CQTs with τ. This learning-free baseline was previously proposed in [10] and, like CVA, does not require pitch annotations. Furthermore, we consider a second baseline that is very similar to CVA but does not utilize cross-version targets. Therefore, for each input excerpt, we choose the same version $V \in \mathcal{V}$ for both I and T. Thus, the network needs to effectively recreate its input, similar to an autoencoder. We refer to this baseline with the shorthand AE. Intuitively, we expect CQT and AE to yield similar results. However, AE allows us to verify that any improvements observed for CVA stem from the cross-version targets and not from the model architecture or training setup. Note that Sup and AE use the same network architecture as CVA.

4.4 Evaluation Metrics

We evaluate the multi-pitch estimates of our proposed approach and all baselines using standard metrics on the test set. For this, we utilize the strongly aligned pitch annotations provided in the test data. As metrics, we use the cosine similarity (CS) between predictions and annotations, averaged over all frames and files in the test set. Furthermore, we compare the average precision (AP, computed as the area under the precision-recall curve), F-measure (F), and the accuracy (Acc.) metric introduced in [29]. For these measures, we average over all pitches. Note that F and Acc. are evaluated on $\tilde{\mathrm{M}}$ and thus depend on the threshold τ, while CS and AP are threshold-free evaluation metrics that directly compare M and A.

5. RESULTS

The main results of our study are summarized in Table 1. Rows correspond to different baselines or configurations of our proposed approach. We write +Ov when adding overtones and +B when including the bias term to account for background noise. Our model including all proposed modules is thus referred to as CVA+Ov+B. Columns contain the evaluation metrics. For the thresholding-based metrics F and Acc., we provide both results based on a fixed threshold ($\tau = 0.4$) and a threshold chosen to optimize F on the validation set ($\tau = \tau^*$).

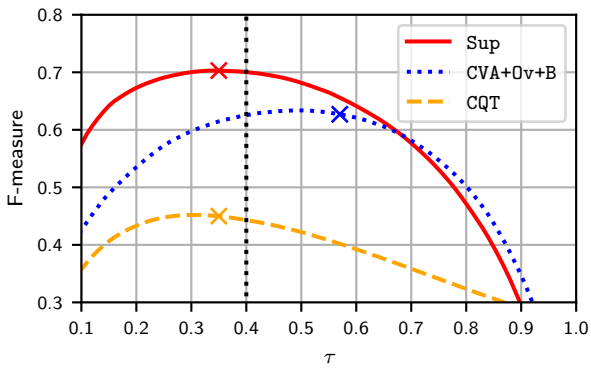


Figure 4: F-measures on the test set for different MPE approaches, depending on the choice of threshold τ. Markers show the optimal threshold τ^* as determined on the validation set.

Our proposed approach CVA outperforms the two baselines CQT and AE across all metrics, demonstrating the effectiveness of using different versions of a piece to capture pitches in M. For example, CS $= 0.632$ for CVA compared to CS $= 0.588$ for AE, and AP $= 0.589$ for CVA compared to AP $= 0.410$ for CQT. Furthermore, our proposed overtone and noise models are effective. By adding overtones (CVA+Ov), we can further increase AP from 0.589 to 0.639. Adding a fixed bias term (CVA+B) does not yield improvements by itself. However, by combining both modules (CVA+Ov+B), we achieve the best results for our approach, further increasing AP to 0.646 and CS to 0.682.

Despite these encouraging results, there remains a gap between the best results for our proposed approach and those for the supervised baseline Sup. We emphasize again that—unlike CVA—Sup requires strong pitch annotations for training.

5.1 Impact of Threshold τ

When using the standard value of $\tau = 0.4$ for thresholding M, our CVA approach also outperforms both baselines in terms of F-measure and accuracy (e. g., F $= 0.553$ for CVA+Ov compared to 0.443 for CQT).

A fixed threshold may be sub-optimal, especially for methods that are not explicitly trained for MPE. When evaluating using the optimized threshold τ^*, we observe increased results for all approaches. CVA and its extensions continue to outperform the two baselines. The F-measure for CVA+Ov, for example, further increases to F $= 0.623$. For that model, the optimal threshold as determined on the validation set is $\tau^* = 0.28$. In this case, our method requires at least a few pitch annotations to determine τ^* and is no longer relying solely on the cross-version targets.

Figure 4 further demonstrates the impact of the parameter τ. F-measures (vertical axis) are shown for different MPE approaches (colored lines), depending on the choice of τ (horizontal axis). Markers indicate τ^*. As shown in this figure, a poor choice of τ may strongly affect test results. Moreover, τ^* as found using the validation set may not always give the highest scores on the test set. For in-

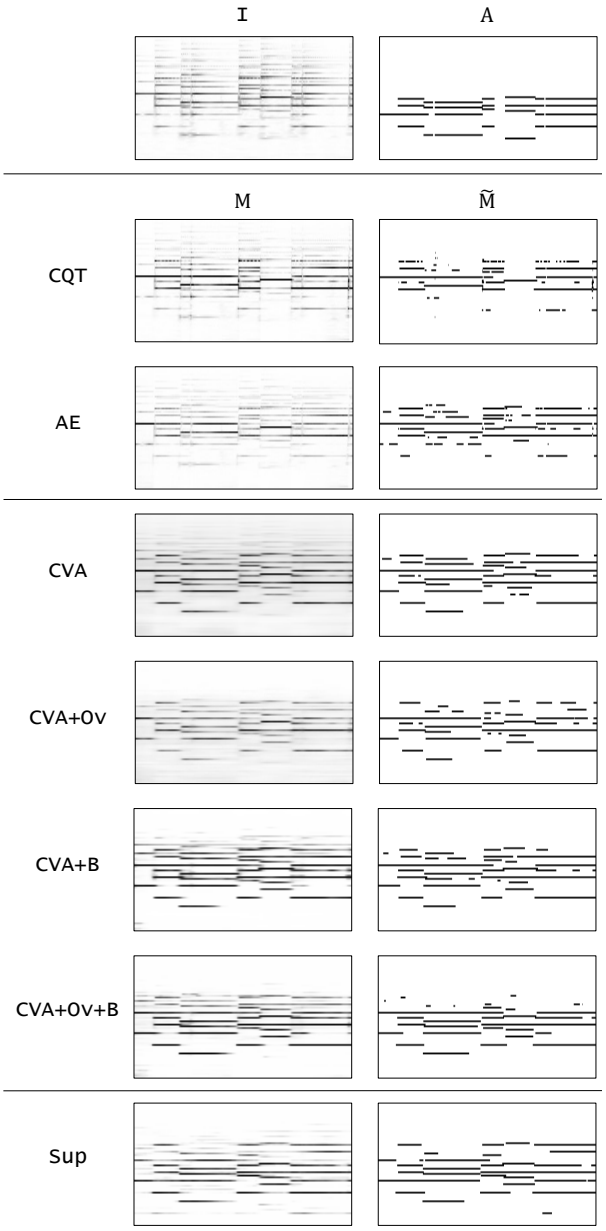


Figure 5: Qualitative results on a test excerpt from SWD.

stance, a choice of $\tau = 0.5$ would yield an even higher F-measure of 0.634 for `CVA+Ov+B`.

5.2 Training Stability

The metrics reported in Table 1 are computed from a single training run per method. When repeating the experiment, results may deviate slightly due to random network initialization, dataset shuffling, or dropout. For `CVA+Ov+B`, we repeat the experiment five times and find low standard deviation σ in results ($\sigma(\text{CS}) = 0.004$, $\sigma(\text{AP}) = 0.009$, $\sigma(\text{F}) = 0.008$, and $\sigma(\text{Acc.}) = 0.009$ for $\tau = \tau^*$).

5.3 Qualitative Results

To complement the quantitative evaluation, we also provide qualitative results on an exemplary excerpt in Figure 5. The first row shows an input excerpt (`I`) and corresponding pitch annotations (`A`), while the remaining rows show multi-pitch estimates before (`M`) and after thresholding ($\tilde{\mathtt{M}}$, computed using $\tau = \tau^*$).

For `CQT` and `AE`, the resulting `M` correspond to the input representation and thus lead to poor multi-pitch estimates.

When training our approach without overtones or noise model (`CVA`), the output representation `M` emphasizes the fundamental frequencies of many of the actual pitches being played. However, `M` also contains a lot of energy from overtone structures and background noise. As a consequence, the resulting $\tilde{\mathtt{M}}$ contains many spurious pitch predictions, especially for higher pitches.

With `+Ov` and `+B`, we see a reduced impact of overtones or background noise in `M`, respectively. In both cases, many erroneous predictions remain after thresholding. By including both modules (`CVA+Ov+B`), we obtain a promising representation that bears visual resemblance to the results for `Sup`. We also observe fewer spurious activations in $\tilde{\mathtt{M}}$ compared to the basic `CVA`. Overall, the proposed extensions are effective in encouraging the model to produce MPE predictions in `M`.

6. CONCLUSION

In this paper, we presented a novel approach for MPE that does not require pitch annotations for training. Instead, our method utilizes multiple versions of the same musical piece as surrogate targets. We train a network that takes a time–frequency representation of one version as input and minimizes an alignment-based distance to time–frequency representations of other versions. We hypothesized that this would result in outputs corresponding to pitch estimates. We further incorporate knowledge about overtones and noise levels into our system to support this hypothesis and improve results. In our experiments, we showed that our approach outperforms two baselines and that our proposed extensions to the model are effective. Overall, our work demonstrates the use of weak cross-version targets to replace strong pitch annotations.

This paper serves as a proof of concept for our core idea, which could be extended in future work. First, better results may be obtained by utilizing larger model architectures and bigger training datasets than in the present study. Here, we also abstained from excessive model and hyperparameter tweaking. In the future, larger and more extensively tuned models may close the gap between fully supervised approaches and the proposed cross-version training. Second, one may extend our approach to align one input excerpt to multiple versions simultaneously within the same training step (rather than choosing one target version at a time). This may further regularize the model output. Finally, future work may explore more elaborate synthesis models that could replace the simplistic overtone and noise models used here. For example, one may incorporate knowledge about the sound production processes of different instruments into the network [13]. In this context, results might also be improved by estimating the synthesis parameters (e. g., amplitudes of the overtone model) from the input recording, rather than using fixed processing steps.

Acknowledgments: This work was supported by the German Research Foundation (DFG MU 2686/7-2, MU 2686/11-2). 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. The authors gratefully acknowledge the compute resources and support provided by the Erlangen Regional Computing Center (RRZE).

7. REFERENCES

[1] 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.

[2] 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.

[3] 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.

[4] K. W. Cheuk, Y. Luo, E. Benetos, and D. Herremans, "Revisiting the onsets and frames model with additive attention," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, Shenzhen, China, 2021.

[5] 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.

[6] 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.

[7] 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.

[8] M. Cuturi and M. Blondel, "Soft-DTW: a differentiable loss function for time-series," in *Proceedings of the International Conference on Machine Learning (ICML)*, Sydney, NSW, Australia, 2017, pp. 894–903.

[9] 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.

[10] C. Weiß and G. Peeters, "Learning multi-pitch estimation from weakly aligned score-audio pairs using a multi-label CTC loss," in *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, USA, 2021, pp. 121–125.

[11] M. Krause, C. Weiß, and M. Müller, "Soft dynamic time warping for multi-pitch estimation and beyond," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Rhodes Island, Greece, 2023.

[12] 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, pp. 1118–1128, 2020.

[13] T. Berg-Kirkpatrick, J. Andreas, and D. Klein, "Unsupervised transcription of piano music," in *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, Montréal, Canada, 2014, pp. 1538–1546.

[14] 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.

[15] M. Müller, *Fundamentals of Music Processing – Using Python and Jupyter Notebooks*, 2nd ed. Springer Verlag, 2021.

[16] I. Hadji, K. G. Derpanis, and A. D. Jepson, "Representation learning via global temporal alignment and cycle-consistency," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Virtual, 2021, pp. 11 068–11 077.

[17] C. Chang, D. Huang, Y. Sui, L. Fei-Fei, and J. C. Niebles, "D3TW: Discriminative differentiable dynamic time warping for weakly supervised action alignment and segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, 2019, pp. 3546–3555.

[18] R. Agrawal, D. Wolff, and S. Dixon, "A convolutional-attentional neural framework for structure-aware performance-score synchronization," *IEEE Signal Processing Letters*, vol. 29, pp. 344–348, 2021.

[19] N. Cleju, M. G. Jafari, and M. D. Plumbley, "Analysis-based sparse reconstruction with synthesis-based solvers," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Kyoto, Japan, 2012, pp. 5401–5404.

[20] K. Choi and K. Cho, "Deep unsupervised drum transcription," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Delft, The Netherlands, 2019, pp. 183–191.

[21] J. Engel, L. Hantrakul, C. Gu, and A. Roberts, "DDSP: Differentiable digital signal processing," in *Proceedings of the International Conference on Learning Representations (ICLR)*, Virtual, 2020.

[22] J. Engel, R. Swavely, L. H. Hantrakul, A. Roberts, and C. Hawthorne, "Self-supervised pitch detection by inverse audio synthesis," in *International Conference on Machine Learning (ICML), Workshop on Self-Supervision in Audio and Speech*, Vienna, Austria, 2020.

[23] 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.

[24] R. M. Bittner, B. McFee, J. Salamon, P. Li, and J. P. Bello, "Deep salience representations for F0 tracking in polyphonic music," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Suzhou, China, 2017, pp. 63–70.

[25] M. Maghoumi, E. M. Taranta, and J. LaViola, "DeepNAG: Deep non-adversarial gesture generation," in *Proceedings of the International Conference on Intelligent User Interfaces (IUI)*, College Station, Texas, USA, 2021, pp. 213–223.

[26] C. Weiß, F. Zalkow, V. Arifi-Müller, M. Müller, H. V. Koops, A. Volk, and H. Grohganz, "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.

[27] C. Weiß, H. Schreiber, and M. Müller, "Local key estimation in music recordings: A case study across songs, versions, and annotators," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2919–2932, 2020.

[28] A. Flexer, "A closer look on artist filters for musical genre classification," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Vienna, Austria, 2007, pp. 341–344.

[29] G. E. Poliner and D. P. Ellis, "A discriminative model for polyphonic piano transcription," *EURASIP Journal on Advances in Signal Processing*, vol. 2007, no. 1, 2007.

THE BATIK-PLAYS-MOZART CORPUS: LINKING PERFORMANCE TO SCORE TO MUSICOLOGICAL ANNOTATIONS

Patricia Hu[1] **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

We present the *Batik-plays-Mozart* Corpus, a piano performance dataset combining professional Mozart piano sonata performances with expert-labelled scores at a note-precise level. The performances originate from a recording by Viennese pianist Roland Batik on a computer-monitored Bösendorfer grand piano, and are available both as MIDI files and audio recordings. They have been precisely aligned, note by note, with a current standard edition of the corresponding scores (the New Mozart Edition) in such a way that they can further be connected to the musicological annotations (harmony, cadences, phrases) on these scores that were recently published by [1].

The result is a high-quality, high-precision corpus mapping scores and musical structure annotations to precise note-level professional performance information. As the first of its kind, it can serve as a valuable resource for studying various facets of expressive performance and their relationship with structural aspects.

In the paper, we outline the curation process of the alignment and conduct two exploratory experiments to demonstrate its usefulness in analyzing expressive performance.

1. INTRODUCTION

Music performance is a complex and nuanced activity that involves the interplay of various expressive features such as timing, dynamics, and articulation. Expressive performance research in music information retrieval (MIR) focuses on modeling expressive aspects of music performance by analyzing how performers use nuances in timing, dynamics, articulation, and other expressive features to convey their musical intentions, with the aim of developing computational models that can analyze, recognize, or synthesize expressive performances [2].

Recent research in this field for Western classical piano has focused on data-driven approaches both for performance generation [3, 4] and data creation in the form of large-scale MIDI performance data transcribed from audio recordings [5, 6]. While such data corpora can be useful for comparative performance analyses and related tasks (e.g., performer identification, performance style transfer), they lack the necessary precision and alignment information (with the underlying musical score) required to precisely map expressive intentions and parameters to underlying score features.

Compared to these large-scale transcribed MIDI datasets, precise MIDI data (as recorded on computer controlled grand pianos such as the Yamaha Disklavier or Boesendorfer SE/Ceus series) along with their corresponding score alignment is somewhat limited in quantity and size [7–9]. The performances in such datasets are typically sourced from advanced piano students or piano competitions, whereas the digital scores are often obtained from open-source, user-curated online libraries such as MuseScore [1].

Regarding the performance-to-score alignment, one would ideally want to have note-by-note correspondence information; unfortunately, in the case of the largest of these datasets [7], score-performance alignments are only given at a rather coarse level of beats. Score annotations conveying structural information such as underlying harmony or phrases are even more scarce.

To address these limitations, we introduce the *Batik-plays-Mozart* dataset [2], in which we provide a set of expert performances of 12 complete Mozart piano sonatas (36 distinct movements) in MIDI format by concert pianist Roland Batik, precisely aligned, at a note-by-note-level, to a standard edition (the New Mozart Edition) of the score, thereby linking the performance information to a previously published dataset [1] of expert annotations of the scores in terms of harmony, cadence, and phrase structure. To the best of our knowledge, this is the first corpus of its kind, combining high quality digital score and structural annotations with expert performances in recorded MIDI format. We report two preliminary experiments to demonstrate the benefits of having precise performance–score–structure annotation alignments.

The remainder of this paper is organised as follows: Section 2 presents a list of comparable expressive performance datasets currently publicly available. Section 3 de-

© P. Hu and G. Widmer. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** P. Hu and G. Widmer, "The Batik-plays-Mozart Corpus: Linking Performance to Score to Musicological Annotations", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

[1] https://musescore.com/sheetmusic
[2] https://github.com/huispaty/batik_plays_mozart

Dataset	Size		Modality		Annotations	
	Pieces	Performances	MIDI	Score	Alignment	Other
ASAP [7]	222	1,068	recorded	MusicXML	beat	time and key signature
Vienna4x22 [8]	4	88	recorded	MusicXML	note	-
CrestMuse PEDB [9]	35	411	recorded	MusicXML	note	phrase
MazurkaBL [10]	44	2,000	-	MusicXML	beat	dynamics, tempo markings
Batik-plays-Mozart	36	36	recorded	MusicXML	note	phrase, harmony, cadence

Table 1. An overview of publicly available comparable piano performance datasets for which precise recorded MIDI data, score-performance alignments and/or musicological annotations are available.

scribes the data origins, the used data formats, and the curation process. Section 4 gives an overview of the dataset, and Section 5 describes two preliminary experiments to demonstrate the benefits of performance–score–structure annotation alignments. Finally, Section 6 concludes the paper with some remarks for future work.

2. RELATED WORK

Several piano performance datasets have been published in the context of expressive performance analysis and performance rendering. While recently published datasets are considerably larger than *Batik-plays-Mozart*, they provide performance recordings solely in the form of MIDI transcribed from audio recordings [5, 6] or do not include a high-quality digital score ground truth [11]. Despite the encouraging results demonstrated by recent transcription models, they often introduce inaccuracies, such as incorrect note fragmentation, missed note onsets, and falsely identified notes [12]. Similarly, certain expressive performance aspects such as (micro-)timing and tempo can only be measured given either a temporal or note-wise score-performance mapping [2]. Nevertheless, these datasets remain useful for various related tasks such as symbolic music generation, music transcription and tagging, or high-level comparative performance analysis.

Table 1 presents an overview of comparable piano performance datasets currently publicly available, for which precise (recorded) MIDI data, score-performance alignments and/or musicological annotations are available. Among these datasets, ASAP [7] stands out as the most extensive one, both in terms of musical pieces and performer range, with 1,068 performances beat-aligned to 222 scores, each annotated with key and time signature. In comparison to ASAP, all other publicly accessible datasets are significantly smaller: The Vienna 4x22 corpus [8] contains 22 different performances for excerpts of four different pieces, each aligned on a note level and provided in MusicXML [3], MIDI and audio format. The CrestMuse PEDB v2.0 [9] provides 35 pieces note-aligned to 411 performances, with scores provided in MusicXML and MIDI and performances in MIDI and WAV. The dataset also contains phrase structure annotations, however, merely in the format of PDF and plain text files, somewhat limiting their (re)usability.

The MazurkaBL dataset [10] consists of a corpus of 44 Chopin Mazurkas with MusicXML scores that have been beat-aligned to 2000 performances. The performances themselves are not provided (neither as MIDI nor as audio); only beat positions and corresponding loudness values are given, along with the positions of tempo/dynamics markings in the score.

[3] https://www.musicxml.com/

3. CURATION PROTOCOL AND FILE FORMATS

3.1 File origins

The MIDI performance files originate from a performance of twelve Mozart piano sonatas by Viennese concert pianist Roland Batik on a computer-controlled Bösendorfer SE290 grand piano, the predecessor of the CEUS model. The Bösendorfer SE series measures each individual keystroke and pedal movement precisely, with onset and offset times being captured at a time resolution of 1.25ms. Hammer velocity values are captured in a proprietary file format, and converted and mapped to the 128 dynamics MIDI values (see [13] for conversion details). The audio recordings corresponding to those MIDI files can be purchased commercially [4].

These MIDI performance data were originally aligned manually, on a note-to-note level, to a symbolic encoding of the score produced by our team [14, 15]. In order to make it possible to link the performance data in an unequivocal way to the musicological score annotations provided in the *Annotated Mozart Sonatas* dataset by Hentschel et al. [1], we decided to replace our score encoding in the alignments entirely by the score notes as given in the their dataset, which link to their annotations directly via absolute temporal score position. The scores in the *Annotated Mozart Sonatas* dataset conform to the New Mozart Edition [5] and are given in MuseScore format, with the harmony, phrase and cadence label annotations provided in tabular format, as tab-separated values (TSV) files.

3.2 The match alignment format

We provide the alignment between the above-mentioned score and performance files in the match file format [16], a file format for symbolic music alignment in a human-understandable textual form. It is structured sequentially, and the alignment information is given at the level of individual notes.

[4] https://www.gramola.at/products/9003643987012

[5] https://dme.mozarteum.at/DME/nma/start.php?l=2

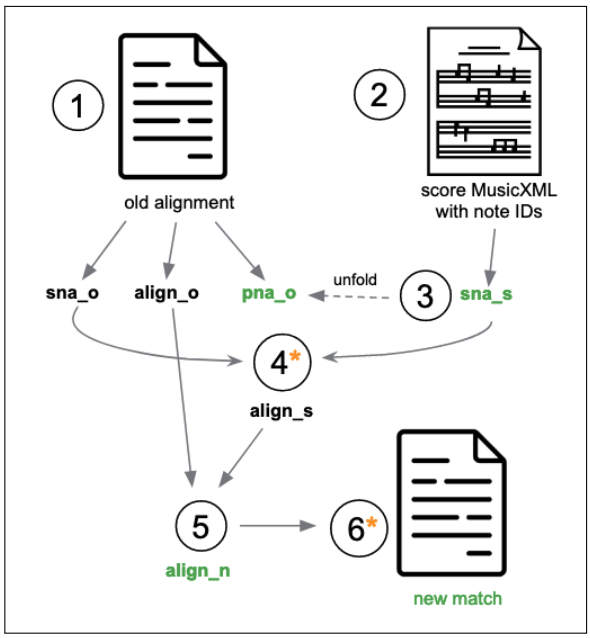


Figure 1. Visual illustration of the alignment process. Each step in the alignment process is numbered according to the textual description in Section 3.3. Steps marked * indicate manual correction / post-processing. Elements highlighted in green are combined in the new alignment match files.

The encoded alignment is complete in the sense that all performance and all score notes are captured. Each performance and each score note is represented with their respective note ID, and their respective alignment can be recorded with one out of three potential tuples: 1. A *match* between score note and performance note, i.e., `(score_id, performance_id)`, 2. a *deleted* score note `(score_id, )` which represents a score note omitted in the performance, or 3. an *inserted* performance note `( , performance_id)`, which marks a performed note for which there is no corresponding score note.

Following this alignment encoding, each line in a match file corresponds to either a *match*, a *deletion* or an *insertion*. Additional lines express (sustain or soft) pedal information, or encode meta information about the musical piece and performer. While the performance part in match corresponds to a lossless encoding of a corresponding performance in MIDI format, the score part captures essential information including onset, offset and duration in beats, and pitch, pitch spelling, and octave information for each score note.

3.3 Curation protocol

To create note-level score-to-performance alignments, encoded in the match file format, between the performance MIDI data by pianist Roland Batik and scores and musicological annotations by Hentschel et al. [1], we follow the workflow as outlined below (see Fig. 1):

1. **Retrieve information from old alignment.** Given an old alignment file, we use partitura [17] to retrieve a score and performance representation which we parse into score and performance note arrays, `sna_o` and `pna_o`, to sequentially capture each (notated and performed) note with a unique note ID. In addition we retrieve a score-to-performance alignment, `align_o`, in the encoding format explained above (i.e., a list of note ID tuples expressing either a match, deletion or insertion).

2. **Retrieve score note array from MusicXML.** In the next step, we convert the annotated MuseScore format scores provided by Hentschel et al. [1] to MusicXML, assign unique note IDs to each note, and convert this score representation into a second score note array (`sna_s`).

3. **Unfold score note array.** We update the score note array obtained from MusicXML, `sna_s`, by unfolding it in accordance to the repetition structure found in the performance note array, `pna_o`. [6]

4. **Create score-score alignment.** In this step, we create a score-to-score alignment (`align_s`) by matching each note in the two score note arrays `sna_o` and `sna_s` using its pitch, onset and duration information in beats. Any notes in `sna_o` and `sna_s` not matched automatically need to be aligned manually. Missed alignments at this stage can occur due to:

   - **Score mistakes**. These reflect mistakes in the score (e.g., a missing note, incorrect pitch, octave, missing modifier, missing repetition or ending markings) and require a manual correction of the score file.
   - **Differing score versions**. For certain sonata movements, the notated score provides an alternative score version reflecting the first edition ("Erstdruck") for certain segments of a piece, expressing the composer's impromptu ornamentation. [7] For the current dataset, such ornamented versions exist in K.284iii, K.332ii, K.457iii.
   - **Double-voiced score notes**. These occur frequently in notated music, and describe a score note that is notated doubly in two different voices but corresponds to one performed note.
   - **Grace notes**. Grace notes in notated music can occur in multiple forms to reflect different types of ornaments such as trills, acciaccature, mordents, turns etc. Depending on the ornament type and the underlying score encoding format, this may result in several notes occurring at the same (notated) onset (and hence

[6] To reflect the same note occuring in a repeated segment, a suffix is added to the ID to reflect the number of occurrence, i.e. for a note with ID `n14`, the repeat structure unfolding is expressed as `n14-1` for the first, and `n14-2` for the second occurrence, respectively.

[7] `https://www.henle.de/en/music-column/mozart-piano-sonatas/`

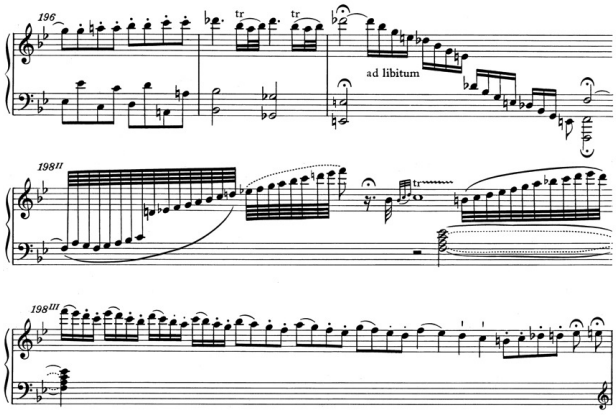


Figure 2. An example of a cadenza within a piano sonata starting in measure 198 in KV333, 3rd movement.

with zero duration) to ensure a regular measure according to the time signature of that piece. Without onset and duration information, these notes must then be manually aligned to their corresponding performed notes.

- **Cadenza and *ad libitum* measures.** Both cadenza measures and those marked *ad libitum* correspond to irregular measures, that is, measures that contain more beats than indicated in the time signature (see Fig. 2). Digitally encoded, the notes in such measures are commonly notated without duration to allow for error-free parsing, and thus share the same beat onset and need to be aligned manually.

5. **Update score-performance alignment.** Here we update the score note IDs in the old alignment (`align_o`) according to the score-score alignment (`align_s`) to create new score-performance alignments, `align_n`. For each alignment in `align_o`, we then need to ensure the validity of the original alignment type (match, insertion or deletion). In particular, for notes in the original score note array (`sna_o`) that could not be aligned to notes in the MusicXML-based score note array (`sna_s`), we consider two cases:

   - If the note in `sna_o` corresponds to type 'match' in `align_o`, the alignment type for the formerly matched performance note is changed accordingly into an insertion.
   - If the note in `sna_o` corresponds to type 'deletion' in `align_o` (i.e., a score note that was not performed), it is is discarded in `align_n`.

   Notes in `sna_s` that could not be aligned with notes in `sna_o`, on the other hand, are recorded as type 'deletion' in `align_n`.

6. **Create match files.** Using the updated performance-to-score alignment `align_n`, we create new match files, and manually add attributional information (e.g., 'diff_score_version', 'voice_overlap') to score notes to reflect edge cases described in step 4.

4. DATASET OVERVIEW

The *Batik-plays-Mozart* dataset contains performances by pianist Roland Batik of twelve Mozart sonatas (see Table 2 for the list of sonatas), corresponding to approx. 102,400 played notes and 223 minutes of music, for which the performances are provided in MIDI, musical scores in MusicXML, and the alignment in match file format. Approximately 98,300 (95.36%) of all performed notes are aligned with a corresponding score note, the remaining 4,100 (4.44%) represent insertions (reflecting mostly ornaments). Roughly 200 score notes have been omitted in the performances.

For each performance, we also provide the performance note arrays, which capture each played note with its note ID along with onset and duration information in seconds and MIDI ticks, as well as velocity and pitch information. Likewise, the dataset includes the score note array (unfolded according to the repeats as played by the pianist and reflected in the alignment), which captures each score note with its (MusicXML) note ID (including repeat suffices, where applicable), onset and duration information in terms of beats (reflecting the time signature), and quarter notes (reflecting a "normalized" score time unit), and pitch and voice information.

We link our aligned score note arrays to the musicological annotations in [1] via their temporal position in the following way: In the second version [8] of the dataset, each annotation label for harmonies, cadences, and phrases is unequivocally referenced to a temporal score position represented in terms of quarterbeats and measure number, where the first expresses the distance of the label from the beginning of the piece in quarter note units. We leverage these two temporal parameters to link each note-aligned score note array by first reducing it to its shortest form (without any unfolded repeats), aligning it temporally with the musicological annotations, and eventually unfolding it according to the performed repetition structure.

5. DATASET DEMONSTRATIONS

This section presents two simple examples of the kinds of studies that are made possible by our dataset. The first is motivated by a directly related study in the Annotated Mozart Sonata corpus paper [1]; the second shows how precise performance alignments permit more detailed investigations relating to cadences and their performance.

5.1 Global tempo and harmonic density

In a first study, we replicate the second experiment in Hentschel et al. [1], aimed at investigating the relationship

[8] `https://github.com/DCMLab/mozart_piano_sonatas`

Sonata	Performed Notes	Duration (min)	Match Notes	%	Insertion Notes	%	Deletion Notes	%
KV279	7,789	16.21	7,385	94.087	404	5.780	11	0.130
KV280	6,277	14.69	6,070	95.793	207	3.983	13	0.223
KV281	7,030	14.43	6,396	90.450	634	9.393	11	0.160
KV282	5,761	14.77	5,552	96.197	209	3.467	20	0.337
KV283	8,231	17.39	7,915	95.657	316	4.233	9	0.107
KV284	13,386	25.92	12,691	93.763	695	6.033	27	0.203
KV330	7,869	18.47	7,589	96.857	280	3.047	7	0.100
KV331	11,760	22.64	11,595	98.283	165	1.370	45	0.347
KV332	9,013	17.84	8,660	93.417	353	6.210	24	0.373
KV333	9,137	20.40	8,827	96.723	310	3.120	16	0.157
KV457	7,290	18.24	7,022	96.043	268	3.843	9	0.110
KV533	8,878	22.12	8,616	97.027	262	2.837	15	0.137
Total	102,421	223.12	98,318	95.358	4,103	4.443	207	0.199

Table 2. List of sonatas in the *Batik-plays-Mozart* dataset. The bottom row represents the sum in all columns except for those expressing percentages, for which the mean is shown.

between tempo and harmonic change rate. The basic question asked in [1] was whether the rate at which the harmony changes in a piece is correlated with the piece's typical performance tempo. Their study involved determining the average (median) performance duration of each sonata movement from 6 complete commercial sonata recordings, and correlating harmonic label density (rate of harmonic labels in their annotations, per performance time unit) with average overall performance tempo (number of quarter notes per performance time unit). We repeat the same experiment with our pianist's performances and our alignment files instead of 6 pianists' audio recordings.

We apply the same procedure as in [1], unfolding the score according to the repeat structure of the piece in order to calculate the actual piece length (in terms of quarter notes). The only difference is that we do this according to the repeats actually performed by the pianist (which are expressed in our match files, thus omitting the need for a dedicated "unfolding" step), whereas [1] seem to have assumed that all repeats were played by all pianists.

Comparing our results (Fig. 3) to Fig. 10 in [1], we see a similar general trend, in the form of a roughly linear increase in harmonic label density with performed tempo (slope = .43, r= .75, compared to .48 and .80, respectively, in [1]). [9] However, we also immediately see a marked difference in the performance tempo distribution: in [1], Fig. 10, there is a relatively large cloud of points (sonata movements) with conspicuously high tempos of 180–200 (quarters per minute), which does not appear in our plot, and which we believe may point to a systematic problem in their way of estimating playing tempo: assuming that all notated repeats are played out by the performers leads them to overestimate the tempo in all cases where some or a majority skipped some repeats. [10]

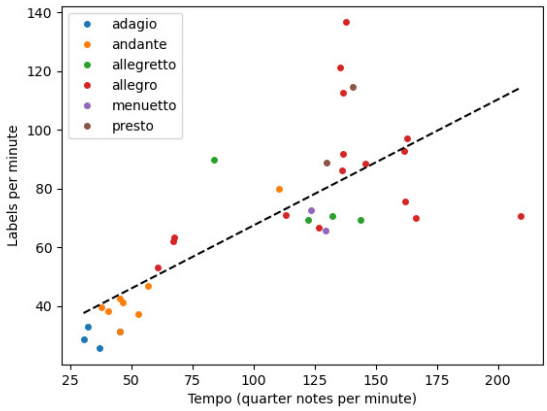


Figure 3. Correlation between global tempo (as measured in quarter notes per minute) and harmony label density

We thus see an immediate advantage of our more precise performance-aligned corpus: the match files naturally give correct tempo and score duration information, being based as they are on score-performance alignments that reflect the actual repeat structure played by our performer. Still, we can say that our results support and confirm the overall hypotheses proposed there, showing a more or less linear relationship between harmonic label density and global performance tempo.

5.2 Performance of different cadence types

Our data permits much more detailed investigations into relationships between structural aspects of a piece, and how these are translated into performance decisions by a pianist. As a simple example, we investigate variations in local tempo before various types of cadences. Specifically, we compare the local tempo prior to a cadence annotation across different tempo classes for authentic (perfect and imperfect, i.e., PAC and IAC) and half cadences (HC), and differentiate between the cases when a cadence falls on either a downbeat or a weak beat. The hypothesis to be tested

[9] Note that we have a somewhat smaller set of points, because we only have 12 of the 18 sonatas in our dataset.

[10] Of course, the authors explicitly acknowledge the problem: "Also, some of the initial assumptions might have to be revisited. For example, the extreme outlier suggesting a tempo of 239 quarter notes per minute is due to the fact that for this particular piece – the first movement of K. 533/494 – there seems to be a convention among pianists to repeat the first part of the piece, but not the second (as the score would suggest), which of course reduces the performance duration." [1] (p.76), but a comparison with our distribution implies it might be more severe than expected.

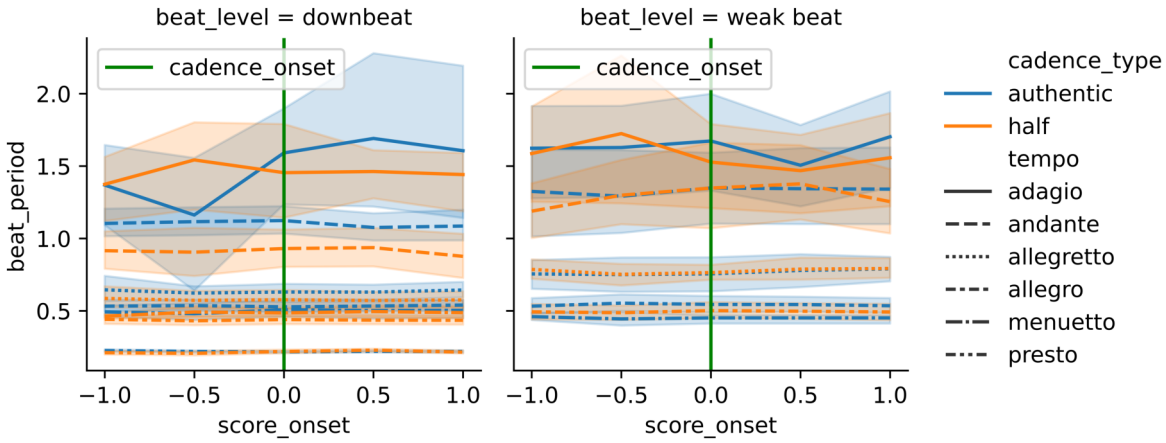


Figure 4. Comparison of local timing strategies one quarter note before and after authentic and half cadence labels, over different tempo classes (in increasing tempo from top to bottom), for cadences falling on a downbeat (left) or weak beat (right). Colour identifies cadence type, line style notated tempo class.

here is that a performer will tend to shape cadences differently, in terms of tempo, depending on their type and degree of 'finality'.

To compute the local tempo curves, we consider a uniform window spanning one quarter note each preceding and following a cadence label.[11] For each score-note-aligned performed note in that window, we define the local tempo via the *beat period (BP)*, which we calculate as the ratio of the inter-onset-interval (IOI) between the current *performed* onset and the subsequent one, and the IOI between the current *notated* onset and subsequent one. We exclude grace notes and their corresponding performed notes from this calculation in order to remove outliers.

Next, we perform time-wise interpolation on these tempo curves to obtain beat period values at eighth note intervals within the window. Given that we are most interested in the local timing strategy immediately before a label (that is, an eighth note before the label position), we discard those curves where that particular time point is interpolated. Following this procedure, we obtain a total of 3,540 local tempo values (corresponding to 708 curves), of which 251 (7.09%) values are interpolated.

Figure 4 shows the mean of local tempo curves across different tempo classes, for cadence labels annotated on a downbeat (left) and on a weak beat (right), respectively. For both authentic and half cadence types, the differences in local tempo diminish with increasing global tempo for both downbeat and weak beat cadences. Likewise, the tempo profiles tend to flatten out with increasing global tempo, suggesting that the pianist takes more liberty, in terms of expressive timing, in slow pieces. For this reason, we focus our analysis on the *adagio* tempo class, the slowest tempo (the solid line plots in Fig. 4) .

The influence of the beat level on the local tempo for half cadences seems to be negligible, with the local beat period decreasing slightly prior to the cadence (causing an increase in local tempo, i.e. an *accelerando*), regardless of whether it falls on a downbeat or weak beat. For authentic cadences, we can see a substantial difference in expressive tempo depending on whether or not the label falls on a downbeat: for authentic cadences falling on a downbeat, the mean tempo curve for the *adagio* tempo class corresponds mostly to what one would expect (i.e., a very clear *ritardando* in preparation of the cadence) based on the underlying harmonies and their notion of tension and release. Interestingly, this *ritard* seems to continue somewhat after the resolution into the tonic, suggesting a lengthening of the tonic arrival. For weak-beat authentic cadences, a similar significant preparation or anticipation is largely missing.

6. CONCLUSION AND FUTURE WORK

We have presented *Batik-plays-Mozart*, a piano performance dataset linking professional Mozart piano sonata performances to expert-labelled musical scores, at the level of notes. The resulting dataset is the first of its kind to combine professional performances in precise, recorded MIDI with curated musical scores and expert musicological and structural annotations [1] at this level of detail.

We presented two preliminary experiments, intended to demonstrate the benefits of having such precise, note-aligned performance–score–structure annotation data for studying expressive features and their relation to the underlying musical structure.

Our plan for future work includes the transcription of the remaining six sonatas of the Mozart piano sonatas corpus from audio recordings by the same pianist, and their subsequent alignment to the musical scores using state-of-the-art transcription and alignment models. By doing so, we hope to advance our understanding of the differences between transcribed and recorded MIDI, and to evaluate the potential benefits of incorporating an alignment step to improve the quality of transcription.

[11] In the *Annotated Mozart Sonatas Corpus* [1], cadence labels are placed at the onset of the final target harmony (e.g., I/i for authentic cadences).

7. ACKNOWLEDGMENTS

We wish to express our gratitude to pianist Roland Batik for his gracious permission to publish the detailed measurements of his performances. We also want to thank the authors of the *Annotated Mozart Sonatas Corpus* for their tremendous efforts, and for permitting us to link our data to theirs. 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.

8. REFERENCES

[1] J. Hentschel, M. Neuwirth, and M. Rohrmeier, "The Annotated Mozart Sonatas: Score, Harmony, and Cadence," *Transactions of the International Society for Music Information Retrieval (TISMIR)*, vol. 4, no. 1, pp. 67–80, 2021. [Online]. Available: https://doi.org/10.5334/tismir.63

[2] C. E. Cancino-Chacón, M. Grachten, W. Goebl, and G. Widmer, "Computational Models of Expressive Music Performance: A Comprehensive and Critical Review," *Frontiers in Digital Humanities*, p. 25, 2018.

[3] 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 International Society for Music Information Retrieval Conference (ISMIR)*, 2019, pp. 908–915.

[4] A. Maezawa, K. Yamamoto, and T. Fujishima, "Rendering Music Performance with Interpretation Variations using Conditional Variational RNN," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2018.

[5] H. Zhang, J. Tang, S. R. M. Rafee, and S. D. G. Fazekas, "ATEPP: A Dataset of Automatically Transcribed Expressive Piano Performance," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2022.

[6] 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)*, May 2022.

[7] 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, pp. 534–541.

[8] W. Goebl. (1999) The Vienna 4x22 Piano Corpus. [Online]. Available: http://dx.doi.org/10.21939/4X22

[9] M. Hashida, E. Nakamura, and H. Katayose, "Crest-MusePEDB 2nd Edition: Music Performance Database with Phrase Information," in *Proceedings of the 15th Sound and Music Computing (SMC) Conference*, 2018.

[10] K. Kosta, O. F. Bandtlow, and E. Chew, "MazurkaBL: Score-Aligned Loudness, Beat, Expressive Markings Data for 2000 Chopin Mazurka Recordings," in *Proceedings of the Fourth International Conference on Technologies for Music Notation and Representation (TENOR) (Montreal, QC)*, 2018, pp. 85–94.

[11] 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. [Online]. Available: https://openreview.net/forum?id=r1lYRjC9F7

[12] 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)*, 2020.

[13] W. Goebl and R. Bresin, "Measurement and Reproduction Accuracy of computer-controlled Grand Pianos," *The Journal of the Acoustical Society of America*, vol. 114, no. 4, pp. 2273–2283, 2003.

[14] G. Widmer, "Discovering simple rules in complex data: A meta-learning algorithm and some surprising musical discoveries," *Artificial Intelligence*, vol. 146, no. 2, pp. 129–148, 2003.

[15] E. Cambouropoulos, "From MIDI to traditional musical notation," in *Proceedings of the AAAI Workshop on Artificial Intelligence and Music: Towards Formal Models for Composition, Performance and Analysis*, vol. 30, 2000.

[16] F. Foscarin, E. Karystinaios, S. D. Peter, C. Cancino-Chacón, M. Grachten, and G. Widmer, "The match file format: Encoding Alignments between Scores and Performances," in *Proceedings of the Music Encoding Conference (MEC)*, 2022.

[17] C. Cancino-Chacón, S. D. Peter, E. Karystinaios, F. Foscarin, M. Grachten, and G. Widmer, "Partitura: A Python Package for Symbolic Music Processing," in *Proceedings of the Music Encoding Conference (MEC)*, 2022.

MONO-TO-STEREO THROUGH PARAMETRIC STEREO GENERATION

Joan Serrà Davide Scaini Santiago Pascual Daniel Arteaga Jordi Pons Jeroen Breebaart Giulio Cengarle

Dolby Laboratories

firstname.lastname@dolby.com

ABSTRACT

Generating a stereophonic presentation from a monophonic audio signal is a challenging open task, especially if the goal is to obtain a realistic spatial imaging with a specific panning of sound elements. In this work, we propose to convert mono to stereo by means of predicting parametric stereo (PS) parameters using both nearest neighbor and deep network approaches. In combination with PS, we also propose to model the task with generative approaches, allowing to synthesize multiple and equally-plausible stereo renditions from the same mono signal. To achieve this, we consider both autoregressive and masked token modelling approaches. We provide evidence that the proposed PS-based models outperform a competitive classical decorrelation baseline and that, within a PS prediction framework, modern generative models outshine equivalent non-generative counterparts. Overall, our work positions both PS and generative modelling as strong and appealing methodologies for mono-to-stereo upmixing. A discussion of the limitations of these approaches is also provided.

1. INTRODUCTION

Single-channel monophonic (mono) signals are found in multiple situations, such as historical recordings or current ones made with a single microphone (e.g., field recordings, amateur band rehearsals, etc.). Even recordings made with two or more microphones that are not spaced enough or that do not have enough directivity may be better treated by downmixing to mono (e.g., mobile phone recordings). Furthermore, many processing algorithms, including modern deep neural network algorithms, cannot yet or are simply not designed to handle more than one channel. Unlike these scenarios, the most common listening experiences, either though loudspeakers or headphones, involve two-channel stereophonic (stereo) signals. Hence the usefulness of mono to stereo upmixing.

Classical approaches to produce a pseudo-stereo effect from a mono signal are based on decorrelation. Initial approaches used time delays and complementary filters [1], although all-pass filters [2] are commonly used nowadays, together with multi-band processing to improve the effect [3–5]. Instead of multi-band, estimation of foreground/background time-frequency tiles can also be performed [6]. Decorrelation approaches, however, only provide a mild stereo effect, with limited width, and cannot spatially separate individual elements in the mix. To overcome the latter, researchers have considered source separation approaches [7–9]. The main idea is that, if individual elements or tracks are available, those can be panned to any location, producing a more realistic spatial image. Nevertheless, this approach presents several drawbacks: firstly, even the best-performing source separation algorithms produce artifacts [10], which can be highly audible in the stereo render; secondly, current separation algorithms are very restrictive in the number and types of elements they can separate [11], thus considerably limiting their application in real-world spatialization tasks; thirdly, after elements or tracks are separated, it remains to be seen how can they be automatically panned in a realistic manner (cf. [12]), which is the reason why separation-based approaches usually involve user intervention in the panning stage [7–9].

Music is a paradigmatic example where, apart from stereo capture, artists and engineers massively exploit the stereo image to serve a creative artistic intent. Instrument panning is a fundamental part of music mixing, and achieving the right balance requires musical sensibility as well as technical knowledge [13]. However, apart from some style conventions, the stereo image of a music mix is a highly subjective construct: given a set of input tracks, there are many plausible stereo renditions from which selecting the final mix is practically only a matter of artistic choice. Hence, we posit that this is a perfect ground for modern deep generative models [14]. However, to our surprise, we only found one work using deep neural networks for mono-to-stereo [15], with very limited generative capabilities.

In this work, we propose the use of machine learning techniques and parametric stereo (PS) decoding [16,17] for converting mono to stereo. PS is a coding technique that allows to transmit a stereo signal through a mono signal plus side information that, with enough bit rate, can be used to recover an almost transparent version of the original stereo content. By leveraging machine learning techniques, we generate (or invent) plausible versions of PS parameters in situations where side information is not available. These

© J. Serrà, D. Scaini, S. Pascual, D. Arteaga, J. Pons, J. Breebaart, and G. Cengarle. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** J. Serrà, D. Scaini, S. Pascual, D. Arteaga, J. Pons, J. Breebaart, and G. Cengarle, "Mono-to-stereo through parametric stereo generation", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

parameters can then be used to decode an existing mono signal into a plausible stereo one. We propose two variants of PS generation: one based on a classical nearest neighbor approach [18] and another one based on deep generative modeling. For the latter, we consider both common autoregressive modeling [19] and more recent masked token modeling [20], and show that there can be noticeable differences between the two. We use subjective testing to compare the proposed approaches and show that PS generation can produce results that are more appealing than considered competitive baselines. We also introduce two objective evaluation metrics and discuss the limitations of both PS and generative approaches for mono-to-stereo.

2. PARAMETRIC STEREO

PS exploits the perceptual cues that are more relevant to our spatial perception of sound, namely the fact that directional sources produce interaural level and phase (or time delay) differences, and the fact that diffuse sound fields manifest as decorrelated signals at the two ears. These cues effectively describe how a mono signal is mapped to the left and right stereo channels, and can be measured using three quantities or parameters [16, 17]: interchannel intensity differences (IID), interchannel time differences (or, equivalently, phase differences), and interchannel coherence or correlation (IC). PS parameters are computed in frequency bands, to reflect the frequency-dependent nature of the spatial properties of stereo content, and also on a frame-by-frame basis, to reflect the time-varying nature of frequency cues and spatial images. An important observation is that PS is capable of capturing spatial attributes that are perceptually relevant and re-instate those without changing signal levels, tonality, or other artifacts that may arise from methods that operate on audio signals directly. In this work, for compactness and ease of implementation, we choose to use the two-parameter approach by Breebaart et al. [17], which models IID and IC without interchannel phase differences, accepting that this two-parameter approach is not providing the best possible quality of PS coding. We now overview this PS coding strategy and introduce the main notation of the article.

2.1 Encoding

Given two complex-valued spectrograms expressed as complex matrices $\mathbf{X}$ and $\mathbf{Y}$, where rows represent frequency bins and columns represent frames, we define the band-based cross-spectrogram function

$$\rho(\mathbf{X}, \mathbf{Y}) = \mathbf{B}\,(\mathbf{X} \odot \mathbf{Y}^*),$$

where $\odot$ denotes elementwise multiplication, * denotes elementwise complex conjugate, and $\mathbf{B}$ is a matrix with ones and zeros that is used to sum frequency bins according to a certain frequency band grouping (using matrix multiplication). In this work, we use the same spectrogram settings and banding as in [17]: frames of 4,096 samples for 44.1 kHz signals, 75% overlap, a Hann window, and 34 bands which are approximately distributed following equivalent rectangular bandwidths.

Given the two complex spectrograms $\mathbf{L}$ and $\mathbf{R}$ corresponding to the left and right channels of a stereo signal, we can compute the IID using

$$\mathbf{P}^{\text{IID}} = 10 \log_{10} \left(\rho(\mathbf{L}, \mathbf{L}) \oslash \rho(\mathbf{R}, \mathbf{R}) \right),$$

where $\oslash$ denotes elementwise division. The IC is similarly derived from the cross-spectrogram following

$$\mathbf{P}^{\text{IC}} = \text{Re}\left\{\rho(\mathbf{L}, \mathbf{R})\right\} \oslash \sqrt{\rho(\mathbf{L}, \mathbf{L}) \odot \rho(\mathbf{R}, \mathbf{R})},$$

where $\text{Re}\{\}$ extracts the real part of each complex value and the square root is applied elementwise. Notice that the use of the real part instead of the absolute value allows to retain information on the relative phase of the two signals that would otherwise be lost. We finally quantize $\mathbf{P}^{\text{IID}}$ and $\mathbf{P}^{\text{IC}}$ by discretizing each matrix element. To do so, we use the same non-uniform quantization steps as in [17]: 31 steps for IID and 8 for IC. We denote the quantized versions as $\mathbf{Q}^{\text{IID}}$ and $\mathbf{Q}^{\text{IC}}$.

To facilitate subsequent operation, and to prevent potential prediction mismatches between IID and IC, we join both parameters and treat them as one. For $\mathbf{P}^{\text{IID}}$ and $\mathbf{P}^{\text{IC}}$, we concatenate them in the frequency axis and form a single matrix $\mathbf{P}$. For $\mathbf{Q}^{\text{IID}}_{i,j}$ and $\mathbf{Q}^{\text{IC}}_{i,j}$, we fuse them elementwise into individual integers using the amount of IC quantization steps. This way, $\mathbf{Q}_{i,j} = 8 \cdot \mathbf{Q}^{\text{IID}}_{i,j} + \mathbf{Q}^{\text{IC}}_{i,j}$ (note that we can recover back $\mathbf{Q}^{\text{IID}}_{i,j}$ and $\mathbf{Q}^{\text{IC}}_{i,j}$ using the division and modulo operators).

2.2 Decoding

To decode the above PS encoding, we perform a mixing between the available mono signal and a decorrelated version of it. We decorrelate a mono signal $\mathbf{S}$ by applying a cascade of 4 infinite impulse response all-pass filters and obtain $\mathbf{S}^{\text{D}}$ (this all-pass filter is an enhanced version of the basic one proposed in [17] thanks to transient detection and preservation, which avoids time smearing). After that, we can decode the estimated left and right channels $\hat{\mathbf{L}}$ and $\hat{\mathbf{R}}$ by carefully mixing $\mathbf{S}$ and $\mathbf{S}^{\text{D}}$. We can do so with

$$\hat{\mathbf{L}} = \mathbf{M}^{\text{a}} \odot \mathbf{S} + \mathbf{M}^{\text{b}} \odot \mathbf{S}^{\text{D}},$$
$$\hat{\mathbf{R}} = \mathbf{M}^{\text{c}} \odot \mathbf{S} + \mathbf{M}^{\text{d}} \odot \mathbf{S}^{\text{D}},$$

using mixing matrices $\mathbf{M}$, which are computed from the coded PS parameters $\mathbf{P}^{\text{IID}}$ and $\mathbf{P}^{\text{IC}}$. The exact calculation of mixing matrices $\mathbf{M}$ is straightforward to obtain by adapting to matrix notation the formulation in [17], to which we refer for further detail and explanation.

3. PARAMETRIC STEREO GENERATION

We now explain the proposed approaches for PS generation. All of them share the above encoding-decoding formulation, either using the quantized or unquantized versions. During training, stereo signals are used to compute input downmixes $\mathbf{S} = (\mathbf{L} + \mathbf{R})/2$ and target PS parameters $\mathbf{P}$ or $\mathbf{Q}$ (hence the proposed approaches aim at producing

$\hat{\mathbf{P}}$ or $\hat{\mathbf{Q}}$). Note that, in the case of the generative models we consider, one has to additionally input contextual PS parameters in a teacher-forcing schema [21]. We also want to note that, since they are quite common practice, it is not in the scope of the current work to provide a detailed explanation of existing generative models (instead, we refer the interested reader to the cited references). In all proposed approaches, we tune model hyperparameters by qualitative manual inspection in a preliminary analysis stage. PS specifications are predefined and correspond to the ones mentioned in Sec. 2. Neural network approaches use Pytorch's [22] defaults and are trained with Adam for 700 epochs using a batch size of 128 and a learning rate of 10^{-4}, with warmup cosine scheduling.

3.1 Nearest neighbor

The first approach proposes to impose the PS parameters of existing, similar stereo fragments to individual mono frames using a nearest neighbor (NN) algorithm [18]. We call the approach PS-NN. The idea is to retrieve frame-based PS parameters using mono frame sequences, and to use the sequence of those retrieved parameters to decode the mono input. At training time, we randomly select a song, randomly extract an $N = 20$ frame spectrogram $\mathbf{S}$ and its corresponding parameters $\mathbf{P}$, and compute a key-value vector pair (we here use the magnitude spectrogram). The key vector is formed by framewise averaging the energy in each band,

$$\mathbf{k} = \frac{1}{N} \sum_{j=1}^{N} \mathbf{B}\, \mathbf{S}_{:,j}, \tag{1}$$

and the value vector corresponds to the PS parameters of the last frame, $\mathbf{v} = \mathbf{P}_{:,N}$, which allows for a fully-causal schema. We repeat the process half a million times and store all pairs in a nearest neighbor structure. At test time, for every frame of the input mono signal, we compute an average as in Eq. 1, query the nearest neighbor structure, retrieve the $\hat{\mathbf{v}}$ vector of the closest neighbor (using Euclidean distance), and assign it as the predicted PS parameter for that frame. This way, we obtain a sequence of estimated PS parameters $\hat{\mathbf{P}}$.

In preliminary analysis, we observed that PS-NN produced a high-rate 'wobbling' effect between left and right (that is, panning was rapidly switching from one channel to the other) and presented some temporal inconsistencies (that is, sources were unrealistically moving with time, even within one- or two-second windows). To counteract these effects, we implemented a two step post-processing based on (i) switching the sign of $\hat{\mathbf{P}}^{\text{IID}}_{:,j}$ if the Euclidean distance to $\hat{\mathbf{P}}^{\text{IID}}_{:,j-1}$ was smaller, and (ii) applying an exponential smoothing on the columns of $\hat{\mathbf{P}}$ with a factor of 0.95. This post-processing substantially reduced the aforementioned undesirable effects.

3.2 Autoregressive

The second approach proposes to model PS parameters with a deep generative approach based on an autoregressive (AR) transformer [19]. We call the approach PS-AR. Our architecture is composed by 7 transformer encoder blocks of 512 channels, with 16 heads and a multilayer perceptron (MLPs) expansion factor of 3. We use sinusoidal positional encoding at the input, and add a two-layer MLP with an expansion factor of 2 at the output to project to the final number of classes (which is 31×8 tokens times 34 bands per frame, see Sec. 2.1). The input is formed by a projection of the mono spectrogram $\mathbf{S}$ and the teacher-forcing information $\mathbf{Q}$ into a 512-channel activation $\mathbf{H}$,

$$\mathbf{H} = \phi(\mathbf{S}) + \sum_{i=1}^{B} \xi_i(\mathbf{Q}_{i,:}), \tag{2}$$

where ϕ is a two-layer MLP with an expansion factor of 2, $B = 34$ is the number of bands, and ξ_i is a learnable per-band token embedding (which includes the mask token, see below). We train the model with weighted categorical cross-entropy, using the weight

$$w = 1 + \lambda \sigma \left(\left[\mathbf{P}^{\text{IID}} \right]_{\pm\epsilon} \right) + \sigma(\mathbf{P}^{\text{IC}}), \tag{3}$$

calculated independently for every element in the batch. In Eq. 3, $\sigma(\mathbf{X})$ corresponds to the elementwise standard deviation of $\mathbf{X}$, $\lambda = 0.15$ compensates for different magnitudes, $[\ ]_{\pm\epsilon}$ corresponds to the clipping operation, and $\epsilon = 20$ is a threshold to take into account the little perceptual relevance of IIDs larger than 20 dB [23]. In preliminary analysis, we observed that using w qualitatively improved results, as it shall promote focus on wider stereo images and more difficult cases.

PS-AR follows a PixelSNAIL recursive approach [24], starting with the prediction of lower frequency bands, then higher frequency bands, and moving into the next frame once all bands are predicted. To efficiently exploit the past context, all input sequences have full-sequence teacher-forcing except for the upper frequency bands of the last frame, which are masked consecutively and uniformly at random during training [24]. At test time, we sample recursively, following the same masking strategy and using a temperature hyperparameter $\tau = 0.9$. In addition, we employ classifier-free guidance [25] with a hyperparameter $\gamma = 0.25$. For that, we use the approach in [26], which modifies the conditional logits $\mathbf{U}^{\text{cond}}$ with unconditional ones $\mathbf{U}^{\text{uncond}}$ such that

$$\mathbf{U} = (1 + \gamma)\mathbf{U}^{\text{cond}} - \gamma \mathbf{U}^{\text{uncond}}. \tag{4}$$

To have both a conditional and an unconditional model within the same architecture, following common practice, we randomly replace $\phi(\mathbf{S})$ in Eq. 2 by a learnable dropout token 10% of the time.

3.3 Masked token modeling

The third approach proposes to model PS parameters with a deep generative approach based on masked token modeling (MTM) [20]. We call the approach PS-MTM. The architecture, loss, inputs, and outputs of the model are the same as in PS-AR, including the cross-entropy weights

(Eq. 3) and classifier-free guidance (Eq. 4). The only difference is the masking approach and the sampling procedure, which implies different hyperparameters for the testing stage (we use $\tau = 4.5$ and $\gamma = 0.75$, but now the temperature τ has a different meaning as explained below).

MTM generates patch representations $\mathbf{Q}$ with quantized elements $\mathbf{Q}_{i,j}$ which are dubbed as tokens (in our case the matrix $\mathbf{Q}$ has dimensions $B \times N$, with N being the number of considered audio frames; the maximum number of tokens in $\mathbf{Q}_{i,j}$ is 31×8, as defined in Sec. 2.1). During training, the teacher-forcing input $\mathbf{Q}$ is masked uniformly at random, and only the ground truth elements corresponding to the masked positions are used to compute the cross-entropy loss at the output. The number of elements to mask is also selected at random following a cosine schedule [20] (this specifically includes the case where all patch elements are masked). During sampling, patch representations are formed with 50% overlap, using no masking for the first half of the patch, similar to [26].

MTM sampling is an iterative process that achieves orders of magnitude speedups compared to autoregressive modeling (in our case PS-MTM uses 20 steps for a 3 s hop, while PS-AR requires $B = 34$ steps for just a single audio frame of a few milliseconds). MTM iteratively samples a masked patch, performs predictions with classifier-free guidance [26], chooses the predictions with the highest logit score for the next iteration (they will become unmasked and fixed), and reduces the percent of masked tokens following the same scheduling as in training until no masked elements remain [20]. Differently from training, the masking used in sampling is not random, but based on logit scores (lowest ones become masked), and noise is added to logit scores to promote diversity [20, 26]. In our case, we employ Gaussian noise with zero mean and a standard deviation τ, which becomes our redefined temperature parameter.

4. EVALUATION

To train and evaluate all approaches we use a collection of professionally-recorded stereo music tracks at 44.1 kHz. We consider 419,954 tracks for training and 10 k for evaluation, and randomly extract a 10 s chunk from each track. During training, we sample 6 s patches from those and perform data augmentation using a random gain and also randomly switching left and right channels.

4.1 Baselines: regression and decorrelation

In addition to the original stereo and its mono downmix, we consider two additional baselines to compare with the previous approaches. The first baseline corresponds to an ablation of the deep generative approaches, and tries to answer the question of whether a generative component is needed or convenient for the task. Thus, the baseline consists of a neural network with the exact same configuration as PS-AR or PS-MTM, but substituting the generative part by standard regression with mean-squared error [18]. We term this baseline PS-Reg, and note that it could be considered an enhanced modern version of the approach of Chun et al. [15], using PS.

It is interesting to mention that, in preliminary analysis, we observed that PS-Reg accurately estimated IC values, but consistently failed to predict IIDs. The predicted IIDs had minimal deviation from zero, which can be attributed to the probability distribution function of IID values being centered around zero with equally plausible deviations to the right and to the left. This was an early indication that the one-to-many mapping of IID prediction cannot be correctly handled by regression methods, and that the task would be better served by a generative approach.

The second baseline we consider corresponds to a variant of classical decorrelation approaches. Here, the decorrelation is implemented by means of an all-pass filter network enhanced by (i) detection and preservation of transients, and (ii) a frequency-dependent mix between original and decorrelated signals to achieve a frequency-dependent IC. We term this baseline Decorr, and we note that it could be considered an improved modern version of the approaches [1–6].

4.2 Objective measures

To the best of our knowledge, there are no objective measurements for plausible stereo renderings nor suitable PS prediction scores. Due to the highly creative/subjective nature of the task, common error measurements may not be appropriate. Therefore, as a way of measuring progress, we propose to use a couple of metrics inspired from the literature on generative modeling (cf. [14]). The first metric we consider is the minimum error on a large sample basis, $E_{\min}$. Given a large sample of generated PS parameters ($K = 128$ for a single audio excerpt), $E_{\min}$ chooses the minimum error with respect to the ground truth:

$$E_{\min} = \min_k \left[\sum_{i,j} \delta\left(\mathbf{P}_{i,j}, \hat{\mathbf{P}}_{i,j}^{(k)}\right) \right],$$

where δ is a suitable error function. The idea is that if we allow the model to generate many samples for every input, in the limit of very large K one of them should come close to the ground truth. For PS parameters, we use absolute errors, weight the IID to compensate magnitudes with IC, and take into account some perceptual relevance for IID as in Sec. 3.2 and Eq. 3:

$$\delta(x, y) = \begin{cases} \lambda \left| [x]_{\pm\epsilon} - [y]_{\pm\epsilon} \right| & \text{for IID,} \\ |x - y| & \text{for IC.} \end{cases}$$

The second metric we consider is the Fréchet distance on the PS parameter space, D_{F}. Given a pool of PS parameters $\boldsymbol{\mathcal{P}}$ and a K times larger pool of generated parameters $\hat{\boldsymbol{\mathcal{P}}}$, assuming Gaussian distributions, D_{F} is computed as

$$D_{\mathrm{F}} = \left| \mu(\boldsymbol{\mathcal{P}}) - \mu(\hat{\boldsymbol{\mathcal{P}}}) \right|^2 + \\ + \mathrm{Tr}\left\{ \sigma(\boldsymbol{\mathcal{P}}) + \sigma(\hat{\boldsymbol{\mathcal{P}}}) - 2\sqrt{\sigma(\boldsymbol{\mathcal{P}})\sigma(\hat{\boldsymbol{\mathcal{P}}})} \right\},$$

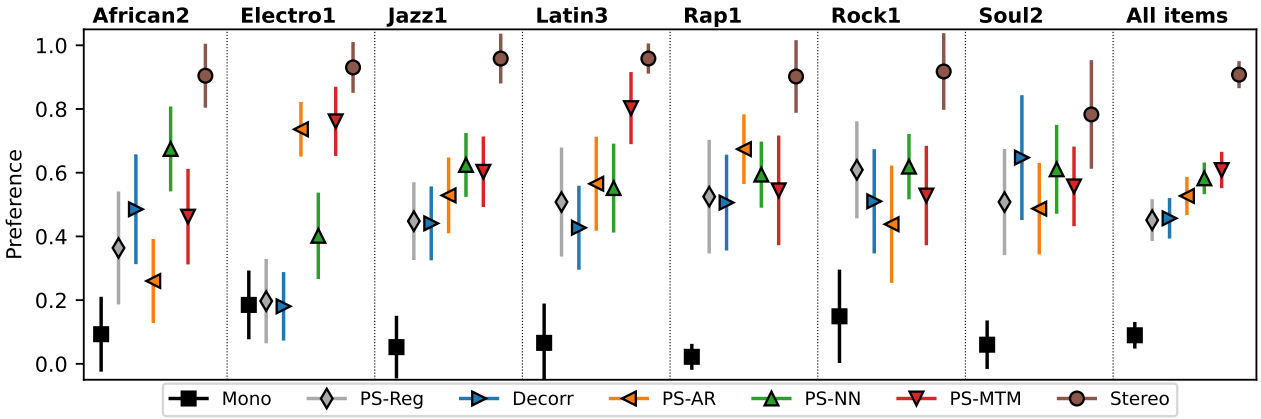


Figure 1. Preference results for the items included in the subjective test (Sec. 4.3). Markers indicate average values and vertical bars indicate the 95% confidence interval associated to them.

where Tr{} denotes the matrix trace and μ and σ correspond to the mean vector and the covariance matrix over frames, respectively. The Fréchet distance has become a standard measure in generative modeling where, instead of the PS parameters used here, activations of pre-trained classification networks are used. We will see that it is also able to provide some informative cues in our task (Sec. 5).

4.3 Subjective evaluation

Given the creative/subjective nature of the task, the best way to measure performance is through subjective testing. In this study, we ran a preference test with 24 listeners on 7 song excerpts of 10 s from the test set. To select those excerpts, we ranked the test excerpts based on w (Eq. 3) and randomly selected them from the top quartile. When doing so, we manually verified that the selected excerpts covered distinct musical genres and ensured that a PS-decoded version did not exhibit significant coding degradation (this way, we prime the listener to focus on the stereo image instead of potential artifacts introduced by our implementation of PS, Sec. 2).

The test consisted in providing a rating between 0 and 100 to 7 approaches: the three proposed ones, the two baselines, the mono downmix, and the original stereo signal (professional mix, non-coded). Mono and stereo signals provide us with intuitive bounds for the analysis of preference, and also serve us to discard non-expert listeners. Indeed, we found that the task is quite hard for non-experts, who provided many inconsistent ratings when asked to evaluate an appropriate balance between the width and the clarity of the mix. We used the most obvious of those inconsistencies to discard listeners from the test, namely the fact that they rated mono (input) over stereo (professional mix) in one or more occasions. Half of the users (12) did not incur into such inconsistency and were considered reliable enough to derive conclusions from their ratings. To compensate for differences in subjective scales, we normalized excerpt preference tuples between 0 and 1 (that is, we normalized the ratings for the 7 approaches independently per audio excerpt and listener). To measure statistical significance, we used pairwise Wilcoxon signed-rank tests and applied the Holm-Bonferroni adjustment for multiple testing with $p = 0.05$. The Wilcoxon signed-rank test is appropriate for our case as it is non-parametric and designed for matched samples.

5. RESULTS

In Fig. 1 we depict the average listener preference for each item and approach. Initially, we see that the pattern differs depending on the test item. For some items, the proposed approaches are preferred over the baselines (e.g., Electro1, Jazz1, and Latin3) while, for some other items, differences between approaches are less clear (e.g., Rock1 and Soul2). All approaches seem to be preferred above the mono signal, except for baseline approaches with Electro1. Noticeably, in some situations, preference for some of the proposed approaches even overlaps with the original stereo (e.g., Electro1, Latin3, and Soul2). The case of Soul2 shows an example where considered approaches are almost as preferred as the original stereo, whereas the case of Jazz1 shows an example where considered approaches are still far from the professional mix.

Despite the different preferences on individual excerpts, upon further inspection we see that a clear pattern emerges when considering all items: proposed approaches rank better than mono and the considered baselines (Fig. 1, right). In Table 1 we confirm that, on average, PS-AR is preferred over the baseline approaches and that, in turn, PS-NN and PS-MTM are preferred over PS-AR. In Table 2, we report statistically significant differences beetween PS-NN/PS-MTM and the baseline approaches, but not between PS-AR and the baseline approaches (and neither between PS-AR and PS-NN/PS-MTM nor between PS-NN and PS-MTM). Overall, the results show that a generative approach to PS prediction can become a compelling system for mono-to-stereo. The performance of PS-NN is a nice surprise that was not predicted by the objective metrics, which otherwise seem to correlate with listener preference (Table 1; perhaps PS-NN does not follow the trend because it is not a generative approach).

Besides quality, another aspect worth considering is

Approach	$E_{\min} \downarrow$	$D_{\mathrm{F}} \downarrow$	Preference $\uparrow$
Mono	0.104	20.89	0.090 ± 0.042
PS-Reg	0.069	8.11	0.451 ± 0.066
Decorr	0.093	8.32	0.457 ± 0.064
PS-AR	0.074	0.62	0.527 ± 0.060
PS-NN	0.089	3.08	0.582 ± 0.057
PS-MTM	0.068	0.59	0.608 ± 0.050
Stereo	0.000	0.03	0.908 ± 0.042

Table 1. Results for the objective ($E_{\min}$, D_{F}) and subjective (Preference $\pm$ 95% confidence interval) evaluations.

	PS-Reg	Decorr	PS-AR	PS-NN	PS-MTM	Stereo
Mono	✓	✓	✓	✓	✓	✓
PS-Reg		✗	✗	✓	✓	✓
Decorr			✗	✓	✓	✓
PS-AR				✗	✗	✓
PS-NN					✗	✓
PS-MTM						✓

Table 2. Pairwise statistical significance for the case of all test items (12 subjects times 7 excerpts, see Sec. 4.3). The obtained p-value threshold is 0.0053.

speed. In Table 3 we observe that PS-AR, as anticipated, is orders of magnitude slower than the other approaches, to the point of making it impractical for real-world operation. Decorr, PS-Reg, and PS-NN are faster than real-time on CPU and PS-MTM is not. However, one should note that with PS-MTM we can easily trade off sampling iterations at the expense of some quality reduction (see [20, 26]). PS-NN may dramatically improve speed if we consider the use of fast nearest neighbor search algorithms or even hash tables, which make this approach very interesting for real-world deployment (note we deliberately made PS-NN comparable in size to the other approaches, see Table 3).

6. DISCUSSION

Despite the good results obtained above, the subjective test reveals that, for some of the considered excerpts, there is still a gap between professional stereo mixes and the proposed approaches. We hypothesize that this gap is due to (i) limitations of the considered PS encoding, and (ii) the difficulty of the task itself. Regarding (i), we suspect that part of the low subjective scores of PS-based approaches is due to the audio distortions and tonal artifacts introduced by the PS decoding. Thus, we hypothesize that using a commercial implementation of PS coding (or perhaps even learning end-to-end the coding operation) could yield better results. Besides, we think that the fact that PS is defined in a banded domain poses a challenge to PS generation approaches, namely that individual bands are panned but approaches do not have an explicit notion of instrument or 'entity'. Indeed, we sometimes observe individual entities being panned into two different positions simultaneously (e.g., for the same instrument, we may get some frequencies panned to the left and some to the right, which is an uncommon stylistic decision). A potential solution to this

Approach	Learnable parameters	RTF $\downarrow$ CPU	RTF $\downarrow$ GPU
Decorr	0	0.25	n/a
PS-Reg	30.1 M	0.32	0.21
PS-NN	34.0 M†	0.82	n/a
PS-MTM	34.5 M	5.81	0.33
PS-AR	34.5 M	255.87	8.38

Table 3. Number of learnable parameters and average real-time factor (RTF). Superscript † indicates an estimation of 0.5 M key-value pairs with $B = 34$ bands (Sec. 3.1). RTFs are measured on a Xeon(R) 2.20 GHz CPU and on a GeForce GTX 1080-Ti GPU.

problem could be to add better (or more) inputs to the models, together with more capacity, with the hope that they achieve a better understanding of what is a source before panning it. Along this line, it would be perhaps interesting to include some techniques used in the context of source separation with neural network models [11]. Regarding (ii), another issue we sometimes observe is with the temporal consistency of panning decisions, with an instrument appearing predominantly in one channel but then moving (without much artistic criterion) to the other channel after 10 or 20 s. Handling temporal consistency is a transversal problem across all generative models, typically handled by brute force (that is, more receptive field and/or larger models) or by some form of hierarchical or recurrent processing. Nonetheless, it is still an open issue, especially in the case of really long sequences like audio and music.

In addition to the limitations inherent to the technology, there are also some shortcomings in the test methodology. The subjective tests were conducted using headphones, whereas stereo images are typically created and mixed in a studio using professional loudspeaker monitoring. This implies that when critically evaluating the proposed approaches on a professional setup, additional subtleties might be discernible. Another methodological challenge was that often users had difficulty in evaluating multiple test excerpts according to the stated evaluation criteria. A potentially contributing factor to it was the absence of a standardized test methodology for multiple preference testing without a reference.

7. CONCLUSION

In this work we study methods to convert from mono to stereo. Our proposal entails (i) the use of PS for mono to stereo upmixing and (ii) the synthesis of PS parameters with three machine learning methods. We also introduce (iii) the use of modern generative approaches to the task and propose two variants of them. We additionally (iv) overview and adapt an existing PS methodology and (v) propose two tentative objective metrics to evaluate stereo renderings. The three proposed approaches outperform the classical and the deep neural network baselines we consider, and two of such approaches stand out with a statistically significant difference in the subjective test.

8. ACKNOWLEDGMENTS

We thank all the participants of the listening test for their input and Gautam Bhattacharya and Samuel Narváez for preliminary discussions on the topic.

9. REFERENCES

[1] M. R. Schroeder, "An artificial stereophonic effect obtained from a single audio signal," *Journal of the Audio Engineering Society*, vol. 6, no. 2, p. 74–79, 1958.

[2] B. B. Bauer, "Some techniques toward better stereophonic perspective," *IEEE Trans. on Audio*, vol. 11, p. 88–92, 1963.

[3] R. Orban, "A rational technique for synthesizing pseudo-stereo from monophonic sources," *Journal of the Audio Engineering Society*, vol. 18, no. 2, p. 157–164, 1970.

[4] C. Faller, "Pseudostereophony revisited," in *Proc. of the Audio Engineering Society Conv. (AES)*, 2005, p. 118.

[5] M. Fink, S. Kraft, and U. Zölzer, "Downmix-compatible conversion from mono to stereo in time- and frequency-domain," in *Proc. of the Int. Conf. on Digital Audio Effects (DAFx)*, 2015.

[6] C. Uhle and P. Gampp, "Mono-to-stereo upmixing," in *Proc. of the Audio Engineering Society Conv. (AES)*, 2016, p. 140.

[7] M. Lagrange, L. G. Martins, and G. Tzanetakis, "Semi-automatic mono to stereo up-mixing using sound source formation," in *Proc. of the Audio Engineering Society Conv. (AES)*, 2007, p. 122.

[8] D. Fitzgerald, "Upmixing from mono - A source separation approach," in *Proc. of the Int. Conf. on Digital Signal Processing (DSP)*, 2011.

[9] A. Delgado Castro and J. Szymanski, "Semi-automatic mono-to-stereo upmixing via separation of note events," in *Proc. of the AES Conf. on Immersive and Interactive Audio*, 2019, p. 12.

[10] J. Pons, S. Pascual, G. Cengarle, and J. Serrà, "Upsampling artifacts in neural audio synthesis," in *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, p. 3005–3009.

[11] 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, 2018.

[12] C. J. Steinmetz, J. Pons, S. Pascual, and J. Serrà, "Automatic multitrack mixing with a differentiable mixing console of neural audio effects," in *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, p. 7175.

[13] D. Gibson, *The art of mixing: a visual guide to recording, engineering, and production*, 2nd ed. Fairview, USA: ArtistPRO, 2005.

[14] J. M. Tomczak, *Deep generative modeling*. New York, USA: Springer Charm, 2022.

[15] C. J. Chun, S. H. Jeong, S. Y. Park, and H. K. Kim, "Extension of monaural to stereophonic sound based on deep neural networks," in *Proc. of the Audio Engineering Society Conv. (AES)*, 2015, p. 139.

[16] H. Purnhagen, "Low complexity parametric stereo coding in MPEG-4," in *Proc. of the Int. Conf. on Digital Audio Effects (DAFx)*, 2004, p. 163–168.

[17] J. Breebaart, S. van de Par, A. Kohlrausch, and E. Schuijers, "Parametric coding of stereo audio," *EURASIP Journal on Advances in Signal Processing*, vol. 2005, no. 9, p. 1305–1322, 2005.

[18] T. Hastie and R. Tibshirani, *The elements of statistical learning: data mining, inference, and prediction*, 2nd ed. New York, USA: Springer, 2009.

[19] A. Radford, K. Narashiman, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," *Technical Report, OpenAI*, 2018.

[20] H. Chang, H. Zhang, L. Jiang, C. Liu, and W. T. Freeman, "MaskGIT: masked generative image transformer," in *Proc. of the IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022, p. 11315–11325.

[21] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural Computation*, vol. 1, no. 2, p. 270–280, 1989.

[22] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, *PyTorch: an imperative style, high-performance deep learning library*. Curran Associates, Inc., 2019, vol. 32, p. 8024–8035.

[23] B. Bartlett, *Stereo microphone techniques*. London, UK: Focal Press, 1991.

[24] X. I. Chen, N. Mishra, M. Rohaninejad, and P. Abbeel, "PixelSNAIL: an improved autoregressive generative model," in *Proc. of the Int. Conf. on Machine learning (ICML)*, 2018, p. 864–872.

[25] J. Ho and T. Salimans, "Classifier-free diffusion guidance," in *Proc. of the NeurIPS Workshop on Deep Generative Models and Downstream Applications*, 2021.

[26] H. Chang, H. Zhang, J. Barber, A. J. Maschinot, J. Lezama, L. Jiang, M.-H. Yang, K. Murphy, W. T. Freeman, M. Rubinstein, Y. Li, and D. Krishnan, "Muse: text-to-image generation via masked generative transformers," *ArXiv: 2301.00704*, 2023.

FROM WEST TO EAST: WHO CAN UNDERSTAND THE MUSIC OF THE OTHERS BETTER?

Charilaos Papaioannou[1,2] **Emmanouil Benetos**[2] **Alexandros Potamianos**[1]
[1] School of ECE, National Technical University of Athens, Greece
[2] Centre for Digital Music, Queen Mary University of London, UK
cpapaioan@mail.ntua.gr

ABSTRACT

Recent developments in MIR have led to several benchmark deep learning models whose embeddings can be used for a variety of downstream tasks. At the same time, the vast majority of these models have been trained on Western pop/rock music and related styles. This leads to research questions on whether these models can be used to learn representations for different music cultures and styles, or whether we can build similar music audio embedding models trained on data from different cultures or styles. To that end, we leverage transfer learning methods to derive insights about the similarities between the different music cultures to which the data belongs to. We use two Western music datasets, two traditional/folk datasets coming from eastern Mediterranean cultures, and two datasets belonging to Indian art music. Three deep audio embedding models are trained and transferred across domains, including two CNN-based and a Transformer-based architecture, to perform auto-tagging for each target domain dataset. Experimental results show that competitive performance is achieved in all domains via transfer learning, while the best source dataset varies for each music culture. The implementation and the trained models are both provided in a public repository.

1. INTRODUCTION

As the time passes by, more and more pre-trained models are being made available in the MIR field. These models can be used in a variety of tasks by providing informative deep audio embeddings for music pieces. In correspondence with publicly available datasets, the vast majority of these models are trained on the so called "Western" [1] musical tradition [1]. While studying world, folk, or traditional music, that fact arises two research questions; on the one hand what is the potential of these models when they are being used in the realm of a different culture and on the other hand how capable can a model be when trained on a specific music tradition on providing meaningful audio embeddings.

There are several experimental setups one can employ in order to derive answers to the above questions. By taking into account the importance of the auto-tagging task in the MIR field [2], it becomes clear that transferring knowledge between domain-specific models to perform this task may lead us to valuable insights. Automatic content-based tagging aims to predict the tags of a music piece given its audio signal. The audio signal includes the acoustic characteristics and some of them are responsible for the occurrence of a tag in a piece, forming a multiple instance problem [3].

A variety of models have been proposed to cope with the automatic tagging of music pieces. They can be divided, according to the input data they process, into the ones that utilize time-frequency representations and the others that accept the raw audio signal. In the first category, CNN-based models which are adopted by the computer vision field can be found, such as VGG-ish [4] as well as specifically developed architectures for music, like Musicnn [5]. A Transformer-based architecture was recently proposed in [6] called Audio Spectrogram Transformer (AST). Regarding the models that process audio, the TCNN [7] and the Wave-U-Net [8] architectures are being commonly used. For the purposes of our study, it is essential to use models of the same category with respect to the input they accept and, thus, we selected the ones that process time-frequency representations because of their popularity in the MIR field.

While using deep neural networks, transfer learning of a trained model can lead to a significant performance improvement on the target domain, compared to one that starts from a random state in the parameters space [9]. Typically, the weights of the target domain model are initialized with the ones of a pre-trained model and then fine-tuning is applied. During this step, one has to determine which of the layers will be trainable and which ones will be kept frozen [10]. In general, it is not clear which part of the network should be allowed to be trained in the target task and, thus, experimentation with different setups is necessary. Standard methods include the fine-tuning of the whole network, as suggested in [11], as well as only the last few layers or a part of the network, as in [12]. We

[1] we use the term "Western" to denote music styles which mostly originate from Western cultures, including pop, rock, and Western classical.

© C. Papaioannou, E. Benetos, and A. Potamianos. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** C. Papaioannou, E. Benetos, and A. Potamianos, "From West to East: Who can understand the music of the others better?", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

experiment with both setups to derive valuable insights on knowledge transfer across domains.

Even though under-represented in general, datasets from specific music cultures are evident in the MIR field and a set of the aforementioned methods have been used to perform several tasks. In [13] a classification of Indian art music was conducted using deep learning models while automatic makam recognition in Turkish music was carried out in [14, 15]. With respect to Western music, there are several research works performing auto-tagging via deep learning models, as in [16] and [17].

In this paper, we incorporate a mosaic of different cultures by including six datasets from Western to Mediterranean and Indian music. Three music audio embedding models, two that mainly consist of convolutional layers and a Transformer-based architecture, are utilized on both single-domain and transfer learning experimental setups for music tagging. Results indicate that any model, despite the music culture that it is trained on, has the potential to adapt to another and achieve competitive results. When comparing the contributions of cross-domain knowledge transfers, we notice that they vary for each music culture and we suggest which one is the best candidate to outperform the single-domain approach. To the authors' knowledge, this is the first study which attempts to explore whether existing music audio embedding models can be used to transfer or learn representations for non-Western cultures. For reproducibility, we share the implementation in a public repository [2].

2. DATASETS

The selection of the datasets is a prominent theme in the current study and it is constrained by the available corpora that reflect different music cultures. By basing our intuition on the location of each culture, we pursue to include three distinct geographic regions each one represented by two corpora.

Even though spread in several continents, we consider the "West" as a single entity and utilize the MagnaTagATune [18] and FMA-medium [19] datasets that mainly belong to this culture. The second region is the eastern Mediterranean represented by the traditions of Greece and Turkey in our study with Lyra [20] and Turkish-makam [21] datasets. The Indian subcontinent is also incorporated with Hindustani and Carnatic corpora [22], corresponding to the music traditions of the Northern and Southern areas of India respectively.

2.1 MagnaTagATune

MagnaTagATune [18] is a publicly available dataset that is commonly used for the auto-tagging problem in the MIR field. It consists of more than 25,000 audio recordings, summing to 210 hours of audio content at total. Each audio recording is annotated with a subset of the unique 188 tags. Typically, only the top 50 most popular tags are used, which include annotations about genre, instruments and mood. In Table 1, the most frequent tags for MagnaTagATune are presented along with the ones of the other datasets.

2.2 FMA-medium

The Free Music Archive [19] is an open and easily accessible dataset that is used for evaluating several tasks. It contains over 100,000 tracks which are arranged in a hierarchical taxonomy of 161 genres. In order to keep the durations of the datasets balanced whenever possible, and to include genres belonging to Western music styles, we use FMA-medium that consist of 25,000 tracks of 30 seconds each. That means that its total duration is 208 hours, almost equal to the one of MagnaTagATune. With regards to the metadata, we include the top-20 hierarchically related genres of the music pieces.

2.3 Lyra

Lyra [20] is a dataset for Greek traditional and folk music that comprises 1570 pieces and metadata information with regards to instrumentation, geography and genre. Its total duration is 80 hours which makes it the only dataset with duration less than 200 hours in our study. We incorporate the top-30 tags retrieved from columns "genre", "place" and "instruments" to form our multi-label classification setup.

2.4 Turkish-makam

The Turkish makam corpus [21, 23] includes thousands of audio recordings covering more than 2,000 works from hundreds of artists. It is part of CompMusic Corpora [3] [24] which comprises data collections that have been created with the aim of studying particular music traditions. Using Dunya [25] and the related software tool [4], we were able to get access to 5297 audio recordings, summing in 359 hours, along with their metadata. In order to keep the dataset sizes similar, we set a maximum audio duration equal to 150 seconds which reduced the total length to 215 hours. For the tags, the top-30 most popular with regards to "makam", "usul" and "instruments" information have been included.

2.5 Hindustani

The Hindustani corpus [22] is also part of CompMusic Corpora. It includes 1204 audio recordings, with a total duration of 343 hours, covering a plethora of artists and metadata categories. By setting the maximum audio duration to 780 seconds, the size of the dataset has been decreased to 206 hours for the needs of our study. Furthermore, information about "raga", "tala", "instruments" and "form" has been used to form the labels of each piece. The top-20 most frequent tags have been incorporated to our study as the target of the classification models.

[2] https://github.com/pxaris/ccml

[3] https://compmusic.upf.edu/corpora

[4] https://github.com/MTG/pycompmusic

MagnaTagATune		FMA-medium		Lyra		Turkish-makam		Hindustani		Carnatic	
guitar	18.76%	Rock	28.41%	Voice	76.21%	Voice	63.33%	Voice	83.90%	Voice	82.35%
classical	16.52%	Electronic	25.26%	Traditional	76.05%	Kanun	31.09%	Tabla	53.03%	Violin	78.45%
slow	13.71%	Punk	13.28%	Violin	57.34%	Tanbur	27.93%	Khayal	41.33%	Mridangam	75.65%
techno	11.42%	Experimental	9.00%	Percussion	53.71%	Ney	27.56%	Harmonium	39.25%	Kriti	70.87%
strings	10.55%	Hip-Hop	8.80%	Laouto	51.69%	orchestra	26.38%	Teentaal	35.35%	adi	51.88%
drums	10.05%	Folk	6.08%	Guitar	37.34%	Oud	24.36%	Tambura	27.88%	Ghatam	30.32%
electronic	9.74%	Garage	5.67%	Klarino	31.05%	kemence	22.79%	Ektaal	21.58%	Khanjira	17.65%
rock	9.17%	Instrumental	5.40%	Nisiotiko	26.85%	Cello	17.83%	Pakhavaj	7.88%	rupaka	11.98%
fast	8.92%	Indie-Rock	5.17%	place-None	25.16%	Violin	17.62%	Sarangi	7.30%	mishra chapu	7.27%
piano	7.95%	Pop	4.74%	Bass	24.76%	Hicaz	10.63%	Dhrupad	7.05%	Tana Varnam	5.21%

Table 1. Relative frequencies of the top 10 most popular tags in each dataset.

2.6 Carnatic

The Carnatic corpus [22] comprises 2612 audio recordings, summing in more than 500 hours of content. As with the previous datasets, by setting a maximum duration cut equal to 330 seconds, the total duration has been decreased to 218 hours. Identical to Hindustani, the top-20 most popular annotations regarding "raga", "tala", "instruments" and "form" have been included for the metadata.

3. METHOD

In this section, the models which are used for the purposes of this study are first presented. We, then, describe how transfer learning is utilized to infer similarities between the music cultures by employing knowledge from the domain adaptation field.

3.1 Models

3.1.1 VGG-ish

All of our models use the mel-spectrogram as their input, a commonly used feature for MIR tasks such as automatic tagging [26]. This selection enables the utilization of CNN-based architectures which have been successfully used in computer vision tasks. The Visual Geometry Group (VGG) network [27] and its variants consist of a stack of convolutional layers followed by fully connected layers.

We use a VGG-ish architecture, similar to the one implemented by the authors in [28], that is a 7-layer CNN, with 3×3 convolution filters and 2×2 max-pooling, followed by two fully-connected layers. It accepts mel-spectrograms that correspond to short chunks of audio as its input, with duration equal to 3.69 seconds.

3.1.2 Musicnn

Musicnn [17] is a music inspired model that uses convolutional layers at its core. Its first convolutional layer consists of vertical and horizontal filters in order to capture timbral and temporal features respectively. These features are, then, concatenated and fed to 1D convolutional layers followed by a pair of dense layers that summarize them and predict the relevant tags. Similar to VGG-ish, it uses mel spectrograms from short audio chunks at its input with duration 3 seconds.

3.1.3 Audio Spectrogram Transformer

As its name indicates, Audio Spectrogram Transformer (AST) is a purely attention-based model for audio classification [6]. Based on the Transformer architecture [29], AST splits the input mel-spectrogram to 16×16 patches in both time and frequency dimensions that are, in turn, flattened to 1D embeddings of size 768 using a linear projection layer. A trainable positional embedding is also added to each patch embedding so that the model will capture the spatial structure of the input 2D spectrogram. The resulting sequence is fed to the Transformer, where only the encoder is utilized since AST is designed for classification tasks. The output of the encoder is followed by a linear layer that predicts the labels. As the authors that introduced the architecture suggest, we set a specific cut to the input length of the AST model that is equal to 8 seconds in all our experiments.

3.2 Transfer Learning

The purpose of transfer learning is to improve the performance of the models on target domains by transferring knowledge from different but related source domains [30]. In the field of MIR, both transferring feature representations to the target domain from a pre-trained model on a source task [31] as well as learning shared latent representations across domains [32] have been proposed in the past. Yet, these methods have not been applied to non-Western music datasets neither by adapting an existing model to them nor by studying to what end these cultures can be valuable source domains for widely developed models, two aspects which are both studied in this work.

According to the categorization conducted by the authors in [33], these methods belong to *parameter sharing* category of the model-based transfer learning techniques. In the deep learning realm, it is, thus, common to use a trained network for a source task, share its parameters and in turn fine-tune some or all layers to produce a target network. While following this method, one expects it to lead to better results when the participating domains are similar

Model	**VGG-ish**		**Musicnn**		**AST**	
Metric / Dataset	ROC-AUC	PR-AUC	ROC-AUC	PR-AUC	ROC-AUC	PR-AUC
MagnaTagATune	0.9123	0.4582	0.9019	0.4333	**0.9172**	**0.4654**
FMA-medium	**0.8889**	0.4949	0.8766	0.4473	0.8886	**0.5024**
Lyra	0.8097	0.4806	0.7391	0.4042	**0.8476**	**0.5333**
Turkish-makam	**0.8696**	0.5639	0.8505	0.5299	0.8643	**0.5669**
Hindustani	**0.8477**	**0.6082**	0.8471	0.6016	0.8307	0.5786
Carnatic	0.7392	0.4278	0.7496	0.4182	**0.7706**	**0.4394**

Table 2. ROC-AUC and PR-AUC scores of the models on single domain auto-tagging tasks.

to each other. Indeed, by studying the prior work on domain adaptation, one will find that the main strategy consists of minimizing the difference between the source and target feature distributions, when transferring representations from a labeled dataset to a target domain where labeled data is sparse or non-existent [34, 35].

By adapting the above rationale to our study, where the participating domains are all rich in labeled data, we expect that when applying transfer learning by parameter sharing, the more the similarity between the participating domains the better the performance of the target domain on its supervised learning task.

In order to study to what end this hypothesis stands in computational musicology with deep neural networks, we utilize the previously presented models which are widely used in the MIR field and consist of different cores, namely convolutional layers (VGG-ish and Musicnn) and a Transformer module (AST). Having the models trained on each single dataset, we apply all the cross-domain knowledge transfers for each architecture by fine-tuning only the output layer as well as the whole network. We then aggregate the results across the models seeking to derive insights with regards to the similarities between the domains as well as specifying which source is the best candidate for each target dataset.

4. EXPERIMENTS

As already mentioned, we use mel spectrograms as the input of all our models. In order to convert the audio recordings of the datasets to this representation, we use Librosa [36] to re-sample them to 16 kHz sample rate. Then, 512-point FFT with a 50% overlap is applied, the maximum frequency is set to 8 kHz and number of Mel bands to 128. Our intention, in this study, is not the optimization of the performance of the single-domain tasks but rather studying the knowledge transfer across the domains. So, we keep our training setup as close as possible to the literature, at each single domain task, in order to have a sanity check for the implementation.

For VGG-ish and Musicnn models, we use a mixture of scheduled Adam [37] and stochastic gradient descent (SGD) for the optimization method, identical to what the authors at [28] have used. The batch size is set to 16 and the learning rate to $1e-4$ for both models while the maximum number of epochs are 200 for VGG-ish and 50 for Musicnn. With regards to the AST model, we follow the setup proposed in [6], namely batch size 12, Adam optimizer, learning rate scheduling that begins from $1e-5$ and is decreased by a factor of 0.85 every epoch after the 5th one as well as pre-trained on Imagenet Transformer weights.

All our models accept a fixed size audio chunk at their input but need to predict song-level tags. During the evaluation phase, we aggregate the tag scores across all chunks by averaging them to acquire the label scores for the whole audio. We use the area under receiver operating characteristic curve (ROC-AUC), a widely used evaluation metric on multi-label classification problems and the area under precision-recall curve (PR-AUC), a suitable metric for unbalanced datasets [38].

During transfer learning, we initialize all parameters of the target model, except for the output layer, from each source dataset and (i) allow only the output layer to be trained and (ii) train the whole network. In both settings, we use the same hyper-parameters and evaluation procedure with the single-domain setups across all datasets for each model architecture.

5. RESULTS

The performance of the three models on all single-domain tasks can be seen in Table 2. The performance of the Musicnn and VGG-ish models on MagnaTagATune is similar to the reported metrics in [28], which indicates the validity of our implementation. In general, the AST model shows the best performance followed by VGG-ish and then Musicnn. This result should not be taken into account solidly, because no hyper-parameter tuning has been taken place for each domain and in order to keep the duration of the training to less than 24 hours for each task, the number of epochs for Musicnn was significantly less than VGG-ish. On the other hand, one should consider that the AST [6] and VGG-ish [28] models may, indeed, perform better for limited time resources.

In Table 3, one can see the ROC-AUC scores in all single-domain and cross-domain setups. The rows are the source datasets while the columns are the target datasets. A sub-table is constructed for each model architecture and for a transfer from domain *A* to *B*, the result of the fine-tuning of only the output layer ('output') as well as all the layers ('all') are reported. The single-domain setup is

Target domain	**MagnaTagATune**		**FMA-medium**		**Lyra**		**Turkish-makam**		**Hindustani**		**Carnatic**	
trainable layer(s) / Source domain	output	all	output	all	output	all	output	all	output	all	output	all
VGG-ish												
MagnaTagATune	-	91.23	88.11	**92.39**	74.69	**85.40**	76.79	86.84	76.09	85.04	67.19	**74.71**
FMA-medium	85.82	**91.29**	-	88.89	68.56	84.04	75.40	**87.78**	75.77	84.39	67.03	74.56
Lyra	84.34	90.93	82.84	92.10	-	80.97	76.98	87.21	77.41	84.24	67.30	73.52
Turkish-makam	85.19	90.90	84.41	91.74	70.93	82.38	-	86.96	77.54	**85.32**	67.16	73.50
Hindustani	84.24	91.02	83.83	91.91	66.27	79.71	77.25	87.63	-	84.77	66.72	74.63
Carnatic	84.18	91.00	82.62	91.73	61.59	76.72	77.07	87.40	78.19	84.81	-	73.92
Musicnn												
MagnaTagATune	-	90.19	87.34	**91.03**	71.79	78.74	74.72	**85.96**	75.87	84.18	66.12	75.57
FMA-medium	85.52	**90.35**	-	87.66	65.94	77.59	75.51	85.13	73.16	**85.49**	66.38	75.77
Lyra	81.38	90.03	82.23	90.80	-	73.91	74.11	85.20	78.10	83.29	65.09	75.51
Turkish-makam	84.35	90.11	83.79	90.81	61.87	**79.83**	-	85.05	75.67	83.75	67.49	74.09
Hindustani	82.38	89.86	83.42	90.85	64.48	78.95	74.60	85.58	-	84.71	65.25	**76.95**
Carnatic	83.02	90.05	82.78	90.74	61.83	77.92	75.09	85.43	75.34	84.19	-	74.96
AST												
MagnaTagATune	-	**91.72**	89.25	91.99	75.68	83.77	76.28	87.20	74.67	**86.57**	66.03	75.43
FMA-medium	88.63	91.62	-	88.86	65.72	82.17	76.37	**87.43**	74.51	85.76	67.33	75.98
Lyra	87.49	91.44	87.44	**92.43**	-	**84.76**	77.08	86.80	72.24	83.73	68.47	76.59
Turkish-makam	87.33	91.40	86.31	91.95	72.70	77.95	-	86.43	70.13	83.56	67.10	75.23
Hindustani	87.40	91.35	87.11	92.26	71.74	84.60	75.70	86.90	-	83.07	67.75	75.85
Carnatic	87.42	91.45	86.83	91.75	63.33	81.44	76.87	87.14	74.11	82.91	-	**77.06**

Table 3. ROC-AUC scores (%) when applying transfer learning using the models VGG-ish, Musicnn and Audio Spectrogram Transformer. Rows are the source domains and columns the target domains. After initializing the network with the parameters of the trained (at the source dataset) model, fine-tuning on the output layer as well as on the whole network is applied. The diagonal values (under the "all" columns) correspond to the respective single-domain models (no transfer learning) where the experimentation with only the output layer trainable has no meaning.

when source and target is the same dataset and, thus, only training of the whole network has meaning. The table is better parsed column-wise, e.g., by inspecting the results of VGG-ish model on MagnaTagATune when transferring knowledge from the other domains at the upper-left pair of columns in the table.

In order to aggregate all the cross-domain knowledge transfers, we follow the subsequent procedure: for each target task that consists of a specific model, target dataset and fine-tuning method, min-max normalization is applied to the $N-1$ transfer learning results, where N is the number of all datasets. The previous step leads to the construction of $M \times F$ matrices, M the number of the models and F the number of fine-tuning methods, where rows are the source domains, columns the target domains and diagonal elements are empty. Each cell has a value in the range $[0, 1]$, as a result of the normalization step, while the value 1 corresponds to the knowledge transfer that led to the best performance in the target domain. By calculating the element-wise mean of the produced $M \times F$ matrices, we reach to the result that can be seen in Figure 2.

6. DISCUSSION

The results indicate that knowledge transfer both from Western to non-Western cultures and the opposite can be beneficial when deep learning models are used to perform automatic music tagging. Indeed, by inspecting Table 3, the general take-home message one should acquire is that regardless of the model architecture, all datasets have the potential to contribute as a source to a target domain by providing their deep audio embeddings. To investigate how valuable knowledge transfers from widely used datasets to non-Western music cultures can be, we focus on the last four datasets, i.e., the last eight columns of the table, and parse the two first rows, corresponding to MagnaTagATune and FMA datasets, at each model architecture. For instance, we notice that for Lyra, when Musicnn is used and fine-tuning only of the output layer is applied, the model coming from MagnaTagATune has the greater ROC-AUC score, namely 71.79%. Additionally, the AST model trained on the FMA-medium dataset, outperforms the others when totally fine-tuned to the Turkish-makam dataset, scoring 87.43%.

In order to study the inverse transfer direction, we center our interest to the first four columns of the entire table. Even though MagnaTagATune and FMA are almost always the best source for each other, the deep audio embeddings provided by the other datasets achieve competitive performance. For example, when MagnaTagATune is the target domain and fine-tuning is restricted to the output layer of the network, we observe that transferring from Turkish-

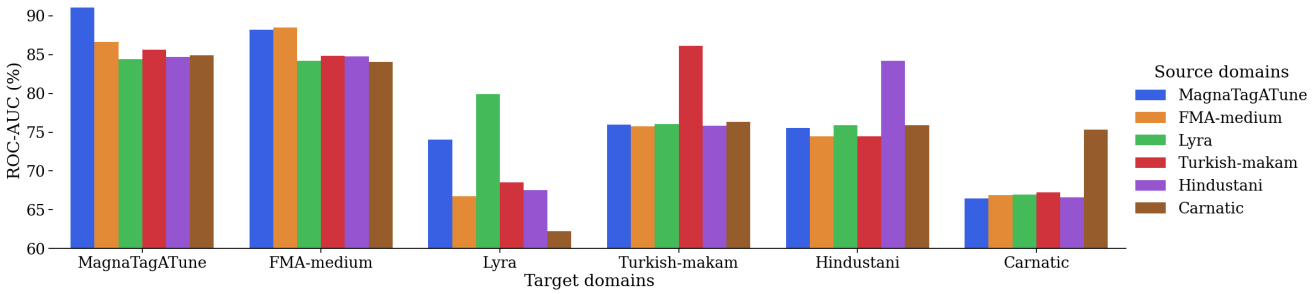


Figure 1. Average, over the three models, ROC-AUC scores of all cross-domain transfers when fine-tuning of the output layer is applied. The highest bar at each group corresponds to the respective single-domain model.

	MagnaTag-ATune	FMA-medium	Lyra	Turkish-makam	Hindustani	Carnatic
MagnaTag-ATune	—	0.89	0.9	0.54	0.64	0.49
FMA-medium	1.0	—	0.44	0.59	0.48	0.6
Lyra	0.17	0.37	—	0.39	0.39	0.59
Turkish-makam	0.35	0.19	0.52	—	0.44	0.37
Hindustani	0.11	0.36	0.55	0.49	—	0.53
Carnatic	0.25	0.05	0.11	0.66	0.54	—

Figure 2. Cross-cultural music transfer learning results. Rows correspond to the source datasets and columns to the target datasets. The value of each cell (knowledge transfer) is normalized and averaged across all models and fine-tuning methods.

makam leads to a performance that is comparable to the best source (FMA-medium) for all models.

By considering all cross-domain knowledge transfers, one can specify the best candidate to provide a trained model, with a specific architecture, for each target dataset. We, thus, notice that the model that is transferred from Hindustani outperforms the others at the Carnatic dataset, when fine-tuning on the whole Musicnn architecture is applied. A holistic picture of the cross-cultural music transfer learning is depicted in Figures 1 and 2.

In Fig. 1 the scores of all cross-domain transfers when fine-tuning the output layer, can be seen, averaged across the three models. The uniformity of the performances of different sources at each target dataset can be examined. We, thus, recognize that the most unbalanced performances are spotted on the Lyra target domain, a result that is probably related to the smaller size of this dataset compared to the others. By exploring Fig. 2 in a column-wise fashion, we observe that for MagnaTagATune as the target domain, FMA-medium is the best source with a value equal to 1. This means that in all transfer learning setups, this source performed better than the others in this domain.

Both figures show that MagnaTagATune and FMA-medium perform consistently well across the domains, something that possibly indicates their appropriateness for the auto-tagging task. However, as we move to the Eastern cultures, we notice that their contribution is somehow decreased and other domains tend to contribute similarly or even more in those targets. The values at Fig. 2 should not be considered solidly as similarity metrics between the domains because other factors may also affect the results we notice. It is, although, a first step towards studying different music cultures using deep learning methods.

7. CONCLUSIONS

In this paper, the transferrability of music cultures by utilizing deep audio embedding models is studied. To that end, six datasets and three models were employed while experimentation with two fine-tuning methods took place. The automatic tagging of music pieces served as the supervised learning task where all cross-domain knowledge transfers were applied and evaluated.

The results show that state-of-the-art models can benefit from knowledge transfer not only from Western to non-Western cultures but also the opposite too. By aggregating the scores across all models and fine-tuning methods, the suitability of each source domain for a target task was calculated and, thus, which domain can be the best candidate to transfer knowledge from for each dataset was proposed. Based on the literature, we suggest that this result can be interpreted to a degree as a similarity metric between the music cultures.

We identify that the current study has limitations. In the future, the semantic similarities between the labels of the involved domains will be examined. More datasets and models, like those that process raw audio signals, will be considered as well as semi-supervised and unsupervised learning techniques. Other tasks may be employed such as mode estimation, assuming that key in Western cultures functions in a similar way with makam or raga in other cultures. All datasets can also be utilized to learn music embeddings in order to unveil cross-cultural links between acoustic features and tags.

8. ACKNOWLEDGEMENTS

The authors would like to thank Sertan Şentürk, Alastair Porter and the Universitat Pompeu Fabra for their willingness to provide us with the data without which this study would not have been possible. We would like to also thank Charalampos Saitis and the reviewers for their valuable and constructive comments that helped us improve our work.

9. REFERENCES

[1] E. Gómez, P. Herrera, and F. Gómez-Martin, "Computational Ethnomusicology: perspectives and challenges," *Journal of New Music Research*, vol. 42, no. 2, pp. 111–112, June 2013.

[2] K. Choi, "Deep Neural Networks for Music Tagging," Ph.D. dissertation, Queen Mary University of London, September 2018.

[3] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez, "Solving the multiple instance problem with axis-parallel rectangles," *Artificial Intelligence*, vol. 89, no. 1, pp. 31–71, January 1997.

[4] 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," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2017, pp. 131–135.

[5] J. Pons and X. Serra, "musicnn: Pre-trained convolutional neural networks for music audio tagging," *arXiv preprint arXiv:1909.06654*, September 2019.

[6] Y. Gong, Y.-A. Chung, and J. Glass, "AST: Audio Spectrogram Transformer," *arXiv preprint arXiv:2104.01778*, July 2021.

[7] A. Pandey and D. Wang, "TCNN: Temporal Convolutional Neural Network for Real-time Speech Enhancement in the Time Domain," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019, pp. 6875–6879.

[8] 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*, June 2018.

[9] Z. Yang, R. Salakhutdinov, and W. W. Cohen, "Transfer Learning for Sequence Tagging with Hierarchical Recurrent Networks," *arXiv preprint arXiv:1703.06345*, March 2017.

[10] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" *arXiv preprint arXiv:1411.1792*, November 2014.

[11] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *arXiv preprint arXiv:1311.2524*, October 2014.

[12] M. Long, Y. Cao, J. Wang, and M. I. Jordan, "Learning Transferable Features with Deep Adaptation Networks," *arXiv preprint arXiv:1502.02791*, May 2015.

[13] A. K. Sharma, G. Aggarwal, S. Bhardwaj, P. Chakrabarti, T. Chakrabarti, J. H. Abawajy, S. Bhattacharyya, R. Mishra, A. Das, and H. Mahdin, "Classification of Indian Classical Music With Time-Series Matching Deep Learning Approach," *IEEE Access*, pp. 102 041–102 052, 2021.

[14] E. Demirel, B. Bozkurt, and X. Serra, "Automatic makam recognition using chroma features," in *Proceedings of the 8th International Workshop on Folk Music Analysis; Thessaloniki, Greece, p. 19-24*, 2018.

[15] K. K. Ganguli, S. Şentürk, and C. Guedes, "Critiquing task-versus goal-oriented approaches: A case for makam recognition," in *Proceedings of the 23rd Int. Society for Music Information Retrieval Conf., Bengaluru, India*, December 2022.

[16] K. Choi, G. Fazekas, and M. Sandler, "Automatic tagging using deep convolutional neural networks," *arXiv preprint arXiv:1606.00298*, June 2016.

[17] J. Pons, O. Nieto, M. Prockup, E. Schmidt, A. Ehmann, and X. Serra, "End-to-end learning for music audio tagging at scale," *arXiv preprint arXiv:1711.02520*, June 2018.

[18] E. Law, K. West, M. Mandel, M. Bay, and J. S. Downie, "Evaluation of algorithms using games: The case of music tagging," in *10th International Society for Music Information Retrieval Conference, ISMIR 2009*, 2009, pp. 387–392.

[19] M. Defferrard, K. Benzi, P. Vandergheynst, and X. Bresson, "FMA: A Dataset For Music Analysis," *arXiv preprint arXiv:1612.01840*, September 2017.

[20] 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 Int. Society for Music Information Retrieval Conf., Bengaluru, India*, December 2022.

[21] B. Uyar, H. S. Atli, S. Şentürk, B. Bozkurt, and X. Serra, "A corpus for computational research of turkish makam music," in *Proceedings of the 1st International Workshop on Digital Libraries for Musicology*, 2014, pp. 1–7.

[22] A. Srinivasamurthy, G. K. Koduri, S. Gulati, V. Ishwar, and X. Serra, "Corpora for music information research in indian art music," in *Proceedings of the 2014 International Computer Music Conference, ICMC/SMC; 2014 Sept 14-20; Athens, Greece*, 2014.

[23] S. Şentürk, "Computational analysis of audio recordings and music scores for the description and discovery of ottoman-turkish makam music," Ph.D. dissertation, Universitat Pompeu Fabra, 2016.

[24] X. Serra, "Creating research corpora for the computational study of music: the case of the compmusic project," in *Audio engineering society conference: 53rd international conference: Semantic audio*, 2014.

[25] A. Porter, M. Sordo, and X. Serra, "Dunya: A system for browsing audio music collections exploiting cultural context," in *Proceedings of the 14th Int. Society for Music Information Retrieval Conf., Curitiba, Brazil*, 2013.

[26] S. Dieleman and B. Schrauwen, "Multiscale approaches to music audio feature learning," in *14th International Society for Music Information Retrieval Conference (ISMIR-2013)*, 2013, pp. 116–121.

[27] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv preprint arXiv:1409.1556*, April 2015.

[28] M. Won, A. Ferraro, D. Bogdanov, and X. Serra, "Evaluation of CNN-based Automatic Music Tagging Models," *arXiv preprint arXiv:2006.00751*, June 2020.

[29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Kaiser, and I. Polosukhin, "Attention is All you Need," in *Advances in Neural Information Processing Systems*, 2017.

[30] S. J. Pan and Q. Yang, "A Survey on Transfer Learning," *IEEE Transactions on Knowledge and Data Engineering*, no. 10, pp. 1345–1359, October 2010.

[31] A. van den Oord, S. Dieleman, and B. Schrauwen, "Transfer learning by supervised pre-training for audio-based music classification," in *Conference of the International Society for Music Information Retrieval, Proceedings*, 2014.

[32] 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 *14th International Conference on Music Information Retrieval (ISMIR '13)*, 2013.

[33] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A Comprehensive Survey on Transfer Learning," *arXiv preprint arXiv:1911.02685*, June 2020.

[34] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko, "Simultaneous Deep Transfer Across Domains and Tasks," *arXiv preprint arXiv:1510.02192*, October 2015.

[35] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting Visual Category Models to New Domains," in *Computer Vision – ECCV 2010*. Springer, 2010, pp. 213–226.

[36] 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.

[37] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv preprint arXiv:1412.6980*, January 2017.

[38] J. Davis and M. Goadrich, "The relationship between precision-recall and roc curves," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 233–240.

ON THE PERFORMANCE OF OPTICAL MUSIC RECOGNITION IN THE ABSENCE OF SPECIFIC TRAINING DATA

Juan C. Martinez-Sevilla[1] **Adrian Rosello**[1]
David Rizo[1,2] **Jorge Calvo-Zaragoza**[1]
[1] U. I. for Computing Research, University of Alicante, Spain
[2] Instituto Superior de Enseñanzas Artísticas de la Comunidad Valenciana, Spain
adrian.rosello@ua.es, {jcmartinez, drizo, jcalvo}@dlsi.ua.es

ABSTRACT

Optical Music Recognition (OMR) has become a popular technology to retrieve information present in musical scores in conjunction with the increasing improvement of Deep Learning techniques, which represent the state-of-the-art in the field. However, its effectiveness is limited to cases where the target collection is similar in musical context and graphical appearance to the available training examples. To address this limitation, researchers have resorted to labeling examples for specific neural models, which is time-consuming and raises questions about usability. In this study, we propose a holistic and comprehensive study for dealing with new music collections in OMR, including extensive experiments to identify key aspects to have in mind that lead to better performance ratios. We resort to collections written in Mensural notation as a specific use case, comprising 5 different corpora of training domains and up to 15 test collections. Our experiments report many interesting insights that will be important to create a manual of best practices when dealing with new collections in OMR systems.

1. INTRODUCTION

Manual sheet music transcription is a tedious process, prone to errors, and generally requires professionals with precise knowledge of the type of notation and/or music at issue. The alternative to this manual digitization of content is to resort to cutting-edge technology based on artificial intelligence, which performs an automated reading of documents. This technology is known as Optical Music Recognition (OMR).

OMR has been an active research area for decades [1], although the field progressed slowly [2]. Recently, the use of modern machine learning techniques, namely Deep Learning, has led to a paradigm shift that has partially unlocked this situation [3, 4]. Indeed, it has been shown that current OMR technologies, despite the fact that they are not yet fully mature, are usually a better alternative than performing the entire transcription by hand [5].

Concerning the machine learning methods, the related literature reports that the models provide sufficient precision when the collections to be transcribed are from the same graphic and content domain as the corpus used to train them. This, however, makes it difficult to transfer technology to new collections, since it is not always possible, desirable, or efficient to invest resources in annotating a small portion of the target collection. Although it is naive to assume the availability of training sets from the same domain as a given target collection, in the current data era we can assume to have at least a series of labeled collections, even with different graphic and musical characteristics. This, of course, can and should be used to improve the efficiency of fitting OMR models to new collections for which we do not have specific training sets.

In this paper, we report on a case study focused on Mensural notation to answer questions about the transferability of OMR models to new music collections. To our best knowledge, this work constitutes the first to analyze this issue in the field. We consider Mensural notation as the structuring experimental body because the OMR technology can be considered mature for this notation. Also, we have a significant number of labeled and unlabeled collections in this notation, which allows us to carry out an exhaustive study that is expected to lead to more generalizable conclusions. Specifically, we consider 5 labeled collections that will be used as training sets, along with their possible combinations, and up to 15 unlabeled collections as target.

The rest of the paper is structured as it follows: in Section 2, we provide some background to the topic; in Section 3, we present our methodology to analyze the question at issue; the experimental setup is described in Section 4, while the results and analysis are given in Section 5; finally, we conclude the paper in Section 6, while pointing out some interesting avenues for future work.

2. BACKGROUND

Recent advances in artificial intelligence, with extensive use of Deep Learning (DL) technologies, resulted in about successful approaches to OMR. Specifically, a holistic ap-

© J. C. Martinez-Sevilla, A. Rosello, D. Rizo and J. Calvo-Zaragoza. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** 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 *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

proach, also known as end-to-end formulation, which has been dominating the state of the art in other applications such as text or speech recognition [6, 7], is currently considered the reference model in OMR. The related literature includes many successful solutions of this type [8–10]. In this work, we resort to this approach as representative of the state of the art based on DL.

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. The underlying issue is an overly unsolved challenge in artificial intelligence [11]: DL works well if the problem is statistically regular and there is abundant training data to adequately and representatively learn such regularity. This is, unfortunately, quite difficult to expect when dealing with ancient documents. Instead of trying to solve the underlying problem of machine learning, we take a more practical path to provide a series of best practices to tackle the situation of target collections in the absence of specific training data successfully.

It is important to highlight that, in the OMR literature, there are very few works dedicated to studying the practical aspects of the technology. Pugin and Crawford [12] estimated through a quantitative evaluation the suitability of using the Aruspix machine-learning-based OMR system on a real collection. Furthermore, Alfaro-Contreras et al. [5] analyzed the benefits of using OMR in cases where the accuracy of the system was not perfect. Our work further contributes to this barely explored line of practical aspects for the application of OMR to real-world scenarios from the perspective of the available training data.

3. METHODOLOGY

The focus of the work is essentially experimental. We want to be able to answer specific questions about how to approach the generation of generalizable OMR models. Our objective is to reduce the uncertainty when facing the recognition of collections for which there is no specific training set.

To answer these questions, we will consider as a starting point the availability of N training sets that, even depicting the same musical notation (Mensural notation), differ in graphic characteristics. This will allow drawing more interesting conclusions about the synergy of using a heterogeneous set of training collections. To cover all possibilities, we create models from all possible combinations of these sets ($2^N - 1$ possibilities). Each of these possibilities will be directly evaluated on M test sets (not seen in any training case), also showing heterogeneous characteristics.

As previously mentioned, we will consider a deep end-to-end model as representative of the state of the art in OMR. Below we explain in more detail how this model works.

3.1 Learning framework

For the task, a Convolutional Recurrent Neural Network (CRNN) scheme is proposed for the end-to-end optical music transcription pipeline. The CRNN architecture consists of a block of convolutional layers that learns the relevant features from the input image (single staff), followed by a 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 the posteriogram, which is decoded to obtain the predicted musical symbols. [1]

The Connectionist Temporal Classification (CTC) [13] training procedure is used to train the CRNN model using unsegmented sequential data. The training set $\mathcal{T}$ consists of pairs of single musical staff images x_i and their corresponding symbol sequence $\mathbf{z}_i$ in a symbol vocabulary Σ, with 261 units corresponding to the number of different symbols among the training sets. To use CTC as an end-to-end sequence labeling framework, an additional "blank" symbol is included in the vocabulary Σ'.

Formally, let $\mathcal{T} \subset \mathcal{X} \times \Sigma^*$ be a set of data where an image $x_i \in \mathcal{X}$ of a single staff is related to symbol sequence $\mathbf{z}_i = \left(z_{i1}, z_{i2}, \ldots, z_{i|\mathbf{z}_i|}\right) \in \Sigma^*$, where Σ represents the symbol vocabulary used for encoding the music score. Note that the use of CTC to model the transcription task as an end-to-end *sequence labeling* framework requires the inclusion of an additional "*blank*" symbol in the Σ vocabulary, i.e., $\Sigma' = \Sigma \cup \{blank\}$.

At prediction, for a given musical staff image input $x_i \in \mathcal{X}$, the model outputs a posteriogram $p_i \in \mathbb{R}^{|\Sigma'| \times K}$, where K represents the number of frames given 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.

4. EXPERIMENTAL SETUP

In this section, we present our choices for the experimental design. First, we describe the considered evaluation metric. Then, we give more implementation details of the deep learning model. Finally, we present and describe the collections selected as train and target sets.

4.1 Evaluation

To evaluate the performance of the OMR model, we resort to the *Symbol Error Rate* (SER). This is computed as the average number of elementary editing operations (insertions, deletions, or substitutions) required to convert prediction $\hat{\mathbf{z}}_i$ into reference $\mathbf{z}_i$, normalized by the length of the latter.

In general, we are interested in computing the amount of effort it would take for a person to correct the remaining errors in the system. Since computing this human effort

[1] Understanding *musical symbol* as the conjunction of `glyph:position`, *i.e.*, `note_half:L2` (a glyph `note_half` present in the second staff line).

does not scale well in practice (it consumes huge amounts of resources), we believe that this metric is suitable to measure the transcription correctness. In addition, it is a metric that has been commonly applied in previous works on this subject (cf. Section 2).

4.2 Neural model configuration

The CRNN topology is based on the one used in the research [14], where the authors adopt a 4 convolutional layer block with batch normalization, Leaky ReLu activation, and max-pooling down-sampling. The feature maps extracted from the convolutional block are fed into two Bidirectional Long Short-Time Memory layers with 256 hidden units each and a dropout value of $d = 50\%$ followed by a fully-connected network with $|\Sigma'|$ units.

The models were trained with a batch size of 16 elements—note that in experiments where multiple training sets were used all the generated batches in the training process were balanced so the net didn't adjust to a certain corpus. The ADAM optimizer [15] was considered and a fixed learning rate of 10^{-3}. We iterate for 300 epochs, keeping the weights that minimize the SER metric in the validation partition with an early stopping policy of 30 epochs. Finally, all experiments were run using the Python language (v. 3.8.13) with the PyTorch framework (v. 1.13.0) on a single NVIDIA GeForce RTX 4090 card with 24GB of GPU memory.

4.3 Datasets

A set of 20 different white Mensural notation works has been collected for this work, consisting of pairs of staff images and their transcription into sequences of musical symbols. The pieces have been selected looking for diverse cases concerning printers or copyists, layouts, authors, the period in history, and extension. [2]

4.3.1 Training Datasets

For training, 4 different datasets were chosen from real collections, trying to cover as much variability as possible. When facing a new transcription project, it is usual that no training collection is similar or big enough for building a model to obtain reliable results from the automatic recognition process. In this scenario, the creation of synthetic training data from scratch is a valid alternative that will be evaluated in the work with the PRIMENS dataset. Therefore, we will add this synthetic collection to the set of training sets, resulting in 5 different collections. These training collections are described below.

- **CAPITAN**. The Capitan dataset contains 100 handwritten pages of ca. 17th-century manuscripts in late white Mensural notation extracted from the work with signature B59.850 in the Catedral del Pilar in Zaragoza [16].
- **SEILS**. The SEILS dataset contains 151 printed pages of the "Il Lauro Secco" collection corresponding to an anthology of 16th-century Italian madrigals in white Mensural notation [17].
- **GUATEMALA**. The Guatemala dataset presents 383 handwritten pages from a polyphonic choir book, part of a larger collection held at the "Archivo Histórico Arquidiocesano de Guatemala" [18].
- **MOTTECTA**. This dataset corresponds to the work "Mottecta (Mottecta Francisci Guerreri, que partim quaternis partim quinis alia senis alia octonis concinuntur vocibus, liber secundus dataset)", authored by Francisco Guerrero in the 16th-century and edited by Giacomo Vincenti in the 17th-century. This 297-printed mensural pages corpus has been obtained from the collection of mensural books of the Biblioteca Digital Hispánica. [3]
- **PRIMENS**. The Printed Images of Mensural Staves (PrIMenS) dataset is a synthetic corpus that tries to resemble low-quality real scans of printed mensural sources. It has been built from works composed by Agricola, Frye, and Ockeghem available in the Josquin Research Project [4]. Given polyphonic scores encoded in **kern [19] format, each voice is separated into a single file. In order to increase the variability, the original clefs are modified according to the instrument annotation in the voice. To obtain single staves, the whole work has been divided into a random number of measures from 3 to 18, and the resulting files have been converted into **mens [20] format. The corresponding agnostic encoding has been generated following the method described in [17]. The images have been obtained using the digital engraver Verovio [21] by applying random values to all the options in the allowed ranges. Finally, those images have been distorted to simulate real printed image scans by using a random sequence of graphical filters with the GraphicsMagick Image Processing. Additionally, this real-image simulation process has been complemented by composing randomly damaged old paper textures with distorted images.

To better understand the differences that might appear among these corpora, we provide a staff example from each corpus in Fig. 1.

4.3.2 Target Datasets

For the task of testing the suitability of each model, 15 datasets have been chosen. These corpora have been carefully and specifically labeled for this work, and are summarized in Table 1 and Fig. 2.

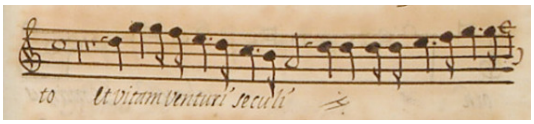
The printed sets have been extracted from the publicly available collection of Mensural books in the Biblioteca Digital Hispánica. [5] The handwritten collections are obtained from archive of Catedral del Pilar in Zaragoza [16].

[2] The whole set, along with a comprehensive description of the contents, can be found at `https://grfia.dlsi.ua.es/polifonia/ismir2023.html`.

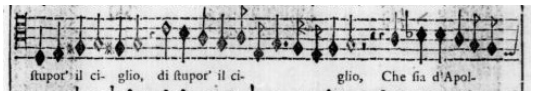
[3] `bdh.bne.es/bnesearch/detalle/bdh0000008932`

[4] `https://josquin.stanford.edu/` (accessed September 1st, 2022).

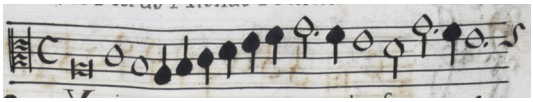
[5] `https://www.bne.es/es/catalogos/biblioteca-digital-hispanica` (accessed March 7th, 2023)



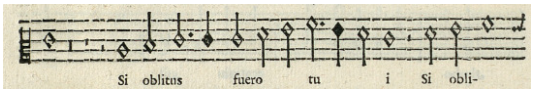
(a) CAPITAN



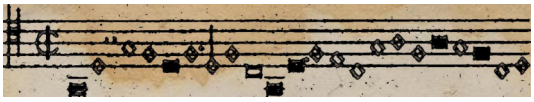
(b) SEILS



(c) GUATEMALA



(d) MOTTECTA



(e) PRIMENS

Figure 1: Samples of staves of the different training datasets employed.

Name (ID)	Number of staves	Printer
Amorosa (Amo)	224	H. of G. Scoto
Chansons (Cha)	173	A. Le Roy, R. Ballard
Dolci (Dol)	170	H. of G. Scoto
Lamentationes (Lam)	528	G. G. Carlino
Madrigali (Mad)	201	G. Scotto
Magnificat (Mag)	1361	Antonio Gardano
Missarum (Mis)	489	H. of G. Scoto
MusicaNova (Mus)	874	Antonio Gardano
Orlande (Orl)	259	A. Le Roy, R. Ballard
Responsoria (Res)	666	G. G. Carlino
Sacrarum (Sac)	460	Antonio Gardano
Villanelle (Vil)	59	G. G. Carlino
B3.28 (B3)	60	*Handwritten*
B50.747 (B50)	80	*Handwritten*
B53.781 (B53)	32	*Handwritten*

Table 1: Features of the different target collections considered in this work.

5. RESULTS

Given the number of training corpora (5), the test datasets (15), and the number of experiments (31), we are able to report up to 465 different SER results. This enables us to properly summarize the experimentation, extracting meaningful learnings that will be used to state the best practices to deal with training data on new projects. The analysis of the results follows. The extended raw results of each experiment are attached to this document in the supplementary material.

5.1 Importance of size and variability

In order to understand which is the best training set selection strategy when facing a new unseen collection, all the possible combinations of the datasets available for training have been evaluated against the different target sets.

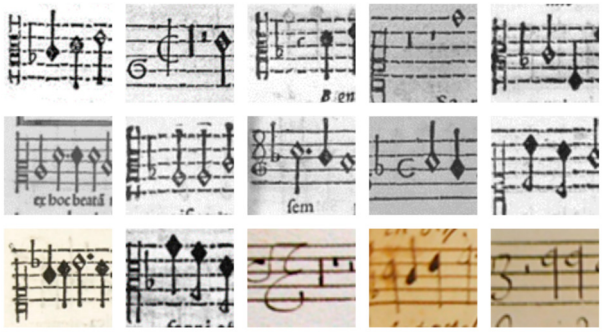


Figure 2: Image examples from the selected corpora as test partition. The images follow a left-right-top-bottom order concerning the list order from Table 1.

The more training sets we include in the combination the greater the number of staves of that combined training set will be. To evaluate which factor is more important, either the variability, given by the number of different training sets included in each combination, or the size as the total number of staves to train, we have plotted in Fig. 3 the summary statistics of the SER obtained by each trained model over all the target collections.

In general, the best behavior has been obtained when merging all the available training corpora. This first outcome may seem obvious, but due to the variability of the training datasets and some of the test works, it was not illogical to expect otherwise. From this result, the fact to be explained is why it performs the best, either due to the size of the training set in terms of the number of staves or the generality the model encompasses due to the training corpora of different natures included.

The plot shows that, although adding more training corpus does not worsen the results, it is not a determining factor. In general, good results are generally obtained with combinations of at least 3 training sets. However, a combination of just two corpora (*i.e.* CS) yields a good performance both in mean and dispersion that denotes its robustness. These two corpora are complementary from the graphical point of view and seem to be representative of both printed sources (SEILS) and handwritten manuscripts (Capitan). When applying 3-corpora training set combinations, the results are equivalent: CGS experiment compared with the GMP, wherein the combination of the first two handwritten corpora and one printed appear compared to the collection of one handwritten and two printed training sets. From these evidences, it can be deduced that the variability of training sets is relevant for better overall performance.

If we focus on the size of the training collection, *i.e.*, the total number of staves used for training, the plot shows that it is not as important as the variability for the final performance. For example, experiment CMS, having less than 4 000 staves, brings better results than experiment GP with over 8 000 samples for training.

To confirm the size is not all that matters, Fig. 4 illustrates the results reported by calculating the number of experiments where the SER is minimized in any of the target datasets, taking into account the number of datasets used

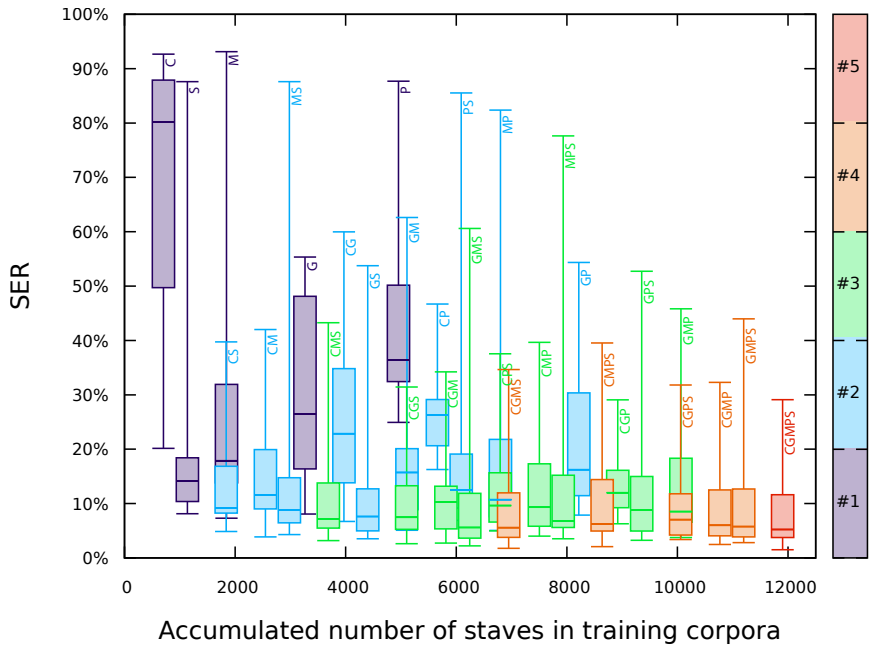


Figure 3: The boxplot shows different statistical SER figures (min, Q1, mean, Q3, max) over the test corpora using a different combination of training corpora. The colors shown in the right bar represent the number of training corpora used in each experiment. The labels are the initials of the corpora included in each training set: **C**: Capitan, **S**: SEILS, **M**: Mottecta, **G**: Guatemala, **P**: PrIMenS.

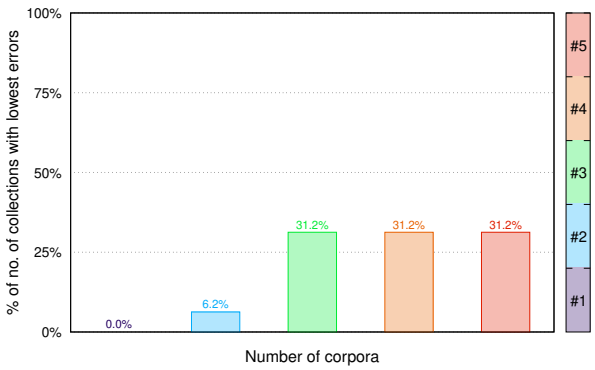


Figure 4: Percentage of experiments that minimize the SER value for any of the available test corpora.

to train. It can be noticed that trained experiments with sizes 3, 4, and 5 report a value of 31.2%. Aside from the value itself, what this aspect exposes is that the size of your dataset at a given point is no longer a critical factor for the transcription quality.

5.2 The complexity of a corpus

The average SER values for all experiments on each target dataset are plotted in Fig. 5. The main noticeable aspect is the difference between Q1 and Q3 (the colored box ends) in the diverse corpora. This substantial contrast in dispersion is what we named "The complexity of a corpus". The plot shows that, as expected, the performance depends on the precise selection of the combination of training corpora to use. The maximum SER values are obtained when the training data is built from just one dataset.

In general, the worst results in the graph are obtained for handwritten target works (those named with the prefix "B") because, intrinsically, they are more difficult to deal with and need a higher variability in the number of training corpora of handwritten works.

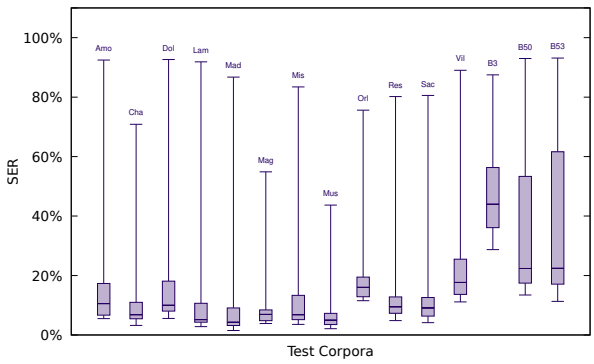


Figure 5: The boxplot shows different statistical SER figures over all experiments made in each one of the testing corpora.

5.3 The importance of leveraging the availability of training corpora

Figure 6 shows the results of the experiments that use each specific training corpus compared to the experiments that do not use it. The image presents the casuistry when having to choose either adding new samples from a different dataset or continue increasing the size of existing labeled samples. As the image reveals, every dataset available for the train, no matter the type—printed or handwritten, real or synthetic—should be included. It is worth mentioning, that the relevance of adding a new corpus is more noticeable than others. For example, referring to the Capitan corpus, if we compare the experiments CMP – MP, CPS – PS,

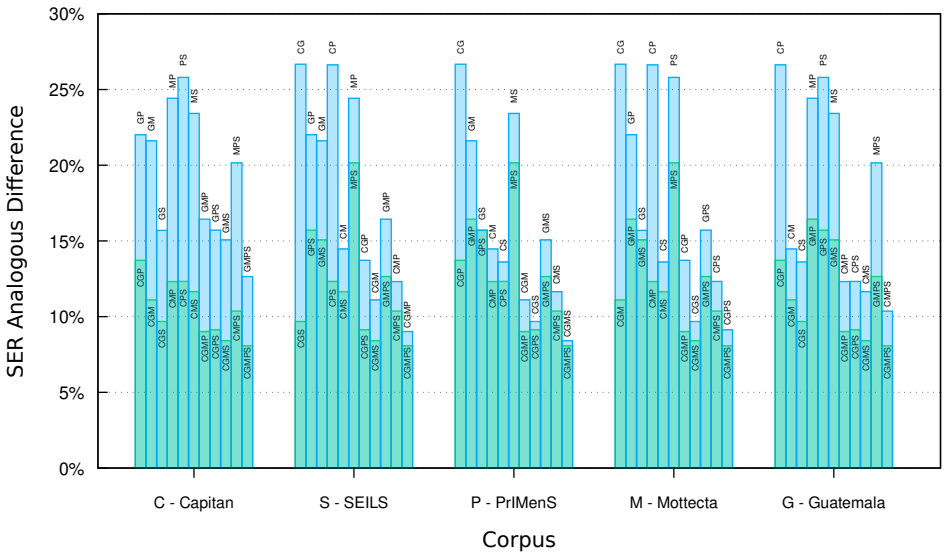


Figure 6: Comparison between all experiments containing (green) and not containing (blue) each training set.

and CMS – MS, we can observe this phenomenon: because of the variability that Capitan adds to the training set, the improvement is noticeable. Therefore, a new corpus seems to generally improve the model performance, as outlined in Fig. 5.

But not only adding a different corpus helps to improve, as the key is to be aware of what is missing in terms of graphical variability in the available training data to build a more robust model. An interesting piece of evidence in the plot that shows how to proceed when this happens is to notice that even a synthetic corpus helps in improving the overall results when it complements the available original training data. Note the reduction in SER when adding PrIMenS, that synthetically simulates printed sources, to complement two other handwritten datasets (Capitan and Guatemala).

5.4 Lessons learned

In order to summarize and establish a set of best practices to improve the generalization performance of OMR systems in the absence of specific training data, we will introduce some questions and answers related to the knowledge acquired from the experimental outcomes.

- **Which is the best choice to transcribe a new collection?** In general, one must use all the available training corpora even if some of them are quite different from the target collection.
- **Is it better to have fewer collections with a high number of samples or more collections with fewer samples each?** It is preferable to have more variability even at the cost of a smaller sample set.
- **How important is it to be aware of the collection to transcribe for selecting the right corpora to train the model?** It is indeed relevant, and depending on the difficulty (for example, whether or not it is handwritten) the differences in performance can be very varied.
- **Does the introduction of a synthetic corpus improve the performance?** Yes, the introduction of a reliable synthetic collection adds size and variability to the training data, enabling better performance rates.

We consider that these answers can be used as general *rules of thumbs*, although of course in certain cases they may not hold.

6. CONCLUSIONS

OMR promises to make written music collections more accessible and browsable by automatically recognizing the symbolic content from their images. However, modern technologies are based on machine learning with deep neural networks, which typically causes unpredictable performance when processing a collection for which no specific training data is available. In this work, we have studied this issue using a large number of training and test collections depicting Mensural notation. This extensive study has been developed considering a state-of-the-art model as representative of the ability to transfer knowledge between collections with dissimilar characteristics.

Our experiments allowed us to analyze various phenomena related to the synergies created between different training collections, the importance of choosing a good recognition trained model to alleviate the uncertainty about performance in a new collection, as well as a series of general good practices on how to proceed for training general OMR models.

As future work, we want to keep on in this line of investigating practical aspects of OMR systems that have a direct impact on particular use cases. For example, we want to extend the case study to the scenario of transfer learning and fine-tuning, where a (limited) amount of training data from a new collection can be assumed. Also, it is interesting to analyze the nature of the errors made by the different OMR models, as well as to have a more precise estimate of the impact of the different errors on the amount of effort required during the post-editing correction process.

7. ACKNOWLEDGMENT

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.

8. REFERENCES

[1] D. Bainbridge and T. Bell, "The challenge of optical music recognition," *Computers and the Humanities*, vol. 35, pp. 95–121, 2001.

[2] 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.

[3] 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)*. IEEE, 2018, pp. 163–168.

[4] 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.

[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. Chowdhury and L. Vig, "An efficient end-to-end neural model for handwritten text recognition," in *British Machine Vision Conference 2018, BMVC 2018, Newcastle, UK, September 3-6, 2018*. BMVA Press, 2018, p. 202.

[7] 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 *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 4774–4778.

[8] 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.

[9] 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, ISMIR 2021, Online, November 7-12*, 2021.

[10] 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.

[11] Y. Bengio, Y. Lecun, and G. Hinton, "Deep learning for AI," *Communications of the ACM*, vol. 64, no. 7, pp. 58–65, 2021.

[12] L. Pugin and T. Crawford, "Evaluating OMR on the early music online collection," in *Proceedings of the 14th International Society for Music Information Retrieval Conference, ISMIR 2013, Curitiba, Brazil, November 4-8, 2013*, A. de Souza Britto Jr., F. Gouyon, and S. Dixon, Eds., 2013, pp. 439–444.

[13] 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.

[14] J. Calvo-Zaragoza, A. Toselli, and E. Vidal, "Handwritten music recognition for mensural notation with convolutional recurrent neural networks," *Pattern Recognition Letters*, vol. 128, 08 2019.

[15] 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.

[16] 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*, M. I. Mandel, J. Devaney, D. Turnbull, and G. Tzanetakis, Eds., 2016, pp. 509–514.

[17] E. Parada-Cabaleiro, A. Batliner, and B. Schuller, "A diplomatic edition of il lauro secco: Ground truth for omr of white mensural notation," 10 2019.

[18] 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.

[19] D. Huron, "Humdrum and Kern: Selective Feature Encoding BT - Beyond MIDI: The handbook of musical codes," in *Beyond MIDI: The handbook of musical codes*. Cambridge, MA, USA: MIT Press, jan 1997, pp. 375–401.

[20] D. Rizo, N. Pascual-León, and C. S. Sapp, "White mensural manual encoding: from humdrum to mei,"

Cuadernos de Investigación Musical, no. 6, pp. 373–393, 2018.

[21] 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*, H. Wang, Y. Yang, and J. H. Lee, Eds., 2014, pp. 107–112.

COMPOSER'S ASSISTANT: AN INTERACTIVE TRANSFORMER FOR MULTI-TRACK MIDI INFILLING

Martin E. Malandro
Sam Houston State University
malandro@shsu.edu

ABSTRACT

We introduce Composer's Assistant, a system for interactive human-computer composition in the REAPER digital audio workstation. We consider the task of multi-track MIDI infilling when arbitrary track-measures have been deleted from a contiguous slice of measures from a MIDI file, and we train a T5-like model to accomplish this task. Composer's Assistant consists of this model together with scripts that enable interaction with the model in REAPER. We conduct objective and subjective tests of our model. We release our complete system, consisting of source code, pretrained models, and REAPER scripts. Our models were trained only on permissively-licensed MIDI files.

1. INTRODUCTION

Many generative models for music exist. For instance, MuseNet [1] and SymphonyNet [2] can generate or continue a piece of music, and Music Transformer [3] can continue a piano performance or harmonize a piano melody. When we tried using these tools as compositional aides, however, we quickly ran into limitations. For instance, while Music Transformer is capable of harmonizing a given melody, it does not offer the ability to keep part of the harmonization and regenerate the other part. MuseNet and SymphonyNet can generate a continuation of a user's prompt, but do not allow the user to regenerate individual instruments or measures within the continuation while keeping the rest of the continuation intact.

DeepBach [4] can perform infilling on Bach-like chorales in a window specified by the user. Motivated by the idea of extending the DeepBach user experience to more styles, arbitrary collections of instruments, and arbitrary infilling target locations, we train a transformer [5, 6] model on the task of multi-track MIDI infilling. Our model allows composers to generate new notes for arbitrary subsets of track-measures in their compositions, conditioned on any contiguous slice of measures containing the subset. (By a *track-measure*, we simply mean a measure within a track—see Figure 1.) We also build a novel system for interacting with our model in the REAPER digital audio workstation (DAW). [1] Our system is cross-platform and easy to install. When a user runs one of our REAPER scripts, a model prompt is built directly from the slice of measures selected in the user's REAPER project, our model evaluates the prompt, and the model output is written back into the user's project—see Figure 1. All of this happens within a few seconds, allowing the user to listen to the output, modify it to create a new prompt, generate an output from that, etc. This allows our model to be used in an interactive manner, where model outputs are refined by the user over the course of several prompts.

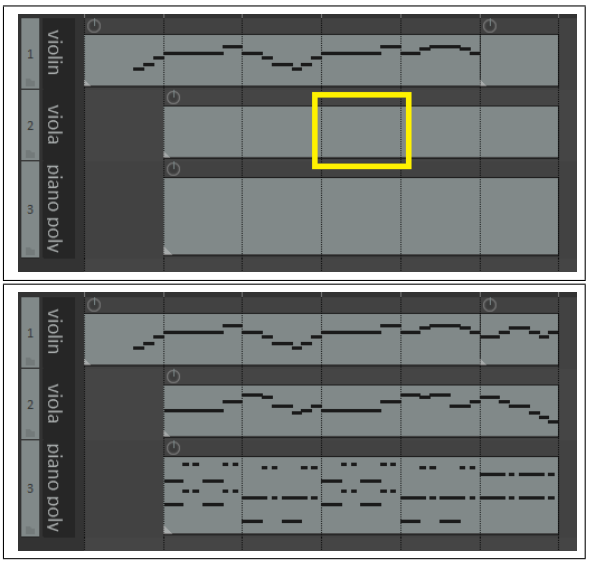


Figure 1. A prompt in REAPER, followed by a model output. Vertical lines separate measures. Users place empty MIDI items in REAPER to tell the model in which measures to write notes, and track names to tell the model what instrument is on each track. A track-measure in the prompt is boxed. Our model writes at least one note into every track-measure in every empty MIDI item in the prompt.

We note that our infilling objective includes continuing a piece of music, simply by including empty measures at the end of the piece in the prompt. Additionally, our model has the ability to write variations: One can randomly mask $1/n$ of the track-measures in a measure selection and ask the model to fill in those parts, then feed the result back into the model with another $1/n$ masked, and so on, until

© M. E. Malandro. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** M. E. Malandro, "Composer's Assistant: An Interactive Transformer for Multi-Track MIDI Infilling", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.*, Milan, Italy, 2023.

[1] Our system and video demo are available at https://github.com/m-malandro/composers-assistant-REAPER.

all track-measures have been masked and filled.

Toward the end of this project we discovered MMM [7, 8], which consists of two separate GPT-2–like [9] models trained on the tasks of measure infilling and track infilling. The authors include code to use these models to infill arbitrary subsets of track-measures, as we do. MMM comes in 4-bar and 8-bar variants, which are limited to inputs with 12 and 6 tracks, respectively, and its web demo is limited to inputs having a 4/4 time signature.

The primary contributions of this work are as follows. First, in Section 3 we introduce a novel data filtering and preprocessing approach, applicable to any MIDI dataset used for training models. Our approach helps rectify certain issues we have encountered when using other models. Second, we train and release a new model, capable of infilling arbitrary track-measures in an arbitrary slice of measures in a MIDI file, with no effective restriction (aside from a soft input token limit of 1650) on tempos, number of measures, or number of instrument tracks. Tracks may be polyphonic or monophonic in any combination. The only time signature restriction is that all measures must be eight quarter notes or fewer. Our model is more flexible than MMM and compares favorably to MMM in both objective and subjective tests—see Sections 6–7. Additionally, our model was trained only on permissively-licensed MIDI files, so its outputs should be usable by composers with minimal risk—see Section 5. Finally, we release our complete system, including training code and scripts that enable rapid interaction with our model in REAPER. Our model is the first DAW-integrated model capable of infilling parts for all 128 pitched MIDI instruments (including repeated instruments) and drums, in any combination.

2. RELATED WORK

As mentioned in Section 1, MMM [7, 8] performs multi-track infilling for all MIDI instruments (subject to bar and track limits), and DeepBach [4] performs multi-track infilling for Bach-like chorales. Coconet [10] also performs multi-track infilling for Bach-like chorales. MusIAC [11] incorporates user controls and performs track-based and measure-based infilling, although its inputs and outputs are limited to a maximum of three tracks, 16 measures, and four common time signatures. MusicVAE [12] can interpolate between two given clips of music, which can be viewed as a type of infilling. To our knowledge, all other existing music infilling systems are limited to monophonic infilling [13–17] or single-instrument infilling [18, 19].

Generating or continuing a piece of music can be seen as a special case of infilling. Models which can generate or continue a piece of music include [1–3, 20, 21].

Previous DAW-integrated generative music systems include [4, 18, 22]. NONOTO [23] is a model-agnostic interface that can be linked with a model to perform interactive measure-based infilling. This interface could potentially be altered to allow for the expanded type of infilling our model is capable of. However, we opt to build an interface between our model and REAPER directly, essentially using REAPER as the GUI for our model.

3. DATA FILTERING AND PREPROCESSING

In this section we describe our filtering and preprocessing approach, any portion of which can be applied to any dataset of MIDI files. First, we remove from the dataset any file whose notes seem to have no relation to the underlying grid (Section 3.1). Next, we dedupe files from the dataset using note onset chromagrams (Section 3.2). Finally, we preprocess all remaining files to standardize properties like track order (Section 3.3). This final preprocessing step includes a method for detecting and removing shifted duplicate and near-duplicate tracks within files (Section 3.4).

3.1 Cosine Similarity for On-Grid Note Detection

Every MIDI file has a measure and grid structure defined by tempo and time signature events. However, MIDI file authors are free to ignore this structure, and frequently do when recording free-flowing performances. Other models we have used occasionally write a note in the wrong place—e.g., a 32nd note away from where it clearly should be—and a small experiment we ran suggests that training on MIDI files that don't quantize well to the grid used by the model is a major cause of this. To address this, we remove from our dataset any MIDI file whose note onsets appear to have no relation to the underlying grid. This is not as simple as checking whether all (or most) note onsets occur on the grid, as many MIDI file authors who use the grid include "humanization" of note timings, where many note onsets that occur slightly off the grid nevertheless quantize correctly to the grid. For instance, in a MIDI file with a resolution of 960 ticks per quarter note, a humanized quarter-note performance might have notes occurring in a 40-tick window centered at every 960th tick.

To perform this filtering, given a MIDI file M, we quantize the note onsets in M to a resolution of 12 ticks per quarter note, and we form a length-12 vector v_M whose ith entry ($i \in \{0, \ldots, 11\}$) is the number of note onsets in M occurring i ticks after a grid quarter note. The idea is that if the note onsets in M have nothing to do with the grid, then v_M will point in a similar direction to the uniform vector $v_1 = (1, \ldots, 1) \in \mathbb{R}^{12}$. We therefore compute the cosine of the angle θ_M between v_M and v_1:

$$\cos(\theta_M) = \frac{\langle v_M, v_1 \rangle}{||v_M|| \cdot ||v_1||},$$

and we declare a threshold T such that when $\cos(\theta_M) > T$ we remove the file M from our dataset. Hand exploration indicated that $T = 0.8$ was a reasonable threshold, which we chose for this project. We note that a straight fully-on-grid 8th-note pattern M has $\cos(\theta_M) \approx 0.41$ and a straight fully-on-grid 16th-note pattern M has $\cos(\theta_M) \approx 0.58$.

3.2 Deduping Using Note Onset Chromagrams

We dedupe our dataset to avoid data imbalance during training and to prevent overlap between our training and test sets. Given a MIDI file M, we compute a size-12 set of note onset chromagrams using the following procedure.

First, we remove all drum tracks from M. Then, using a 12-tick-per-quarter-note grid, we quantize the note onsets in M to the nearest 16th note or 8th note triplet. Then, we remove all empty measures at the beginning and end of M, and we replace each set of contiguous empty measures within M with one empty measure. Then, for each tick in M and for each pitch class, we record whether M has at least one note onset of that pitch class at that tick. This information comprises one note onset chromagram for M. The other 11 come from repeating this procedure for each possible transposition of M. We dedupe the dataset by keeping only one file with a given set of note onset chromagrams. Quantization helps us catch files that differ only trivially in grid resolution and/or note onset times, while transposition helps us catch files that differ only in key.

3.3 Preprocessing of Individual MIDI Files

After deduping, we preprocess each MIDI file in our dataset in the following way.

First, we arrange the information in the MIDI file so that each track holds notes for one instrument. We order tracks according to their MIDI instrument number (0–127), taking drums as instrument 128. We also consolidate all drum tracks to a single track, and we apply a drum simplification map (consolidating, e.g., three different bass drum pitches to a single pitch).

Next, we apply pedal information in the file (if present) to extend note lengths, and then delete all continuous controller (cc) data. We do not model cc data in this project.

With the exception of drums, we allow multiple tracks to use the same instrument. However, when this happens, if there is more than one track having a given instrument, we remove all but one of those tracks that are equal to, a shift of, or close to a shift of another track with the same instrument, using the procedure in Section 3.4.

We impose the restriction that all measures must be eight quarter notes or fewer. If any time signature in the file declares longer measures, we alter the time signatures to shorten the measures.

Finally, using a 24-tick-per-quarter-note grid, we quantize the events in the file to the nearest 32nd note or 16th note triplet. This is ultimately the level of quantization we use to train our model. (Earlier experiments involved quantizing to 16th notes or 16th notes + 8th note triplets, which we found insufficient for expressive generation.)

3.4 Removing Shifted Duplicate and Near-Duplicate Tracks

A MIDI file may contain duplicate tracks. Such tracks contain no useful information for modeling, so we remove them. Shifted duplicate tracks are frequently used by MIDI file authors to encode *delay* effects (as the MIDI spec offers no way to encode the use of a delay directly). Choosing to use a delay is a mixing decision, not a compositional decision, and we want our model to focus on making compositional decisions, so we remove shifted duplicate tracks as well. We have also seen tracks that are duplicates or shifted duplicates of other tracks within a file, plus or minus a few notes and/or humanization. We remove such near-duplicate tracks as well.

Given a note n in a track T, let $\text{st}(n)$ and $\text{end}(n)$ indicate the start and end times of the note n, respectively, and let $\text{pitch}(n) \in \{0, \ldots, 127\}$ indicate the MIDI pitch of n. We record, for $p \in \{0, \ldots, 127\}$, the union of closed intervals

$$I_T(p) = \cup_{\text{n} \in T:\text{pitch}(n)=p} \{[\text{st}(n), \text{end}(n)]\} \subseteq \mathbb{R},$$

and we define $|I_T| = \sum_{p=0}^{127} |I_T(p)|$, where $|I_T(p)|$ is the sum of the lengths of the disjoint intervals in $I_T(p)$.

Given tracks T_1 and T_2, we define the *overlap measure* $O(T_1, T_2) \in [0, 1] \subseteq \mathbb{R}$ to be

$$O(T_1, T_2) = \frac{\sum_{p=0}^{127} |I_{T_1}(p) \cap I_{T_2}(p)|}{\max\left(|I_{T_1}|, |I_{T_2}|\right)}.$$

The idea is that $O(T_1, T_2)$ measures the percentage of the note intervals in the larger of the two tracks accounted for by the note intervals in the smaller of the two.

We use a threshold of 0.9 for asserting near-overlap between two tracks. As we go through the tracks in a MIDI file in order, a later track T is thrown out if there exists an earlier track T_0 using the same instrument such that some shift T_s of T of no more than a half note has the property that $O(T_0, T_s) \geq 0.9$. In our experience with our trained model, we have found this preprocessing step sufficient to prevent the model from outputting duplicates or shifted duplicates of tracks in its inputs.

4. OUR LANGUAGE

After applying the procedure from Section 3 to a collection of MIDI files, we process the files into an event-based language for modeling. Our language is similar to the standard event-based MIDI language used for piano performance modeling in [3]. However, we use explicit measure tokens to denote the start of each measure. Also, we do not model velocity of individual notes directly. Instead, we assign a dynamics level to each measure based on the average velocity of the notes in the measure. We use eight dynamics levels, with thresholds learned from data.

The tokens used by our language are as follows:

- M:x, $x \in \{0, \ldots, 7\}$. Declares a measure of dynamics level x.
- B:x, $x \in \{0, \ldots, 7\}$. Declares the tempo (BPM) level at the start of a measure. We use eight tempo levels, with thresholds learned from data.
- L:x, $x \in \{1, \ldots, 192\}$. Declares that a measure has length equal to x ticks.
- I:x, $x \in \{0, \ldots, 128\}$. Changes the current instrument to MIDI instrument x ($128 =$ drums).
- R:x, $x \in \{1, \ldots, 63\}$. Declares that the current instrument is the same MIDI instrument as another instrument in the file/project, but on a different track. Higher x values indicate lower average pitch.

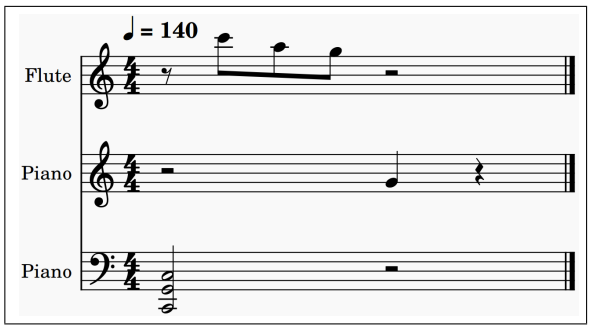


Figure 2. We tokenize this measure as M:5 B:6 L:96 I:0 w:48 d:24 N:67 I:0 R:1 d:48 N:36 N:43 N:48 I:73 w:12 d:12 N:84 w:12 N:81 w:12 N:79. Note that piano and flute are MIDI instruments 0 and 73.

- N:x, $x \in \{0, \ldots, 127\}$. Note of pitch x. Used by instruments 0–127.
- D:x, $x \in \{0, \ldots, 127\}$. Drum hit of drum pitch x. Used by instrument 128.
- d:x, $x \in \{0, \ldots, 192\}$. Sets the duration of each note declared from this point forward to x ticks.
- w:x, $x \in \{1, \ldots, 191\}$. Advances the current insertion point for new notes in the measure by x ticks.
- `<extra_id_x>`, $x \in \{0, \ldots, 255\}$. Mask tokens.
- `<mono>`, `<poly>`. Instructs the model to write a monophonic or polyphonic part for a masked part. For our purposes a monophonic part is one where no two notes in the part have the same onset tick.

Tokens are assembled in a standardized manner to represent measures. Each measure begins with M:x, B:x, and L:x tokens. I: commands are included for a measure only when that instrument is present in the measure. We do not intermingle instrument note instructions as we write each measure from left to right (as MuseNet [1] did), as that would make it difficult to mask individual instrument parts within measures. Rather, we write the full part for one instrument within the measure before writing the full part for the next instrument within the measure. Figure 2 contains an example of a tokenized measure.

To form songs, we simply concatenate measures.

5. MODEL, DATA, AND TRAINING PROCEDURE

We use recent recommendations from the language modeling community to design and train our model. Based on the recommendations in [24–26], we choose a T5 (full, relative-attention) encoder-decoder architecture [6]. We opt for a full attention model because such models were found to outperform memory-efficient models in [24] when the full input sequence fits in memory, as we expect to be the case in most real-world applications of our model. Also, we adopt the *DeepNarrow* strategy of [27], opting for a model dimension of 384, 10 encoder layers, and 10 decoder layers. For training, we use the `pytorch` [28] Hugging Face [29] implementation of T5. For inference, we use nucleus sampling [30] with a threshold of $p = 0.95$.

To train a model that is essentially free of copyright worry, we collect MIDI files from the internet marked as being in the public domain, freely available to use without attribution, or available under a CC BY license. We exclude files marked as having share-alike or non-commercial licenses, since we want composers to be able to use model outputs however they wish. We also collect private donations and files from the internet for which we secure direct permission from the MIDI file authors to use for training. This results in a dataset of approximately 40K MIDI files after filtering. Most of our training files are in Western classical, folk, and hymnal styles, although some modern styles are also present.

We follow the standard approach to the training of large language models of splitting our training procedure into pretraining and finetuning phases. A similar approach was also used in [31]. For pretraining, we use the T5 corrupted-span sequence-to-sequence objective [6]. We begin by pretraining on the 192K training files in the CocoChorales [32] dataset and their piano reductions for three epochs. The CocoChorales are only used for this initial pretraining to teach the model the basics of music theory and our language. We then move on to our dataset of 40K MIDI files. After tokenization and corruption, we greedily chunk each song into inputs of 512 or fewer (short) or 1650 or fewer (long) tokens. Additionally, each song in our dataset is transposed a random amount between -5 and +6 semitones (inclusive) for each epoch. Following the recommendations in [24], we train our model on short examples for 20 epochs and then long examples for 11 epochs. We release the resulting pretrained model, which others may find useful for finetuning on downstream tasks.

For finetuning, we continue to leverage the corrupted-span sequence-to-sequence objective to finetune our model on the task of multi-track MIDI infilling. We create training examples from songs in our training dataset by taking slices of measures from the songs and masking subsets of track-measures from these slices. During finetuning every N:, D:, d:, and w: token for a given track-measure is masked, and corresponds to a single mask token. With probability 0.75, we add a `<poly>` or `<mono>` token corresponding to the nature of the masked tokens for each mask. (We choose not to include these tokens in every training example since users will not always include these instructions in their prompts.) Finetuning examples are limited to inputs with a maximum of 1650 tokens and outputs with a maximum of 1650 tokens.

We generate our finetuning masks by selecting from mask patterns that we consider to be musically relevant and/or useful for training. To help train our model for use on small numbers of measures, we also occasionally (15% of the time) truncate examples to a random smaller number of measures than the number allowed by our token limits. As with pretraining, each example is transposed randomly. We finetune our model for 51 epochs, and we release the resulting finetuned model.

Task	Our Model	Our Model -MP	MMM-8	MMM-4
Note F_1 results. Higher is better.				
8-bar random infill	**0.5414** $\pm$ $(0.1887)^a$	0.5315 $\pm$ $(0.1904)^b$	0.4153 $\pm$ $(0.1819)^c$	0.4025 $\pm$ $(0.1652)^d$
16-bar random infill *	**0.5771** $\pm$ $(0.1661)^a$	0.5705 $\pm$ $(0.1669)^b$	0.4133 $\pm$ $(0.1534)^c$	0.4059 $\pm$ $(0.1399)^d$
8-bar track infill	**0.179** $\pm$ $(0.1902)^a$	0.1634 $\pm$ $(0.18)^b$	0.1063 $\pm$ $(0.1573)^d$	0.1427 $\pm$ $(0.1646)^c$
16-bar track infill	**0.1773** $\pm$ $(0.1752)^a$	0.1609 $\pm$ $(0.165)^b$	0.1107 $\pm$ $(0.1383)^d$	0.1467 $\pm$ $(0.1489)^c$
8-bar last-bar fill	0.5019 $\pm$ $(0.2719)^a$	**0.5063** $\pm$ $(0.2751)^a$	0.4329 $\pm$ $(0.2445)^b$	0.3756 $\pm$ $(0.2289)^c$
16-bar last-bar fill *	**0.5415** $\pm$ $(0.2853)^a$	0.539 $\pm$ $(0.2823)^a$	0.4338 $\pm$ $(0.2468)^b$	0.3818 $\pm$ $(0.2293)^c$
Pitch class histogram entropy difference results. Lower is better.				
8-bar random infill	**0.2845** $\pm$ $(0.1627)^a$	0.2948 $\pm$ $(0.1597)^b$	0.3045 $\pm$ $(0.1561)^c$	0.3049 $\pm$ $(0.1497)^c$
16-bar random infill	**0.2691** $\pm$ $(0.1325)^a$	0.2797 $\pm$ $(0.1326)^b$	0.3093 $\pm$ $(0.124)^c$	0.3063 $\pm$ $(0.1138)^c$
8-bar track infill	0.3933 $\pm$ $(0.3032)^c$	0.42 $\pm$ $(0.3134)^d$	**0.2864** $\pm$ $(0.2966)^a$	0.3021 $\pm$ $(0.2517)^b$
16-bar track infill	0.3842 $\pm$ $(0.2654)^c$	0.3995 $\pm$ $(0.2763)^c$	**0.284** $\pm$ $(0.2348)^a$	0.3036 $\pm$ $(0.2072)^b$
8-bar last-bar fill	**0.3018** $\pm$ $(0.2661)^a$	0.3072 $\pm$ $(0.2692)^a$	0.3213 $\pm$ $(0.2602)^b$	0.3439 $\pm$ $(0.2777)^c$
16-bar last-bar fill *	**0.2851** $\pm$ $(0.2652)^a$	0.2925 $\pm$ $(0.2672)^a$	0.3209 $\pm$ $(0.2619)^b$	0.3454 $\pm$ $(0.2741)^c$
Groove similarity results. Higher is better.				
8-bar random infill	**0.9534** $\pm$ $(0.0298)^a$	0.9519 $\pm$ $(0.0306)^b$	0.9333 $\pm$ $(0.0369)^c$	0.9314 $\pm$ $(0.0364)^d$
16-bar random infill *	**0.956** $\pm$ $(0.027)^a$	0.9552 $\pm$ $(0.0275)^b$	0.9323 $\pm$ $(0.0337)^c$	0.9317 $\pm$ $(0.032)^c$
8-bar track infill	**0.9115** $\pm$ $(0.0592)^a$	0.9069 $\pm$ $(0.0617)^b$	0.8921 $\pm$ $(0.0695)^d$	0.8987 $\pm$ $(0.0626)^c$
16-bar track infill	**0.9113** $\pm$ $(0.0547)^a$	0.9082 $\pm$ $(0.0553)^b$	0.8946 $\pm$ $(0.0561)^d$	0.9011 $\pm$ $(0.0536)^c$
8-bar last-bar fill	0.9517 $\pm$ $(0.0414)^a$	**0.9524** $\pm$ $(0.0411)^a$	0.9381 $\pm$ $(0.045)^b$	0.9334 $\pm$ $(0.0457)^c$
16-bar last-bar fill *	**0.9544** $\pm$ $(0.0481)^a$	0.9542 $\pm$ $(0.0424)^a$	0.938 $\pm$ $(0.051)^b$	0.9339 $\pm$ $(0.0475)^c$

Table 1. Objective infilling summary statistics. All cells are of the form mean $\pm$ (std dev)^s, where s is a letter. Different letters within a row indicate significant location differences ($p < 0.01$) in the samples for that row according to a Wilcoxon signed rank test with Holm-Bonferroni correction. Asterisks ($*$) indicate a significant performance difference ($p < 0.01$) between a 16-bar task and the 8-bar task in the previous row for our model according to a Wilcoxon rank sum test.

6. OBJECTIVE EVALUATION OF OUR MODEL

To form our test set, we select a set of 2500 MIDI files from the Lakh MIDI dataset [33, 34] that is disjoint (according to the procedure in Section 3.2) from our training set, all in 4/4 time and having at least 16 measures. Given a MIDI file in our test set, for each of the three mask patterns below, we select an 8- and a 16-measure slice of the file and mask the selected slice with that mask pattern. We thus generate six test examples from each test file, corresponding to the six different tasks on which we evaluate models. Given a slice of measures, our mask patterns for testing are:

1. Random mask: Each track-measure in the slice is masked with probability 0.5.
2. Track mask: Up to half of the tracks t are selected at random from the slice, and every measure of each such track t is masked.
3. Last-bar mask: Given the last measure m of the slice, measure m of every track is masked. This pattern is used to measure the ability of models to continue songs.

The ground truth for each example consists of the masked notes in the example. In our test data, 99% and 75% of our 8-measure and 16-measure prompts (respectively) encode to 1650 or fewer tokens. When input prompts are longer than 1650 tokens, we chunk the prompts prior to evaluating them with our model.

To compare our model to MMM [7, 8], we modify the MMM Colab worksheet to run our examples through the MMM models in batches. We recreate our test examples, quantizing them from their underlying MIDI files to MMM's 12-tick-per-quarter-note resolution, and then modify them to accommodate the restrictions of the MMM models: Since the 4-bar and 8-bar MMM models are limited to inputs containing a maximum of 12 and 6 tracks, respectively, we chunk each test example into 4-bar and 8-bar chunks, and then we split each chunk into sub-chunks consisting of up to 12 and 6 tracks. The MMM models have a strict input + output token limit of 2048, so when sub-chunking, we only add enough tracks to a sub-chunk to ensure that the input + ground truth has no more than 2048 tokens. This biases the comparison in favor of the MMM models somewhat, as this requires us to look at the length of the ground truth as part of the input chunking procedure. Also, our test set is contained in MMM's training set, but there is no reasonable way to avoid this as the MMM models were trained on the full Lakh MIDI dataset. (We wanted a diverse and well-randomized test set, and the Lakh MIDI dataset is the only publicly-available dataset we are aware of that fits this bill.)

We evaluate models with standard metrics: Note F_1 [35], average pitch class histogram entropy difference [19, 36], and average groove similarity [19, 36]. Note F_1 measures how closely the generated notes match, on a note-for-note basis, the notes in the ground truth. (For our purposes, a generated note matches a note n in the ground

	Real Music	Our Model	MMM
1st place	**66**	32	27
Avg rank	**1.664**	2.032	2.304

p-values	Our Model	MMM
MMM	0.0239	-
Real Music	0.0034	$2.3 \cdot 10^{-5}$

Table 2. Subjective results from our listening test.

truth if and only if its onset tick, measure, pitch, and track match exactly those of n.) The other metrics measure how well certain higher-level statistics of the generated notes match those of real music. For pitch class histogram entropy calculations, drums are ignored. Each metric is computed on a per-example basis, and then for each model, task, and metric, the 2500 results are averaged to give the results in Table 1. For fairness of groove similarity comparison, we use a denominator of 48 for all models. (This is reasonable, as our models and the MMM models both effectively have 48 possible note onset positions per 4/4 measure.) For our model, "-MP" indicates that the examples were evaluated without `<mono>` or `<poly>` tokens present.

For each row of Table 1, we perform a Wilcoxon signed rank test [37] with Holm-Bonferroni correction [38]. We find significant differences between our model and the MMM models in all 18 rows, with our model outperforming the MMM models in 16 out of 18 rows. The MMM models outperform our model only for pitch class histogram entropy difference for full-track infilling.

Additionally, we find a significant difference in our model's performance when `<mono>` and `<poly>` tokens are included in prompts in 11 out of 18 rows. All significant differences favor including these tokens, suggesting that the development of additional user controls (as in [11]) would be a useful line of future work.

Finally, a Wilcoxon rank sum test [37] reveals significant differences ($p < 0.01$) in 8-bar versus 16-bar results for our model in five out of nine comparisons. All significant differences favor the 16-bar results, emphasizing the importance of training on longer measure slices. However, we never observe a significant difference in 8-bar versus 16-bar results for track infilling, suggesting that larger context windows generally provide no additional useful information for completing this particular task.

Additional experiments not reported here indicate that scaling our training approach (training larger models on more data) is a feasible path for improving model performance on the metrics presented here.

7. SUBJECTIVE EVALUATION OF OUR MODEL

While the results in Section 6 are encouraging, the ground truth may not reflect the only reasonable way to fill in missing notes. To help address this, we conducted a small listening test with 25 participants. We prepared nine examples mostly involving melodic generation. Each example consisted of three 8-measure clips, one of which was real multi-track music. The other two clips were created by removing some tracks from the real music and using our model and MMM to fill those tracks. Participants were shown five of the nine examples at random, and for each example were asked to rank the three clips in order of preference. Results are given in Table 2.

A Wilcoxon signed rank test with Holm-Bonferroni correction reveals significant differences in rankings between all three types of music, with p-values given in Table 2. In this test we see a clear preference for real music, and a significant ($p < 0.05$) preference for music generated by our model over music generated by MMM. One expert participant commented that melodies generated by the models were generally more directionless than those in real music, often failing to drive towards a cadence or "payoff." We agree with this assessment, and this is a shortcoming of our model that we hope to address in future work.

8. LIMITATIONS AND RISKS

Our model writes music that is reflective of its training set. Most of our training files are in Western classical, folk, and hymnal styles. While we included in our training set only files marked as being permissively licensed, it is possible that some files were mismarked. It is also theoretically possible for our model to output copyrighted music, even if such music was not present in the training set.

The most common request we have heard from composers to whom we have shown our system is *personalization*. Generally speaking, they do not want systems that write full songs, and they do not want systems that write "generic" music. Rather, they want systems that can suggest ideas in their style. Some small experiments indicate that our finetuned model can be personalized by individuals (by continuing to finetune the model on their own MIDI files) to write in their styles. Low-rank adaptation [39] of our model may also be possible. Personalization is an avenue we would like to explore formally in future work. For now, our code supports training by users, and our model dimensions were chosen carefully to enable this on consumer hardware. A video card with 6 GB of RAM is sufficient to train our released model on examples with input and output lengths of 1024, and 12 GB of RAM is sufficient to train on examples with input and output lengths of 1650. While this can benefit composers who wish to use our system, there is also the risk that our models may be trained by users to impersonate the styles of others.

9. CONCLUSION

We have introduced Composer's Assistant, a system for interactive human-computer composition in the REAPER digital audio workstation. Composer's Assistant performs multi-track MIDI infilling and comes with an easy-to-use interface. We have released our source code, a pretrained model, a finetuned model, and scripts for interacting with our finetuned model in REAPER. Our models were trained only on permissively-licensed MIDI files.

10. ACKNOWLEDGMENT

We thank the many contributors to our MIDI training set for this project. Contributor acknowledgments can be viewed at our website. [2] We also thank the IT department at Sam Houston State University for building and maintaining the computational server on which we trained our model.

11. REFERENCES

[1] C. Payne, "MuseNet," openai.com/blog/musenet, 2019.

[2] 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.

[3] 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 *Int. Conf. Learning Representations*, 2019.

[4] 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.

[5] 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.

[6] 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.

[7] J. Ens and P. Pasquier, "MMM : Exploring Conditional Multi-Track Music Generation with the Transformer," *arXiv preprint arXiv: 2008.06048*, 2020.

[8] ——, "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.

[9] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language Models are Unsupervised Multitask Learners," 2019. [Online]. Available: https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[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] K. Chen, C. 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.

[17] 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.

[18] G. Hadjeres and L. Crestel, "The Piano Inpainting Application," *arXiv preprint arXiv: 2107.05944*, 2021.

[19] 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.

[20] Y.-S. Huang and Y.-H. Yang, "Pop Music Transformer: Beat-Based Modeling and Generation of Expressive Pop Piano Compositions," in *Proc. 28th ACM Int. Conf. Multimedia*, ser. MM '20. New York,

[2] https://github.com/m-malandro/composers-assistant-REAPER.

NY, USA: Association for Computing Machinery, 2020, p. 1180–1188. [Online]. Available: https://doi.org/10.1145/3394171.3413671

[21] 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 *35th AAAI Conf. Artificial Intelligence*, 2021, pp. 178–186.

[22] 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.

[23] 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.

[24] J. Phang, Y. Zhao, and P. J. Liu, "Investigating Efficiently Extending Transformers for Long Input Summarization," *arXiv preprint arXiv: 2208.04347*, 2022.

[25] N. Shazeer, "GLU Variants Improve Transformer," *arXiv preprint arXiv: 2002.05202*, 2020.

[26] Y. Tay, M. Dehghani, S. Abnar, H. W. Chung, W. Fedus, J. Rao, S. Narang, V. Q. Tran, D. Yogatama, and D. Metzler, "Scaling Laws vs Model Architectures: How does Inductive Bias Influence Scaling?" *arXiv preprint arXiv: 2207.10551*, 2022.

[27] Y. Tay, M. Dehghani, J. Rao, W. Fedus, S. Abnar, H. W. Chung, S. Narang, D. Yogatama, A. Vaswani, and D. Metzler, "Scale Efficiently: Insights from Pre-training and Finetuning Transformers," in *Int. Conf. Learning Representations*, 2022.

[28] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2019, vol. 32, pp. 8024–8035.

[29] 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. Le Scao, S. Gugger, M. Drame, Q. Lhoest, and A. Rush, "Transformers: State-of-the-Art Natural Language Processing," in *Proc. 2020 Conf. Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, Oct. 2020, pp. 38–45.

[30] A. Holtzman, J. Buys, L. Du, M. Forbes, and Y. Choi, "The Curious Case of Neural Text Degeneration," in *Int. Conf. Learning Representations*, 2020.

[31] 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," in *Proc. 20th Int. Society for Music Information Retrieval Conf.*, Delft, The Netherlands, 2019.

[32] Y. Wu, J. Gardner, E. Manilow, I. Simon, C. Hawthorne, and J. Engel, "The Chamber Ensemble Generator: Limitless High-Quality MIR Data via Generative Modeling," *arXiv preprint arXiv:2209.14458*, 2022.

[33] C. Raffel, "The Lakh MIDI Dataset v0.1," https://colinraffel.com/projects/lmd/.

[34] ——, "Learning-Based Methods for Comparing Sequences, with Applications to Audio-to-MIDI Alignment and Matching," Ph.D. dissertation, 2016.

[35] J. Gardner, I. Simon, E. Manilow, C. Hawthorne, and J. Engel, "MT3: Multi-Task Multitrack Music Transcription," in *Int. Conf. Learning Representations*, 2022.

[36] 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.

[37] F. Wilcoxon, "Individual Comparisons by Ranking Methods," *Biometrics Bulletin*, vol. 1, no. 6, pp. 80–83, 1945.

[38] S. Holm, "A Simple Sequentially Rejective Multiple Test Procedure," *Scandinavian Journal of Statistics*, vol. 6, no. 2, pp. 65–70, 1979.

[39] E. 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 *Int. Conf. Learning Representations*, 2022.

THE FAV CORPUS: AN AUDIO DATASET OF FAVORITE PIECES AND EXCERPTS, WITH FORMAL ANALYSES AND MUSIC THEORY DESCRIPTORS

Ethan Lustig
Ethan@EthanLustig.com

David Temperley
Eastman School of Music, University of Rochester

ABSTRACT

We introduce a novel audio corpus, the FAV Corpus, of over 400 favorite musical excerpts and pieces, formal analyses, and free-response comments. In a survey, 140 American university students (mostly music majors) were asked to provide three of their favorite 15-second musical excerpts, from any genre or time period. For each selection, respondents were asked: "Why do you love the excerpt? Try to be as specific and detailed as possible (music theory terms are encouraged but not required)." Classical selections were dominated by a very small number of composers, while the pop and jazz artists were diverse. A thematic coding of the respondents' comments found that the most common themes were melody (34.2% of comments), harmony (27.2%), and sonic factors: texture (27.6%), instrumentation (24.3%), and timbre (12.5%). (Rhythm (19.5%) and meter (4.6%) were less present in the comments.) The comments cite simplicity three times more than complexity, and energy gain 14 times more than energy decrease, suggesting that people's favorite excerpts involve simple moments of energy gain or "build-up". The complete FAV Corpus is publicly available online at EthanLustig.com/FavCorpus. We will discuss future possibilities for the corpus, including potential directions in the spaces of machine learning and music recommendation.

1. INTRODUCTION

Why do we like the music we like? Perusal of the range of options in a record store or streaming platform, or of live music offerings in a large city, shows the enormous diversity of musical taste among individuals. Research has probed some of the factors involved in this variability, such as gender, age, personality, social identity, cultural background, and musical training [1-6]. Still, there seems to be general agreement that particular pieces of music are especially "good." Certain hymns, Christmas carols, folk songs, and classical pieces remain favorites across decades and centuries; certain popular songs cause world-wide and lasting explosions of enthusiasm. It seems, also, that specific sections or moments within these pieces are especially pleasurable, giving rise to what are sometimes called *peak experiences* [7-8]. Our own personal reflections certainly confirm this, and anecdotally, there seems to be at least some agreement as to what the "best" moments of a piece are. But what makes a certain part of a piece especially enjoyable?

Music psychology has begun to address this issue, though in tentative and exploratory ways. Most of this research has focused on the physiological manifestations of peak experiences, such as chills, which have been shown to correlate with pleasure. A pioneering study by Sloboda [9] asked participants to identify passages causing strong physiological effects—what he called "thrills" (p. 110, after [10])—and to describe the nature of those responses. More recent studies follow Sloboda's model in having participants identify pieces that cause physiological responses, especially chills, and then probing the possible causes and correlates of these responses: neurological correlates [11, 12], musical elicitors [13, 14], and self-reported perceptual correlates such as the perceived sadness or happiness of the music [15]. With regard to musical elicitors of chills, studies have found many factors including sequences, appoggiaturas, new or unexpected harmonies, crescendi, climaxes, sudden dynamic or textural changes, and entrances of instruments [9, 13-15]. Also deserving mention is a large project by Gabrielsson and Wik [16] focusing on the effects (physical, emotional, and cognitive) of "strong experiences" of music (p. 158). Musical elicitors are mentioned only briefly and in very general terms: "instruments, rhythm, melody, harmony, musical form, performance qualities etc." (p. 198; see also [17], p. 568).

In this study we offer a novel approach to the study of peak experiences and the musical factors that elicit them. In contrast to the exploratory research cited above, our study takes a systematic, survey-based approach. Our conception of peak experience is close to Maslow's [7]—a primarily internal albeit not physiological, intensely positive experience—and falls within the fairly broad range of ways that the term is used [8].

Our project differs from most research on peak experiences by focusing on passages of music directly reported to be strongly liked, rather than those causing chills and other physiological responses. While chills have generally been shown to coincide with pleasurable experiences [12, 13], they may not always do so; conversely, one can certainly get great enjoyment from a musical passage without experiencing chills.

In a survey, 140 respondents identified favorite musical excerpts. Respondents also provided free-response comments explaining their choices, and we provide a content

© E. Lustig and D. Temperley. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** E. Lustig and D. Temperley, "The FAV Corpus: An audio dataset of favorite pieces and excerpts, with formal analyses and music theory descriptors", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.*, Milan, Italy, 2023.

analysis of these comments. We also present a publicly available corpus, the FAV Corpus, which includes audio files of excerpts and complete pieces, formal analyses of a subset of the pieces, and the respondents' free-response comments.

2. METHOD

2.1 Participants

In 2017, 140 students at the University of Rochester (New York) were given a survey regarding their favorite musical excerpts. Approximately 85% of the respondents to the survey were students at the Eastman School of Music (a division of the university) and were therefore music majors. The remaining 15% were students in an introductory music psychology course; while students in this course were mostly not music majors, the course required basic knowledge of music theory as a prerequisite. While students with music-theory training may not be representative of the broader population, we deliberately chose them for their ability to articulate the musical reasons for their preferences, with regard to matters such as harmony, rhythm, and form. On average, the respondents had 11.1 years of music training on a musical instrument (including voice) (*SD* = 4.2). There were 73 females, 63 males, and four who preferred not to say. The average age of the respondents was 19.7 years old, with a range of 17 to 29 years old (*SD* = 1.9). Respondents received extra credit points in their courses for participation. The survey received ethical approval by the Institutional Review Board of the University of Rochester.

2.2 Data Collection

The survey asked each respondent to identify "three of your favorite excerpts of music... in any style and from any time period." For each excerpt, they were instructed to provide a URL (web address) to a recording of the piece/song/movement on YouTube or Spotify. We used the phrase "piece/song/movement" to avoid stylistic bias, but hereafter we will refer only to "pieces." Respondents were then asked to "identify the 15-second excerpt that's your favorite" by providing start- and end-points for the excerpt in relation to the recording. The choice of 15 seconds was fairly arbitrary. We chose it, in part, because it roughly corresponds to the length of some of our favorite musical passages.

Following each selection, respondents were prompted to write a response to the following question: "Why do you love the excerpt? Try to be as specific and detailed as possible (music theory terms are encouraged but not required)." We take the term *love* to indicate a high degree of liking or preference, similar in meaning to *enjoy* or *greatly like*. Our mention of "music theory terms" was aimed at encouraging respondents to identify the musical features giving rise to their preferences. There is a possible downside to this wording; by drawing attention to our own music-theoretical background, it may have steered respondents toward pieces or excerpts that they thought were theoretically "respectable" in some way. However, the huge stylistic variety of the chosen excerpts (described below), including many from very recent popular music, suggests to us that this was not a concern for many respondents. Additionally, respondents were asked to choose between either "I enjoy this excerpt much more than the other parts of the piece" or "I enjoy this excerpt about as much as the other parts of the piece."

2.3 Creating the Corpus

Recordings of the complete pieces provided by the respondents were extracted from the YouTube/Spotify URLs and saved as WAV audio files; audio files were also made of each preferred 15-second excerpt. In some cases, the beginning of the internet recording did not correspond to the true beginning of the piece. To adjust for this, any time before the beginning of the piece was subtracted from the timepoints of the preferred excerpt, so that the adjusted timepoints indicated the excerpt's location in relation to the piece. In about 8% of cases, the chosen excerpt was not exactly 15 seconds long, but usually just a few seconds shorter or longer. In such cases, the excerpt was converted to a 15-second excerpt with the same midpoint as the chosen excerpt. (For example, 0:00-0:25 would be converted to 0:05-0:20.) For more detail about this process, see [18].

The corpus, which we call the FAV Corpus, is publicly available at EthanLustig.com/FavCorpus. The corpus contains 420 items (three excerpts from each of the 140 respondents). A spreadsheet indicates, for each item, (a) the respondent's number, which had been assigned arbitrarily, (b) the excerpt number for that respondent (1, 2, or 3), (c) the artist and title of the piece, (d) the style and historical era or year (explained below), (e) the duration of the piece, (f) the timepoints of the preferred excerpt, (g) whether the respondent indicated that they enjoyed the excerpt "much more than" [A] or "about as much as" [B] the rest of the piece, and (h) the respondent's comment about why they liked the excerpt. In what follows, we indicate excerpts by respondent and excerpt number; for example, Respondent 1's three excerpts are 1_1, 1_2, and 1_3. We also provide sound files for both the complete pieces and the preferred 15-second excerpts.[1]

3. RESULTS

3.1 Stylistic Content of the Corpus

The distribution of styles and artists in the corpus was examined. While this is not the main focus of the current study, it provides a window into the musical tastes and passions of students at an American music school in 2017 (recall that roughly 85% of respondents were music students). Each excerpt was categorized as classical (49.5%), pop (41.8%), or jazz (8.7%). For most excerpts, classification was clear; there were a few borderline cases, such as jazz-rock fusion pieces. The most popular artists in the survey

[1] Due to reasons such as invalid web links, respondent errors, etc., the actual corpus contains 399 excerpt audio files and 402 piece audio files. (See EthanLustig.com/FavCorpus for details.)

are listed in Table 1, with the number of excerpts for each. Following convention, for classical works, we identify the composer as the artist; for jazz and pop, we identify the performer(s) as the artist. Table 1 alone might give the impression that respondents strongly favored classical music, but the style statistics just cited show otherwise; the preponderance of classical composers in Table 1 indicates, rather, that classical selections were dominated by a small handful of artists, while pop and jazz selections were much more widely dispersed.

Composer	# Excerpts
Bach	17
Brahms	14
Beethoven	12
Tchaikovsky	10
Rachmaninov	9
Mahler	6
Kendrick Lamar	6
Debussy	6
Sibelius	5
Handel	5

Table 1. Artists (composers/performers) most represented in survey.

Among the classical excerpts, 12.1% were from the Baroque period (1600-1749), 8.5% Classical (1750-1819), 31.7% Romantic (1820-1899), and 47.7% 20th/21st-century (1900-present). (Each composer was assigned to a single period, based on their years of greatest activity.) Again, the large number of 20th/21st-century selections is not reflected in Table 1 since they are distributed over a much larger number of composers. We also observed that many of these 20th/21st-century composers were toward the conservative end of the stylistic spectrum; the most popular was Rachmaninoff, with nine excerpts. For the pop and jazz selections, we identified the year of release of each recording. The pop selections strongly favored recent music: 69.0% were from 2010–2017 (more than half of these from 2016–2017 alone), and 17.9% from the 2000s. Jazz selections had a weaker bias toward recent music, with 31.4% of selections from 2010 through 2017.

3.2 Formal Analysis

One of us (David Temperley) did a formal analysis of a subset of pieces in the corpus. He did not know which excerpts were preferred when doing the analysis. The subset consisted of pieces in which respondents had said that they liked their preferred excerpt "much more than" the rest of the piece; this yielded a set of 127 pieces (about 30% of the survey responses).[2] The recordings of the pieces were divided into sections to the nearest second, and the sections were given formal labels, as the genre warranted (for instance, P = primary theme for a sonata-form piece; V (verse) and CH (chorus) for pop songs). It was assumed that each section continued until the beginning of the next section, so that each piece was exhaustively partitioned into sections. As an arbitrary constraint to simplify the analysis, no section was allowed to be less than 15 seconds long. Two main criteria were used for determining the location of formal sections: change and repetition. A significant change in any musical parameter, such as harmony, melody, instrumentation, texture, meter, or rhythmic pattern, was considered to make a good candidate for a section break. Repetition could also define sections: for instance, the return of the opening theme in a sonata-form movement might define a new section beginning even in the absence of obvious local changes. Repetition of the same label signified exact or slightly modified repetition; for example, V would be used for two verses of a pop song, with different lyrics and perhaps some changes in instrumentation, but mostly similar melody and harmony. For more substantially modified repetitions, numbers were used (e.g., V1 and V2 for two verses that had significantly different melody or harmony). See [18] for more detail about the annotation system.

We analyzed the preferred excerpts in relation to their location within the piece. First, we wondered if people tend to choose excerpts that are near section boundaries. For each preferred excerpt, in the set of 127 excerpts for which formal analyses were available, we found the temporal distance between the midpoint of that excerpt and the closest formal section boundary; we then performed the same process for random 15-second excerpts from the same pieces, repeating the process 10 times to mitigate the effect of extreme values. (One piece had a 7-minute section that seemed to create outliers in the data; this piece was removed from the analysis.) Midpoints of preferred excerpts have an average (absolute) temporal distance from the nearest section boundary of 11.41 seconds, while for midpoints of random excerpts, the distance is 13.67 seconds—a modest but significant difference ($t(168.73) = -2.46$, $p < 0.01$). Thus, preferred excerpts show a slight tendency to be located near formal boundaries. A total of 49.2% of the preferred excerpts actually contain a section boundary; among the random excerpts, only 37.6% do. We then reanalyzed the same distances as signed values, to see whether preferred excerpts tend to be near the beginning or end of a formal section. For preferred excerpts, the mean signed difference between the midpoint and the nearest boundary is 2.90 (i.e., on average, the midpoint occurs 2.90 seconds after the boundary), significantly greater than zero (one-sample *t*-test, $t(125) = 2.22$, $p < 0.05$). This indicates a slight tendency to choose excerpts near the beginning of a section rather than near the end, or, perhaps, overlapping more with the beginning of a section than with the end of the previous one.

Finally, we examined the location of each excerpt in relation to the piece as a whole. For this analysis, we used all 399 excerpts in the corpus. Each excerpt received a value for its proportional position in the piece, where 0 would be at the very beginning, and 1 would be at the very end. The

[2] Altogether there were 137 eligible pieces, 10 proved impossible to analyze into formal sections, because there was no large-scale repetition and no clear moments of change demarcating reasonably-sized sections. Some of these were contemporary pieces; others were Baroque pieces, for example imitative textures with a rapid or seamless alternation between subject entries and episodes.

mean value was 0.46; clearly there was not a strong bias toward choosing excerpts early or late in the piece.

3.3 Content Analysis of Comments

As mentioned earlier, respondents were asked to comment on their reasons for liking each excerpt in their own words. Responses varied from a few words to several sentences. While a few responses were flippant or minimal, a great many respondents showed enthusiasm for the task and took considerable effort in explaining their choices. We did a content analysis of the respondents' comments. One of us (David Temperley) coded all 420 comments, identifying 17 themes that seemed to appear repeatedly in the comments. We then provided a list of the 17 themes and their definitions (Table 2) to an independent coder (a music theory Ph.D. student at the Eastman School of Music) and asked him to assign themes to the comments using that list. Each comment could be tagged with any number of themes (including zero). In choosing themes, both coders aimed to represent the respondents' actual reasons for liking the excerpts, as opposed to aspects mentioned simply to aid reference, although this distinction was not always easy to make. For example, a comment like "I love the violin melody" was encoded as MEL (melody) rather than INS (instrumentation).

BIO Autobiographical connection: references to the respondent's past experience with the piece or excerpt, OR incidents in their life that it reminds them of for any reason.
COM (+/-) Complexity (or its opposite, simplicity).
DYN (+/-) Dynamics.
EN (+/-) Energy. Energy level in music is thought to be conveyed by such as dynamics, register, rhythmic activity, and textural thickness; an increase in any of these dimensions could create a rise in energy. However, when the change is described in these more specific terms (e.g. dynamics) it can be coded in that way; EN should be reserved for more general descriptions of energy change or level, e.g. "buildup" or "climax".
HAR Harmony: includes harmonic progression, function, or chord quality; also tonality (e.g. modulation), mode (major/minor), and dissonance/consonance.
INS Instrumentation: choices of instrument or instrument combinations; also includes general uses of an instrument (e.g. "I like the clarinet in a high register"), or special timbral effects prescribed by composer, e.g. extended techniques; also synthesized parts in popul music textures. (Compare to TIM).
INT Interpretation (e.g. expressive timing; also gener statements about beauty/expressiveness of a performance or quality of performer).
LYR Lyrics.
MEL Melody: the main melody in this particular part of the piece. Also includes improvised solos, e.g. in jazz.
MET Meter (incl. tempo).
PHY Mentions of a physical or physiological response to the music.
RET Return of earlier thematic material.
RH Rhythm. Includes references to general rhythmic feel, e.g. "groove".
SUR Explicit mentions of surprise or denial of expectation.
TEX Texture: a catch-all category including aspects of pitch-rhythmic patterns other than melody, such as details of accompaniment or bass lines, chord voicings, or polyphonic patterns.
TIM Timbre: when credited to performer (e.g. a singer's tone), or synthesized/electronic sounds that are not a consistent part of the texture. (Compare to INS).
VIR Virtuosity (or just proficiency, i.e. playing a very difficult bit accurately; also intonation).

Table 2. Themes and definitions used in content analysis.

Agreement between the two coders regarding the assignment of each theme was measured using Cohen's kappa, where 1.0 would indicate that the two coders assigned the theme to exactly the same comments. Agreement levels varied between 0.37 and 0.84, depending on the theme, and thus were mostly in the range of *moderate* or *substantial* according to Landis and Koch's [19] rubric. In what follows we discuss the results of this content analysis. We also analyzed word frequencies in the comments, grouping together similar words such as "simple," "simpler," and "simplicity". We include some results of that analysis in the following discussion.

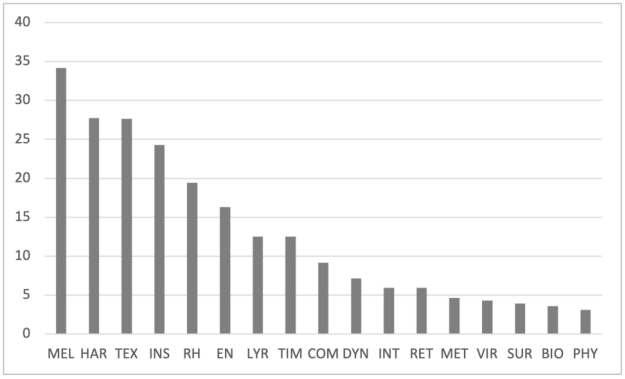


Figure 1. The percentage of respondents' comments identified with themes in the content analysis. For explanation of abbreviations, see Table 2.

Figure 1 shows the percentage of occurrences of each theme in the comments. The counts of each theme were averaged between the two coders. The frequent mentions of melody (MEL, occurring in 34.2% of comments) and harmony (HAR, 27.2%) indicate the importance of the pitch domain in respondents' preferences. Rhythmic factors—rhythm (RH, 19.5%) and meter (MET, 4.6%)—were less important, though it should be remembered that melody has a rhythmic aspect as well. Notably, the word "groove" occurred 20 times—confirming the widely held view that this is a significant factor in musical enjoyment [20, 21]. What might be called sonic factors were also mentioned frequently: texture (TEX, 27.6%), instrumentation (INS, 24.3%), and timbre (TIM, 12.5%). There were comparatively few mentions of performance aspects: interpretation (INT, 6.0%) and virtuosity (VIR, 4.3%).

Given that a large majority of respondents were majoring in classical music performance, we had expected these factors to weigh more heavily. Lyrics (LYR) were mentioned in 12.5% of comments, and autobiographical factors (BIO, connections with the respondents' life experience) in just 3.6%. The PHY theme, physiological responses (such as chills), was mentioned in just 3.1% of comments. It is possible that the survey instructions—which encouraged the use of music-theory terms—steered respondents' attention towards musical features and away from autobiographical and physiological factors.

Three of the themes—complexity (COM), energy (EN), and dynamics (DYN)—were parametric: They could be subscripted as "+" (indicating an increase or relatively high level) or "–" (a decrease or relatively low level), though this was optional. While there were 6.5 instances of COM+ in the comments, there were 22 instances of COM– (again, theme counts reported here and below are averaged across the two coders' analyses). This result suggests that respondents favored moments of relatively low or decreasing complexity. Our analysis of word frequencies also supports this view: "simple" (and related words) occurs 32 times in the comments, while "complex" (and its variants) only occurs 11 times. Related to this, the words "tension/tense" and "resolution/resolve" were used about equally often (19 and 18 times, respectively). However, seven of the comments mentioning tension refer specifically to the resolution of the tension (sometimes using other words like "relax," "release," or "relief"); in the remaining cases, the tension seems to be valued in itself.

The energy (EN) theme shows an even stronger parametric tilt than complexity: 35.5 of its mentions are EN+, while only 2.5 are EN–. Energy is often treated as more or less synonymous with the arousal/activation dimension in Russell's [22] two-dimensional model of emotion, and this in turn has been associated with musical parameters such as loudness, pitch register, and rhythmic activity [23].[3] Note from Table 2 that this theme reflects general references to energy, as opposed to mentions of energy-invoking musical dimensions such as dynamics, rhythm, or texture. The dynamics (DYN) theme also showed a parametric tilt, marked "+" eight times and "–" only three times. Analysis of word frequencies shows further evidence of a preference for increasing energy. For example, the word "build" and related words such as "build-up" occurs 47 times. It is not obvious what the opposite of "build" would be, indicating a general *decrease* in energy level; one thinks of such words such as "decrease," "decline," "fade," "wane," "subside," and "dwindle." None of these words occurred even once, except "fade," which occurred just three times.[4] Several other frequent word categories indicate an increase or peak in energy, such as "climax/climactic" (used 28 times), "power(ful)" (27 times), and "crescendo" (9 times; "diminuendo" is never used and "decrescendo" just once).

In this connection, a result from our analyses of formal structures, described earlier, is relevant. In pop songs, which nearly always contain both choruses and verses, respondents' preferred excerpts were more often in choruses (13 times) than verses (7 times). (Recall that our analysis of formal structures included only about 30% of the survey responses.) Respondents' comments also mentioned choruses (44 times) much more often than verses (18 times).[5] It has been observed that choruses tend to be higher than verses in the "energetic" dimensions mentioned above, such as pitch register and textural thickness [24, pp. 39-40]. Thus, several patterns in our data point to an increase in energy as an important elicitor of musical pleasure.

Perusal of the comments suggests other possible themes as well. For instance, many comments contain terms or phrases that could be described as emotional. In the first 20 comments, we see "aggressive" (1_2), "raw emotion" (2_3), "intensity" (2_3), "exciting" (4_2, 6_2), "[the singer] let[s] emotions loose," (5_2) "dramatic" (6_1), and "triumphant" (6_2). In many cases, such terms are used to describe a specific aspect of the music that could also be encoded in some other way: for example, "a triumphant theme" (MEL); "the buildup is very exciting" (EN). Another issue is the distinction between induced and perceived emotion [25]. Sometimes the distinction is clear—"It is insanely happy" is perceived emotion, "[It] always makes me so happy" is induced emotion—but not always: if a passage is described as "exciting" or "relaxing," is that induced or perceived emotion? If induced emotion is included in the "emotion" theme, one could potentially include a large number of comments implying a positive emotional reaction: for example, "I love the cellist's interpretation." Indeed, one might say that such a reaction is implicit in all of the comments, given the nature of the task.

4. DISCUSSION

In our study, 140 college students, mostly music students, identified three of their favorite 15-second passages of music. One result emerging from our analysis of the survey comments was a preference for passages that increase in energy—often described by respondents as "builds" or "build-ups," or as sections that "build." As noted earlier, energy in music is generally associated with parameters such as loudness, pitch register, and rhythmic activity. It also seems intuitive to us, although this does not seem to have been widely studied, that textural thickness is also associated with energy, perhaps partly because a thickening of texture implies greater loudness, whether or not the loudness actually increases. Our finding that increases in

[3] In experiments on music and emotion, manipulations in the temporal dimension usually involve changing the speed of a melody, and are therefore described (correctly) as variations in tempo (for a survey, see [23]). Within a piece, however, the tempo (i.e., the speed of the main beat) rarely changes, except for small fluctuations; temporal variation is more likely to involve changes in rhythmic values (e.g., from a quarter-note texture to a 16th-note texture). In both of these cases, though, the variations involve a change in the temporal density of events; if an increase in tempo conveys an increase in energy, it seems likely that an increase in rhythmic activity over a fixed tempo would also do so.

[4] Some of these words, such as "build," "decrease," and "fade," could be either nouns or verbs; we counted both, including all verb forms. The word "drop" is also of interest; it occurs 12 times, as noun or verb, but only five of those uses could be taken to refer to energy level. Sometimes the term is used to refer to the re-entrance of the kick drum in a pop or EDM song.

[5] One might wonder if choruses are more frequent than verses in our corpus, and therefore take up more time. Actually they do not: choruses take up a total of 1769 seconds, in the portion of the corpus that was formally analyzed; verses take up a total of 1977 seconds.

energy are often pleasurable accords well with other work on peak musical experiences that has linked them to crescendi and increases in texture [13-15, 26]. It also appears that there is a strong preference for passages perceived as having relatively low or decreasing complexity and tension, compared to passages perceived to be high in complexity or tension. This is in line with Meyer's [27] observation that in music, "The greater the buildup of suspense, of tension, the greater the emotional release upon resolution" (p. 28) and Huron's [28] idea of "contrastive valence" (p. 39).

Earlier studies have found that a wide range of factors affect peak experiences [9, 13, 14], and this is apparent from the free-response comments in our survey. The single most common theme in the comments was melody. While it is hardly news that people like a good melody, this result draws our attention to the huge importance of this factor; the question of what makes a melody good is one that music theory and music psychology are still a long way from answering. Our corpus might provide a useful starting point for an exploration of this topic. Other frequent themes in respondents' comments—such as harmony, instrumentation, rhythm, and texture—also point to factors that greatly influence listeners' preferences; how they do so is, at present, largely mysterious.

4.1 Future Directions

In terms of future directions, the first avenue of exploration could be expanding the existing dataset. The survey could be re-run online, and globally, with many more participants, increasing sample size and statistical power, as well as diversifying the participant set. Instead of three songs and excerpts, many more songs and excerpts could be requested from each participant, allowing for better trend analysis *within* participants, to potentially identify different listener-types. The usefulness of this kind of data for the music-recommendation space, and associated industry applications, is clear [29].

As the corpus grows in size, the potential for using machine learning and related methods (which tend to excel with larger datasets) to analyze the data becomes more viable. An acoustical signal-analysis-based approach, using the many tools available in the field of music information retrieval, for instance, could be applied to the corpus, to determine which audio features (e.g. spectral flux, dissonance, loudness, etc.) are determinative of the favorite excerpts as compared to random controls from the same pieces. This acoustical approach could be effectively combined with a symbolic, music-theoretic approach.

In fact, even without venturing outside of the symbolic space, there is immense potential for further coding and analysis of corpus features such as scale-degree distributions, metric position, harmonic root patterns, and so forth, akin to the statistical work applied to the Rolling Stone Corpus [30, 31]. This computational approach to the corpus could be supplemented by a more humanistic, analytical approach in which more speculative and traditional analysis is conducted to attempt to understand why these particular excerpts are so powerful. For instance, given the overwhelming emphasis on pitch (melody and harmony) in participants' comments, it would be interesting to determine the melodic, harmonic, rhythmic, and contrapuntal structures characteristic of the excerpts in the corpus; and what distinguishes a favorite excerpt from a non-favorite excerpt in the same piece.

Another possible direction for future research could be to measure the energy and complexity trajectories of the pieces in our corpus. While energy can be measured using low-level spectral features such as root-mean-square (RMS) acoustic energy, some efforts have been made to create more sophisticated predictors of perceived musical energy using combinations of features [32, 33]. Such algorithms could be applied to our corpus. Meanwhile, measuring complexity (especially in an automatic way) presents more of a challenge. Complexity—in its information-theoretic sense—is inherently subjective, since it depends on the listener's expectations, which in turn can vary widely depending on their musical experiences. Furthermore, complexity presumably depends heavily on patterns of pitch and rhythm, which cannot yet be reliably extracted from polyphonic audio [34]. For classical pieces, MIDI encodings could be used, but for popular songs, transcriptions would need to be created. Once these problems were solved, it might be possible to create measures of complexity using probabilistic models (such as Markov models); indeed, there have been interesting efforts in this direction, though they relate only to melody [35, 36].

Another intriguing area is the correlation of personality, personal values, and socio-economic data with music taste [1-6]. An expanded iteration of the survey could perhaps include a personality inventory and collect socio-economic data, building a more holistic and accurate model of music taste.

We hope that the current study has taken a small step toward advancing our understanding of peak musical experiences, and that our publicly available corpus will be useful to other researchers in this area, as we continue to answer the question: why do we like the music that we like?

5. REFERENCES

[1] D. J. Hargreaves, C. Comber, and A. Colley, "Effects of age, gender, and training on musical preferences of British secondary school students," *Journal of Research in Music Education*, vol. 43, no. 3, pp. 242-250, 1995.

[2] P. J. Rentfrow and S. D. Gosling, "The do re mi's of everyday life: The structure and personality correlates of music preferences," *Journal of Personality and Social Psychology*, vol. 84, no. 6, pp. 1236–1256, 2003.

[3] S. Manolios, A. Hanjalic, and C. C. S. Liem, "The influence of personal values on music taste: Towards value-based music recommendations," *Proceedings of the 13th ACM Conference on Recommender Systems*, pp. 501-505, 2019.

[4] T. Schäfer and C. Mehlhorn, "Can personality traits predict musical style preferences? A meta-analysis,"

Personality and Individual Differences, vol. 116, pp. 265-273, 2017.

[5] A. Mohan and E. Thomas, "Effect of background music and the cultural preference to music on adolescents' task performance," *International Journal of Adolescence and Youth*, vol. 25, no. 1, pp. 562–573, 2020.

[6] B. I. L. M. Mendis *et al.,* "Exploration of music preferences among the socioeconomic stereotypes: A cross-sectional study," *Journal of Advanced Research in Social Sciences*, vol. 4, no. 4, pp. 1–18, 2021.

[7] A. H. Maslow, *Religions, Values, and Peak-experiences*. Columbus, OH: Ohio State University Press, 1964.

[8] J. Whaley, J. Sloboda, and A. Gabrielsson, "Peak experiences in music," in *The Oxford Handbook of Music Psychology,* S. Hallam, I. Cross, and M. Thaut, Eds. Oxford, UK: Oxford University Press, 2009, pp. 452–61.

[9] J. Sloboda, "Music structure and emotional response: Some empirical findings," *Psychology of Music*, vol. 19, no. 2, pp. 110-120, 1991.

[10] A. Goldstein, "Thrills in response to music and other stimuli," *Physiological Psychology,* vol. 8, pp. 126-129, 1980.

[11] A. J. Blood and R. J. Zatorre, "Intensely pleasurable responses to music correlate with activity in brain regions implicated in reward and emotion," *Proceedings of the National Academy of Sciences*, vol. 98, no. 20, pp. 11818–11823, 2001.

[12] V. N. Salimpoor, M. Benovoy, K. Larcher, A. Dagher, and R. J. Zatorre, "Anatomically distinct dopamine release during anticipation and experience of peak emotion to music," *Nature Neuroscience*, vol. 14, no. 2, pp. 257–262, 2011.

[13] O. Grewe, F. Nagel, R. Kopiez, and E. Altenmüller, "Listening to music as a re-creative process: Physiological, psychological, and psychoacoustical correlates of chills and strong emotions," *Music Perception*, vol. 24, no. 3, pp. 297–314, 2007.

[14] S. Bannister, "A survey into the experience of musically induced chills," *Psychology of Music*, vol. 48, no. 2, pp. 297–314, 2020.

[15] J. Panksepp, "The emotional sources of "chills" induced by music," *Music Perception*, vol. 13, no. 2, pp. 171–207, 1995.

[16] A. Gabrielsson and S. L. Wik, "Strong experiences related to music: A descriptive system," *Musicae Scientiae*, vol. 7, no. 2, pp. 157–217, 2003.

[17] A. Gabrielsson, "Strong experiences with music," in *Handbook of Music and Emotion*, P. Juslin and J. Sloboda, Eds. Oxford, UK: Oxford University Press, 2010, pp. 547-574.

[18] E. Lustig, "The effect of perceived complexity and formal location on musical preference," Ph.D. dissertation, Dept. Music Theory, University of Rochester, Rochester, NY, 2021.

[19] J. R. Landis and G. G. Koch, "The measurement of observer agreement for categorical data," *Biometrics*, vol. 33, no. 1, pp. 159–174, 1977.

[20] P. Janata, S. T. Tomic, and J. M. Haberman, "Sensorimotor coupling in music and the psychology of the groove," *Journal of Experimental Psychology: General*, vol. 141, no. 1, pp. 54-75, 2012.

[21] M. Witek, E. Clarke, M. Wallentin, M. Kringelbach, and P. Vuust, "Syncopation, body-movement and pleasure in groove music," *PLoS ONE*, vol. 9, no. 4, 2014.

[22] J. A. Russell and L. F. Barrett, "Core affect, prototypical emotional episodes, and other things called emotion: Dissecting the elephant," *Journal of Personality and Social Psychology*, vol. 76, no. 5, pp. 805–819, 1999.

[23] A. Gabrielsson and E. Lindström, "The role of structure in the musical expression of emotions," in *Handbook of Music and Emotion*, P. Juslin and J. Sloboda, Eds. Oxford, UK: Oxford University Press, 2010, pp. 367–400.

[24] T. de Clercq, "Sections and successions in successful songs: A prototype approach to form in rock music," Ph.D. dissertation, Dept. Music Theory, University of Rochester, Rochester, NY, 2012.

[25] P. Evans and E. Schubert, "Relationships between expressed and felt emotions in music," *Musicae Scientiae*, vol. 12, no. 1, pp. 75–99, 2008.

[26] B. K. Hurley, P. A. Martens, and P. Janata, "Spontaneous sensorimotor coupling with multipart music," *Journal of Experimental Psychology: Human Perception and Performance*, vol. 40, no. 4, pp. 1679–1696, 2014.

[27] L. Meyer, *Emotion and Meaning in Music.* Chicago, IL: University of Chicago Press, 1956.

[28] D. Huron, *Sweet Anticipation: Music and the Psychology of Expectation.* Cambridge, MA: MIT Press, 2006.

[29] A. Agostinelli *et al.,* "MusicLM: Generating music from text," *arXiv preprint arXiv:2301.11325*, 2023.

[30] T. de Clercq and D. Temperley, "A corpus analysis of rock harmony," *Popular Music*, vol. 30, pp. 47-70, 2011.

[31] I. Tan, E. Lustig, and D. Temperley, "Anticipatory syncopation in rock: A corpus study," *Music Perception*, vol. 36, no. 4, pp. 353-370, 2019.

[32] A. Zils and F. Pachet, "Extracting automatically the perceived intensity of music titles," in *Proceedings of the 6th COST-G6 Conference on Digital Audio Effects (DAFX03)*, 2003.

[33] P. Wood and S. Semwal, "An algorithmic approach to music retrieval by emotion based on feature data," in *Proceedings of 2016 Future Technologies Conference (FTC)*, 2016, pp. 140-144.

[34] 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.

[35] T. Eerola, "Expectancy-violation and information-theoretic models of melodic complexity*," Empirical Musicology Review*, vol. 11, no. 1, 2016.

[36] 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.

LYRICWHIZ: ROBUST MULTILINGUAL ZERO-SHOT LYRICS TRANSCRIPTION BY WHISPERING TO CHATGPT

Le Zhuo[1] **Ruibin Yuan**[2,3] **Jiahao Pan**[4] **Yinghao Ma**[5] **Yizhi Li**[6] **Ge Zhang**[2,7] **Si Liu**[1]
Roger Dannenberg[3] **Jie Fu**[2] **Chenghua Lin**[6] **Emmanouil Benetos**[5] **Wenhu Chen**[7] **Wei Xue**[4] **Yike Guo**[4]
[1] Beihang University [2] Beijing Academy of Artificial Intelligence [3] Carnegie Mellon University
[4] Hong Kong University of Science and Technology [5] Queen Mary University of London
[6] University of Sheffield [7] University of Waterloo
zhuole1025@gmail.com, ruibiny@andrew.cmu.edu, fujie@baai.ac.cn

ABSTRACT

We introduce LyricWhiz, a robust, multilingual, and zero-shot automatic lyrics transcription method achieving state-of-the-art performance on various lyrics transcription datasets, even in challenging genres such as rock and metal. Our novel, training-free approach utilizes Whisper, a weakly supervised robust speech recognition model, and GPT-4, today's most performant chat-based large language model. In the proposed method, Whisper functions as the "ear" by transcribing the audio, while GPT-4 serves as the "brain," acting as an annotator with a strong performance for contextualized output selection and correction. Our experiments show that LyricWhiz significantly reduces Word Error Rate compared to existing methods in English and can effectively transcribe lyrics across multiple languages. Furthermore, we use LyricWhiz to create the first publicly available, large-scale, multilingual lyrics transcription dataset with a CC-BY-NC-SA copyright license, based on MTG-Jamendo, and offer a human-annotated subset for noise level estimation and evaluation. We anticipate that our proposed method and dataset will advance the development of multilingual lyrics transcription, a challenging and emerging task.

1. INTRODUCTION

Automatic lyrics transcription (ALT) is a crucial task in music information retrieval (MIR) that involves converting an audio recording into a textual representation of the lyrics sung in the recording. The importance of this task stems from the fact that lyrics are a fundamental aspect of many music genres and are often the main way in which listeners engage with and interpret a song's meaning. Additionally, ALT has numerous applications in the music industry, such as enabling better cataloging [1], music searching [2, 3], music recommendation [4], as well as facilitating the creation of karaoke tracks and lyric videos. Moreover, ALT can assist in various music-related research tasks, including sentiment analysis [5], music genre classification [1], lyrics generation, which is further used for music generation [6], security review, and music copyright protection. Thus, accurate and efficient ALT is essential for advanced MIR and the development of new music-related applications.

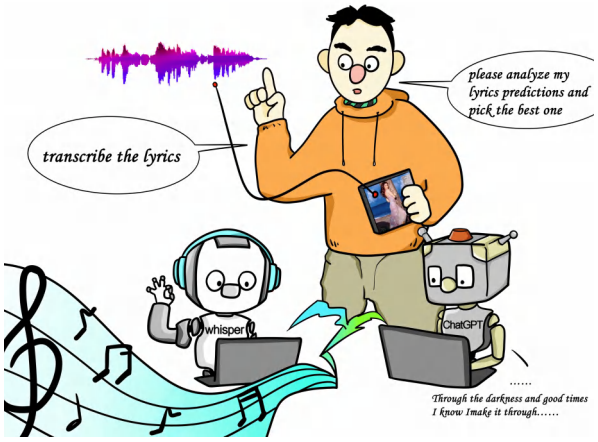


Figure 1. Concept illustration of the working LyricWhiz, where user prompts the two advanced models, Whisper and ChatGPT, to perform automatic lyrics transcription.

However, to date, no sufficiently robust and accurate ALT system has been developed. Even major commercial music streaming platforms still rely heavily on manually-annotated lyrics, incurring high costs. One key reason is the challenging nature of lyrics transcription. The diversity of singing styles and skills leads to varied timbres of the same pronunciation. Moreover, the phonemes in singing may be pronounced in vastly different ways, such as longer duration, tone changes, or even vowel substitutions, to accommodate the melody. Lastly, the inclusion of various music accompaniments across different genres makes it challenging to distinguish the vocal signals from other sounds. To surmount these challenges, a more robust ALT system is necessary, capable of outperforming existing models in diverse scenarios, including the transcription of multilingual lyrics.

Another significant factor hindering the progress of

© L Zhuo, R Yuan, and J Pan. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** L Zhuo, R Yuan, and J Pan, "LyricWhiz: Robust Multilingual Zero-shot Lyrics Transcription by Whispering to ChatGPT", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

ALT systems is the absence of large-scale singing datasets. Currently, only two relatively sizable datasets [7, 8] exist for ALT systems. However, all existing datasets are in English, with no multilingual datasets available. Besides, these datasets often have stringent copyright licensing restrictions, which significantly hampers their utilization by researchers. Consequently, developing a more comprehensive and representative dataset, encompassing multiple languages and without copyright issues, is essential for supporting the creation of a robust and accurate system.

In this paper, we present LyricWhiz, a novel method for automatic lyrics transcription. LyricWhiz surpasses existing methods on various ALT datasets, resulting in a significant reduction in WER for English lyrics and providing accurate transcription results across multiple languages. Our system is robust, multilingual, and training-free. To achieve these results, we combined two powerful models from their respective domains as shown in Figure 1: Whisper, a weakly supervised speech transcription model, and GPT-4, a large language model (LLM) from the ChatGPT family. Whisper acts as the "ear" while GPT-4 serves as the "brain" by providing contextualized output selection and correction with strong performance [9]. We further use LyricWhiz to build a multilingual lyrics dataset, named MulJam, which is the first large-scale, multilingual lyrics transcription dataset without copyright-related issues.

The contributions of our work are as follows:

- We propose a novel, robust, training-free ALT method, LyricWhiz, which significantly reduces WER on various ALT benchmark datasets, including Jamendo, Hansen, and MUSDB18, and is close to the in-domain state-of-the-art system on DSing.
- We introduce the first ALT system that can perform zero-shot, multilingual, long-form ALT by integrating a large speech transcription model and an LLM for contextualized post-processing.
- We create the first publicly-available, large-scale, multilingual lyrics transcription dataset with a clear copyright statement which eliminates further reviewing of the users and facilitates public usage. We provide a human-annotated subset to estimate noise levels and evaluate multilingual ALT performance.

2. RELATED WORK

2.1 Automatic Lyrics Transcription

Automatic lyrics transcription (ALT) is an essential task in music information retrieval and analysis, aiming to recognize lyrics from singing voices. It remains challenging due to facts such as the sparsity of training data and the unique acoustic characteristics of the singing voice that differ from normal speech. Traditional methods treat ALT in the automatic speech recognition (ASR) framework, which generally utilizes a hybrid of language model and acoustic model, e.g., HMM-GMM. Music-related characteristics have been used to further address these challenges [11–13].

Despite integrating domain-specific music priors into system designs, the data scarcity issue persists. Recently, some researchers have constructed datasets for end-to-end learning, which greatly advances ALT, but most datasets are either noisy (DALI [7, 14], Hansen [15], DAMP-MVP [1]); not large (Vocadito [16]); or not diverse in terms of genre and language (MUSDB18 [17], DSing [8]).

Recent rapid progress in ASR has greatly benefited ALT. Some work focuses on applying the ASR model architectures [18–20], such as the Transformers, to ALT, and other work leverages the vast amount of public annotated ASR datasets [19–21] to bridge between the speech and music data. For the first time, a recent study [22] transferred a large-scale self-supervised pre-trained ASR model, mus2vec 2.0, to the singing domain, and exhibited superior performance on multiple benchmark datasets. Nevertheless, this approach consists of pre-training, fine-tuning, and transfer learning phases, thereby remaining relatively complicated and still requiring singing datasets.

2.2 Weakly Supervised Automatic Speech Recognition

The paradigm of large-scale unsupervised pretraining and non-large annotated dataset finetuning has dominated end-to-end ASR research [23]. Well-known pretrained ASR models include contrastive learning based Wav2vec [24], Wav2vec 2.0 [25], HuBert [26], WavLM [27], Whisper [28], and Vall-E [29], which have performed impressively in various downstream tasks, including ASR and speech synthesis. Among them, Whisper has been most recognized for its ASR robustness across different datasets and its multilingual and multitasking capabilities, making Whisper potentially applicable to music tasks. Besides, specifically for ALT, pre-trained musical audio models including JukeBox [6], MusicLM [30], MULE [31], SingSong [32], music2vec [33], and MERT [34], may also contribute to achieving strong performance.

2.3 Chat-based Large Language Models

ChatGPT [2], a chat-based large language model (LLM), has found broad application in optimizing workflows across a variety of domains, including multimodal intelligence [35, 36]. Recent breaking AutoGPT [3] is even recognized as an embryonic form of artificial general intelligence. Inspired by these developments, LyricWhiz collaborates with both Whisper [28] and ChatGPT to optimize the workflow of ALT. Prompt engineering is known to be important to navigate LLMs to perform better [37]. LyricWhiz mainly adopts three primary strategies:

a) As shown in [38, 39], a well-formalized task description prompt can effectively improve ChatGPT's performance on downstream tasks with strict format requirements. We follow this empirical observation to strictly formalize the expected format of ALT post-processing outputs. We also refer to the prompt pattern catalog in [39] for an intuitive understanding of prompt engineering.

[1] https://zenodo.org/record/2747436#.ZDqBQOzML0o
[2] https://openai.com/blog/chatgpt
[3] https://github.com/Torantulino/Auto-GPT

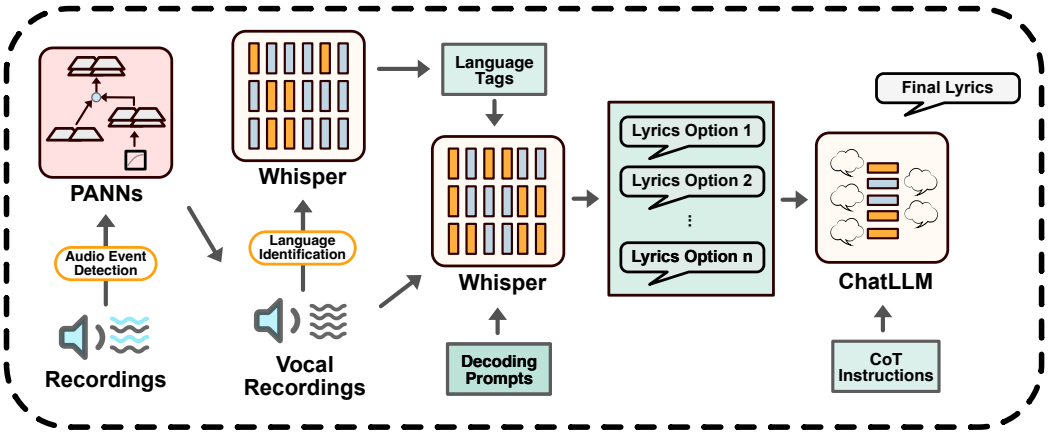


Figure 2. Framework of the proposed LyricWhiz. In the first stage, we employ PANNS [10], to detect audio events and filter out non-vocal recordings. In the second stage, we utilize the language identification module in Whisper to predict input audio language. We then construct language-specific prompts for Whisper and transcribe input audio multiple times. In the final stage, we request ChatGPT with CoT instructions to ensemble multiple predictions and generate the final lyrics.

b) Inspired by [40, 41], LyricWhiz utilizes prompt augmentation to ask ChatGPT to analyze the prompt and input lyrics, in order to select the most accurate prediction from multiple Whisper trials, which is done in the first phase as illustrated in Section 3.2. [41] designs a gradient-guided strategy to select prompts. By contrast, we simply feed ChatGPT with an instruction to select prompts for itself.

c) The importance of a well-designed CoT [42], which effectively divides a complicated task into several phases and designs specific prompts for each phase, is widely acknowledged for enhancing LLM performance. We also proposes a concise CoT strategy, depicted in Section 3.2.

3. METHODOLOGY

The overall framework of our method is presented in Figure 2. This section will provide an in-depth analysis of the design of the Whisper and ChatGPT components, and our multilingual dataset.

3.1 Whisper as Zero-shot Lyrics Transcriptor

In the Whisper [28] paper, the authors scaled the weakly supervised ASR to 680,000 hours of labeled audio data, which covers 96 languages and includes both multilingual and multitask training. This approach demonstrates high-quality results without the need for fine-tuning, self-supervision, or self-training techniques. By leveraging weak supervision and large-scale training, Whisper generalizes well to standard benchmarks and achieves robust speech recognition in various downstream tasks.

Motivated by this, we discovered that the weakly supervised Whisper model, trained on speech data, also excels in lyrics transcription within the music domain. We directly apply Whisper to transcribe lyrics of music from various genres, including pop, folk, rock, and rap, and find that the model consistently achieves accurate transcription results. The model excels at long-form transcription and is robust to different song styles, even for challenging genres such as rock and electronic music, where Whisper still provides reasonable results. We further test Whisper on multiple benchmark datasets for lyric transcription. The results indicate that Whisper, without any training or fine-tuning, can achieve or surpass SOTA performance across multiple lyric transcription datasets.

Upon analyzing the transcription results from Whisper, we observed that the model occasionally outputs content unrelated to lyrics, such as music descriptions, emojis, website watermarks, and YouTube advertisements. We attribute this to the weakly supervised training of Whisper on large-scale noisy speech datasets. To address this issue, we utilize the input prompt designed in Whisper as a prefix prompt to guide it toward the lyric transcription task. Unlike prompt designing philosophy in other large language models, Whisper's prefix prompt does not work well with explicit task instructions and has difficulty understanding lengthy explanations. In practice, we notice that using the simplest prompt, "lyrics:", effectively prevents the model from outputting descriptions of the music in most cases, resulting in a significant improvement in transcription results. Therefore, in the following sections, this prompt is consistently used for Whisper's transcription input.

Additionally, we apply post-processing tricks to Whisper's output, utilizing the model's predicted no-speech probability to handle situations where predictions are made despite the absence of vocals in the song. Specifically, we drop predicted lines of lyrics with a no speech probability greater than 0.9. This effectively filters out watermarks and advertisements, further enhancing the transcription results.

3.2 ChatGPT as Effective Lyrics Post-processor

Although we addressed some issues with Whisper's predictions through prompt design and post-processing, we still cannot avoid transcription translation errors, as well as grammatical and syntactical errors. Furthermore, due to the inherently stochastic nature of temperature scheduling in Whisper, the transcription predictions vary with each run, leading to fluctuations in evaluation metrics. To reduce this variance and enhance overall accuracy, we gen-

GPT-4 Instruction Prompt

Task: As a GPT-4 based lyrics transcription post-processor, your task is to analyze multiple ASR model-generated versions of a song's lyrics and determine the most accurate version closest to the true lyrics. Also filter out invalid lyrics when all predictions are nonsense.
Input: The input is in JSON format:
{"prediction_1": "line1;line2;...", ...}
Output: Your output must be strictly in readable JSON format without any extra text:
{
"reasons": "reason1;reason2;...",
"closest_prediction": <key_of_prediction>
"output": "line1;line2..."
}
Requirements: For the "reasons" field, you have to provide a reason for the choice of the "closest_prediction" field. For the "closest_prediction" field, choose the prediction key that is closest to the true lyrics. Only when all predictions greatly differ from each other or are completely nonsense or meaningless, which means that none of the predictions is valid, fill in "None" in this field. For the "output" field, you need to output the final lyrics of closest_prediction. If the "closest_prediction" field is "None", you should also output "None" in this field. The language of the input lyrics is English.

Table 1. Instruction prompt for GPT-4 contextualized post-processing. We decompose this task into three consecutive phases, inspired by Chain-of-Thought prompting. Note that lines in blue indicate additional prompts used exclusively for multilingual dataset construction.

erate 3 to 5 predictions for each input music under identical settings and employ ChatGPT as an expert in lyrics to ensemble these multiple predictions.

The crux of the problem lies in designing an effective prompt for ChatGPT to accomplish the ensemble task reasonably. As shown in Table 1, we first assign ChatGPT the role of a transcription post-processor, indicating that its task is to analyze multiple lyric transcription results and select the one it deems most accurate. We then stipulate that both input and output should be in JSON format to facilitate structured processing and provide detailed descriptions for each output field.

Drawing on the Chain-of-Thought in large language models for reasoning, we devised a concise thought chain for ChatGPT that decomposes lyrics post-processing into three consecutive phases. This involves first having ChatGPT analyze multiple lyric inputs and provide reasons for selection, then making a choice, and finally outputting the chosen lyric prediction. We test this approach using GPT-3.5 and the newly released GPT-4. The results demonstrate that using the analysis-selection-prediction prompt for ChatGPT's inference effectively enhances the final transcription results, with GPT-4 exhibiting a noticeably superior performance compared to GPT-3.5.

3.3 Multilingual Lyrics Transcription Dataset

Building upon the exceptional performance of the proposed framework in lyric transcription tasks, we further extend it to the challenging task of multilingual lyric transcription, introducing the first large-scale, weakly supervised, and copyright-free multilingual lyric transcription dataset. We utilize the publicly available MTG-Jamendo dataset for music classification, which comprises 55,000 full audio tracks, 195 tags, and music in various languages.

Dataset	Languages	Songs	Lines	Duraion
DSing [8]	1 (en)	4,324	81,092	149.1h
MUSDB18 [17]	1 (en)	82	2,289	4.6h
DALI-train [14]	1 (en)	3,913	180,034	208.6h
DALI-full [14]	30*	5,358*	-	-
MulJam (Ours)	6	6,031	182,429	381.9h

Table 2. Comparison between different lyrics transcription datasets. Our model operates with a longer window (~30s), resulting in fewer lines compared to other datasets.

Since the MTG dataset contains a considerable proportion of non-vocal music, we first employ PANNs [10], a large-scale pre-trained audio pattern recognition model, to detect audio events and filter out non-vocal music with vocal-related tag probabilities below a predefined threshold. This filtering method eliminates approximately 60% of the music, thereby substantially reducing the time and resources required for dataset construction. We then utilize Whisper to transcribe lyrics from the music.

As the music in the MTG dataset encompasses multiple languages, we first utilize the Language Identification module within Whisper to predict the language of input music. Based on the predicted language, we translate the prefix prompt "lyrics:" into the corresponding language for input, *e.g.*, "paroles" in French, and "liedtext" in German. After obtaining the transcription results, we discard lyrics that are too short or too long. When ensembling the prediction results with ChatGPT, we also incorporate the language of lyrics as an input condition in the prompt. Given the prevalence of nonsensical content in the transcription results, we additionally require ChatGPT to evaluate the validity of the transcribed lyrics in the prompt. If all input lyrics are deemed nonsensical, *e.g.*, all special Unicode characters, or extremely divergent, the transcription result for that piece of music is considered invalid and discarded.

To prepare the dataset for training, it is essential to conduct line-level annotation. Timestamps can be obtained from the output of Whisper by aligning the lyrics both before and after ChatGPT processing. For the alignment of strings, the Levenshtein distance [43] is employed. To exclude aligned lines of lower confidence, the distance is normalized, setting a threshold at 0.2. The quality of annotation is further enhanced through two subsequent filtering stages. In the first stage, lines that exhibit unusually high character rates, exceeding 37.5 Hz, are eliminated. The second stage encompasses another Whisper iteration; segments yielding a transcription of "Thank you." are excluded. These segments, which typically represent instrumental sections, are believed to originate from Whisper's training on data similar to video transcripts.

Following the construction process outlined above, we ultimately obtained a multilingual lyric transcription dataset, MulJam, consisting of 6,031 songs with 182,429 lines and a total duration of 381.9 hours. The dataset's statistical information and comparisons with existing ALT datasets are presented in Table 2.

Method	Jamendo	Hansen	DSing
TDNN-F [8]	76.37	77.59	19.60
CTDNN-SA [44]	66.96	78.53	14.96
Genre-informed AM [12]	50.64	39.00	56.90
MSTRE-Net [13]	34.94	36.78	15.38
DE2-segmented [45]	44.52	49.92	-
W2V2-ALT [22]	33.13	18.71	**12.99**
LyricWhiz (Ours)	**24.25**	**7.85**	13.78
w/o ChatGPT Ens.	28.18	8.07	15.22
w/o Whis. Prompt	33.21	8.75	13.40

Table 3. The WERs (%) of various ALT systems, including ablation methods, on multiple datasets. Note that W2V2-ALT is an in-domain baseline that natively train on DSing. The results of our method on Jamendo, Hansen are obtained from full-length transcription results, and the results on DSing are obtained from utterance-level segments.

To our best knowledge, MulJam is the first publicly available large-scale dataset for multilingual lyrics transcription without copyright restrictions. While DALI [7] is another large-scale music dataset featuring multilingual lyrics, its restricted access and strict licensing requirements limit its applicability for downstream tasks. In contrast, MulJam is free from copyright-related constraints and can be utilized without approval, as the audio can be legally downloaded directly from public sources without the need for approval, making it easily accessible. This even includes audio that is permitted for use in the development of commercial software. Researchers are permitted to legally modify our dataset for derivative works and redistribution, provided they cite our work and adhere to the CC BY-NC-SA license. Furthermore, in contrast to the imbalanced language distribution in DALI, where English songs account for over 80% of the total songs, our dataset includes a greater proportion of songs in other languages, which is advantageous for multilingual lyrics transcription.

4. EXPERIMENTS

In this section, we first outline our experimental setup, including datasets and evaluation metrics. Next, we report lyrics transcription results on various benchmark datasets. We also conduct extensive ablation studies to verify the effectiveness of our methods. Finally, we demonstrate the reliability of our dataset through noise level estimation.

4.1 Experimental Setup

Datasets. Our proposed method does not require any training; thus, we directly test it on several accessible lyric transcription benchmark datasets, including Jamendo [46], Hansen [15], MUSDB18 [17], DSing [8]. Among these, Jamendo, Hansen, and DSing are widely used test datasets in music transcription. MUSDB18, originally a dataset for music source separation, contains 150 rock-pop songs. The authors in [17] provided line-level lyric annotations for MUSDB18, making it a challenging real-world dataset for lyric transcription. Additionally, we manually collected 40 multilingual songs with lyrics annotations from MTG-Jamendo as a test set for the proposed dataset, which can be used to validate the reliability of our proposed dataset via transcription accuracy.

Method	a)	b)	c)
CTDNN-SA-mixture [17]	76.06	78.44	89.24
Ours-mixture	**50.90**	**47.04**	**50.70**
CTDNN-SA-vocals [17]	37.83	30.85	58.45
Ours-vocals	**26.29**	**25.27**	**33.30**

Table 4. The WERs (%) of our method and baseline [17] on three subsets of annotated MUSDB18. The results of our method are obtained from utterance-level segments.

Evaluation. We report the Word Error Rate (WER) as the evaluation metric, which is the ratio of the total number of insertions, substitutions, and deletions with respect to the total number of words. We calculate the average WER on the test sets. Since Whisper possesses the capability for long-form transcription, we directly evaluate entire songs using Jamendo, Hansen, and the multilingual test set. We perform utterance-level evaluations on MUSDB18 and DSing since they only have utterance-level annotations. We discovered that many songs in these evaluation datasets are problematic, such as incorrect lyric annotations and excessively short song segments. One notable problem is that sometimes there are prominent harmony parts in the background of a song. However, it is not provided in the lyric annotations (e.g., Adele's "Rolling in the Deep"). LyricWhiz is powerful enough to transcript both the leading vocal and the background vocal with high accuracy. Therefore, we removed these problematic data from our evaluations. Finally, we normalize the transcription results to match the standardized ground truths. We remove all special Unicode characters, such as emojis. All text is converted to lowercase, and numeric characters are converted to their alphabetic correspondence.

Budget. To ensure fast and multi-round inference of the Whisper-large model on various datasets, including the large-scale MTG-Jamendo dataset, we conducted our experiments concurrently on a server with 8xA100 80G GPUs. It takes approximately 9 hours to complete one round of inference, and each process uses up to 12G VRAM. The vocal probability threshold is set to 0.07 for PANNs-based vocal event detection. To carry out contextualized post-processing using ChatGPT, we invested a total of US$2,000 on GPT-4 API for the entire project.

4.2 Comparative Experiments

In order to verify the superiority of our approach, we compare it with several previous studies on benchmark datasets. W2V2-ALT [22], a transfer learning method based on ASR self-supervised models, represents the current state-of-the-art in lyric transcription tasks. In our experiments, we primarily compare our method with W2V2-ALT, as well as other previous methods. The experimental results, as shown in Table 3, indicate that our method achieves the best performance on Jamendo and Hansen and the second-best performance on DSing. In long-form

transcription datasets such as Jamendo and Hansen, our method significantly outperforms all previous approaches due to the strong contextual memory capabilities of both Whisper and ChatGPT. Furthermore, our method also leads by a considerable margin on MUSDB18, shown in Table 4, demonstrating the robust performance and resilience of our proposed method in more diverse and complex musical scenarios. It is worth noting that our method did not surpass previous results on the DSing dataset, which we attribute to two factors. First, previous models were trained on the DSing training set, making the DSing test set an in-distribution dataset for the models, while our approach does not require any training and directly employs large-scale ASR models for zero-shot lyric transcription. Second, the segmented evaluation on DSing results in the loss of contextual information, which consequently leads to inaccurate transcriptions.

4.3 Ablation Studies

To further substantiate the efficacy of each component within our proposed approach, we conducted comprehensive ablation experiments.

Whisper Prompt. In our experiments, we investigate the Whisper prompt mechanism and test various prompts. First, we construct a complex prompt following the format of ChatGPT prompts, including task descriptions, format specifications, and specific requirements. We then gradually reduce the constituent elements of the prompt and observe the results. We discover that, unlike general large language models, Whisper has weaker task understanding capabilities for complex prompts and can only comprehend shorter task prompts. In practice, using the simplest prompt "lyrics:" yielded the best results. For multilingual transcription, we translate "lyrics:" into the corresponding language. As shown in Table 3, the designed prompt performs better in long-form transcription scenarios, assisting the model in producing meaningful lyrics for difficult tasks. However, its performance is less effective at the utterance level, possibly because predicting a single line of lyrics does not require additional contextual information.

ChatGPT Ensemble. In order to confirm that ChatGPT can analyze and infer the most accurate version of lyrics, we first conduct a simple experiment. In this experiment, we add the ground truth lyrics to the predicted results and input them together into ChatGPT for ensembling. We then calculate the proportion of times ChatGPT ultimately chose the ground truth. If ChatGPT is able to choose the most accurate lyrics, *i.e.*, the ground truth, the final proportion should be close to 100%. The computed results on the Hansen dataset is 72.7% for ground truth data, which is sufficient to demonstrate that ChatGPT can make correct choices based on the constructed prompt and input lyrics. As further observed in Table 3, ChatGPT ensembling is particularly effective for long-form lyric transcription, suggesting that ChatGPT requires contextual information (the content of preceding and following lyrics, as well as the content of different versions of predicted lyrics) for inference. In contrast, utterance-level lyric inputs lack both context and diversity among different prediction results, leading to inferior performance.

Language	Songs_{train}	Songs_{test}	WER_{test}
English	3,791	20	21.86
French	1,030	7	26.64
Spanish	620	5	22.54
Italian	311	3	44.01
Russian	147	4	39.18
German	132	1	25.43
Overall	6,031	40	26.26

Table 5. The distribution of our dataset and WERs (%) on test set. We manually constructed a test set of 40 songs following the language distribution of the collected training set. Then, we applied our proposed method to the test set and computed the WER.

4.4 Dataset Analysis

In order to demonstrate the reliability of the dataset constructed using Whisper and ChatGPT on MTG-Jamendo, we manually create a multilingual test set for noise level estimation. Specifically, we first select six languages from the intersection of the languages in MTG and those in which Whisper performs best. We then conduct a stratified sampling of 40 songs on Jamendo and manually annotate their lyrics. We use these 40 songs as a test set, assessing the WER to estimate the noise level of our collected dataset. Table 5 presents the number of songs in each language and the WER results for the test set, where our method achieves decent WER levels for the majority of languages. As our goal is to construct a large-scale, multilingual dataset for weak supervision, our method's transcription results are acceptable. Furthermore, we have not implemented specific normalization for multilingual transcription results, such as removing diacritical marks, which could be employed to enhance performance.

5. CONCLUSION

This paper presents LyricWhiz, a novel zero-shot automatic lyrics transcription system excelling in various datasets and music genres. Combining Whisper and GPT-4, our approach significantly reduces WER in English and efficiently transcribes multiple languages. LyricWhiz further generates the first publicly accessible, large-scale, multilingual lyrics dataset with a human-annotated subset for noise level estimation and evaluation. The successful integration of the large speech model and large language model in LyricWhiz offers a novel avenue for traditional Music Information Retrieval (MIR) tasks, as previous task-specific solutions are being eclipsed by general-purpose models. Notably, large language models have demonstrated their superior language understanding abilities across various tasks. Hence, we anticipate further applications of large language models to a broader spectrum of music-related domains, such as text-to-music generation, to enhance the performance of various models.

6. ACKNOWLEDGEMENTS

We gratefully acknowledge the dataset post-processing work described in Section 3.3 offered by Jiawen Huang. Jiahao Pan and Wei Xue were supported by the Theme-based Research Scheme (T45-205/21-N) and Early Career Scheme (ECS-HKUST22201322), Research Grants Council of Hong Kong. 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]. Yizhi Li is fully funded by an industrial PhD studentship (Grant number: 171362) from the University of Sheffield, UK.

7. REFERENCES

[1] A. Tsaptsinos, "Lyrics-based music genre classification using a hierarchical attention network," *arXiv preprint arXiv:1707.04678*, 2017.

[2] H. Fujihara, M. Goto, and J. Ogata, "Hyperlinking Lyrics: A method for creating hyperlinks between phrases in song lyrics." in *ISMIR*, 2008, pp. 281–286.

[3] T. Hosoya, M. Suzuki, A. Ito, S. Makino, L. A. Smith, D. Bainbridge, and I. H. Witten, "Lyrics recognition from a singing voice based on finite state automaton for music information retrieval." in *ISMIR*, 2005, pp. 532–535.

[4] P. Knees and M. Schedl, "A survey of music similarity and recommendation from music context data," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 10, no. 1, pp. 1–21, 2013.

[5] 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.

[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] 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 *19th International Society for Music Information Retrieval Conference*, ISMIR, Ed., September 2018.

[8] G. R. Dabike and J. Barker, "Automatic lyric transcription from karaoke vocal tracks: Resources and a baseline system." in *Interspeech*, 2019, pp. 579–583.

[9] P. Törnberg, "ChatGPT-4 outperforms experts and crowd workers in annotating political twitter messages with zero-shot learning," *arXiv preprint arXiv:2304.06588*, 2023.

[10] 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, 2020.

[11] C. Gupta, H. Li, and Y. Wang, "Automatic pronunciation evaluation of singing." in *Interspeech*, 2018, pp. 1507–1511.

[12] C. Gupta, E. Yılmaz, and H. Li, "Automatic lyrics alignment and transcription in polyphonic music: Does background music help?" in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 496–500.

[13] E. Demirel, S. Ahlbäck, and S. Dixon, "MSTRE-Net: Multistreaming acoustic modeling for automatic lyrics transcription," *arXiv preprint arXiv:2108.02625*, 2021.

[14] 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, 2020.

[15] J. K. Hansen and I. Fraunhofer, "Recognition of phonemes in a-cappella recordings using temporal patterns and mel frequency cepstral coefficients," in *9th Sound and Music Computing Conference (SMC)*, 2012, pp. 494–499.

[16] R. M. Bittner, K. Pasalo, J. J. Bosch, G. Meseguer-Brocal, and D. Rubinstein, "vocadito: A dataset of solo vocals with f_0, note, and lyric annotations," *arXiv preprint arXiv:2110.05580*, 2021.

[17] K. Schulze-Forster, C. S. 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.

[18] X. Gao, C. Gupta, and H. Li, "Genre-conditioned acoustic models for automatic lyrics transcription of polyphonic music," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 791–795.

[19] S. Basak, S. Agarwal, S. Ganapathy, and N. Takahashi, "End-to-end lyrics recognition with voice to singing style transfer," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 266–270.

[20] C. Zhang, J. Yu, L. Chang, X. Tan, J. Chen, T. Qin, and K. Zhang, "PDAugment: Data augmentation by pitch and duration adjustments for automatic lyrics transcription," *arXiv preprint arXiv:2109.07940*, 2021.

[21] A. M. Kruspe and I. Fraunhofer, "Training phoneme models for singing with" songified" speech data." in *ISMIR*, 2015, pp. 336–342.

[22] L. Ou, X. Gu, and Y. Wang, "Transfer learning of wav2vec 2.0 for automatic lyric transcription," in *ISMIR*, 2022.

[23] R. Tang, K. Kumar, G. Yang, A. Pandey, Y. Mao, V. Belyaev, M. Emmadi, C. Murray, F. Ture, and J. Lin, "SpeechNet: Weakly supervised, end-to-end speech recognition at industrial scale," *arXiv preprint arXiv:2211.11740*, 2022.

[24] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised pre-training for speech recognition," *arXiv preprint arXiv:1904.05862*, 2019.

[25] 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, pp. 12 449–12 460, 2020.

[26] 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.

[27] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao *et al.*, "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, 2022.

[28] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," *arXiv preprint arXiv:2212.04356*, 2022.

[29] C. Wang, S. Chen, Y. Wu, Z. Zhang, L. Zhou, S. Liu, Z. Chen, Y. Liu, H. Wang, J. Li *et al.*, "Neural codec language models are zero-shot text to speech synthesizers," *arXiv preprint arXiv:2301.02111*, 2023.

[30] 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.

[31] M. C. McCallum, F. Korzeniowski, S. Oramas, F. Gouyon, and A. F. Ehmann, "Supervised and unsupervised learning of audio representations for music understanding," *arXiv preprint arXiv:2210.03799*, 2022.

[32] 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.

[33] 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.

[34] 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.

[35] Y. Shen, K. Song, X. Tan, D. Li, W. Lu, and Y. Zhuang, "HuggingGPT: Solving ai tasks with chatgpt and its friends in huggingface," *arXiv preprint arXiv:2303.17580*, 2023.

[36] T. Schick, J. Dwivedi-Yu, R. Dessì, R. Raileanu, M. Lomeli, L. Zettlemoyer, N. Cancedda, and T. Scialom, "Toolformer: Language models can teach themselves to use tools," *arXiv preprint arXiv:2302.04761*, 2023.

[37] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing," *ACM Computing Surveys*, vol. 55, no. 9, pp. 1–35, 2023.

[38] J. White, S. Hays, Q. Fu, J. Spencer-Smith, and D. C. Schmidt, "ChatGPT prompt patterns for improving code quality, refactoring, requirements elicitation, and software design," *arXiv preprint arXiv:2303.07839*, 2023.

[39] J. White, Q. Fu, S. Hays, M. Sandborn, C. Olea, H. Gilbert, A. Elnashar, J. Spencer-Smith, and D. C. Schmidt, "A prompt pattern catalog to enhance prompt engineering with ChatGPT," *arXiv preprint arXiv:2302.11382*, 2023.

[40] T. Shin, Y. Razeghi, R. L. Logan IV, E. Wallace, and S. Singh, "AutoPrompt: Eliciting knowledge from language models with automatically generated prompts," *arXiv preprint arXiv:2010.15980*, 2020.

[41] K. Shum, S. Diao, and T. Zhang, "Automatic prompt augmentation and selection with chain-of-thought from labeled data," *arXiv preprint arXiv:2302.12822*, 2023.

[42] J. Wei, X. Wang, D. Schuurmans, M. Bosma, E. Chi, Q. Le, and D. Zhou, "Chain of thought prompting elicits reasoning in large language models," *arXiv preprint arXiv:2201.11903*, 2022.

[43] V. I. Levenshtein *et al.*, "Binary codes capable of correcting deletions, insertions, and reversals," in *Soviet physics doklady*, vol. 10, no. 8. Soviet Union, 1966, pp. 707–710.

[44] E. Demirel, S. Ahlbäck, and S. Dixon, "Automatic lyrics transcription using dilated convolutional neural networks with self-attention," in *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2020, pp. 1–8.

[45] E. Demirel, S. Ahlbäck, and S. Dixon, "Low resource audio-to-lyrics alignment from polyphonic music recordings," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 586–590.

[46] D. Stoller, S. Durand, and S. Ewert, "End-to-end lyrics alignment for polyphonic music using an audio-to-character recognition model," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 181–185.

SOUNDS OUT OF PLÄCE? SCORE-INDEPENDENT DETECTION OF CONSPICUOUS MISTAKES IN PIANO PERFORMANCES

Alia Morsi[1] **Kana Tatsumi**[2] **Akira Maezawa**[3] **Takuya Fujishima**[3] **Xavier Serra**[1]

[1] Music Technology Group, Universitat Pompeu Fabra, Barcelona, Spain
[2] Nagoya Institute of Technology, Nagoya, Japan
[3] Yamaha Corporation, Hamamatsu, Japan

ABSTRACT

In piano performance, some mistakes stand out to listeners, whereas others may go unnoticed. Former research concluded that the salience of mistakes depended on factors including their contextual appropriateness and a listener's degree of familiarity to what is being performed. A *conspicuous* error is considered to be an area where there is something *obviously* wrong with the performance, which a listener can detect regardless of their degree of knowledge of what is being performed. Analogously, this paper attempts to build a score-independent conspicuous error detector for standard piano repertoire of beginner to intermediate students. We gather three qualitatively different piano playing MIDI data: (1) 103 sight-reading sessions for beginning and intermediate adult pianists with formal music training, (2) 245 performances by presumably late-beginner to early-advanced pianists on a digital piano, and (3) 50 etude performances by an advanced pianist. The data was annotated at the regions considered to contain conspicuous mistakes. Then, we use a Temporal Convolutional Network to detect the sites of such mistakes from the piano roll. We investigate the use of two pre-training methods to overcome data scarcity: (1) synthetic data with procedurally-generated mistakes, and (2) training a part of the model as a piano roll auto-encoder. Experimental evaluation shows that the TCN performs at an F-measure of 0.78 without pretraining for sight-reading data, but the proposed pretraining steps improve the F-measure on performance and etude data, approaching the agreement between human raters on conspicuous error labels. Importantly, we report on the lessons learned from this pilot study, and what should be addressed to continue this research direction.

1. INTRODUCTION

A commonly held notion in automatic music performance analysis (MPA) research is that deviations of music performances from their underlying music score can be regarded as performance mistakes. But previous music pedagogy research suggests that some of such deviations are more apparent to a listener than others [1, 2]. For example, a chord that is voiced differently from that written in the score might be overlooked, but missing a note in a characteristic motif or playing a note that clashes with the underlying harmony would stand out. Repp [1] referred to errors of the former category as *perceptually inconspicuous*. Accordingly, we consider a **conspicuous error** to be "a performance error that can be detected by the majority of listeners with a formal music training, regardless of their degree of knowledge about the underlying music score of a performed piece."

This paper explores the potential of building score-independent models that detect regions of *conspicuous errors* in MIDI piano performances of piano solo pieces based on Western music theory, as shown conceptually in Figure 1. Based on the intuition that a listener is capable of detecting obvious mistakes in piano performances by listening to the surrounding context, we use a non-causal variant of the Temporal Convolutional Network (TCN) [3] We gather datasets for our task, since despite the plethora of work in automatic MPA that has spanned both the score-dependent (or reference-dependent) [4–7] and score-independent paradigms [8–13], there is no data available to support our desired goal.

More specifically we: (1) gather three datasets of conspicuous errors in various performance situations, reporting on the dataset creation process and annotation procedure, (2) study the properties of the annotated data through (i) observing the annotated data for sources of inconsistencies, (ii) analyzing the relationship between inconspicuous and conspicuous errors and (ii) analyzing the ambiguity of the task through listening experiments, (3) present a model based on TCN to identify conspicuous errors from piano MIDI performance and discuss its effectiveness through experimental evaluation, and (4) present and evaluate two pre-training strategies, depending on the nature of the unlabeled data that can be acquired. A subset of the gathered data and listening examples can be found on the companion page [1]

2. RELATED WORK

We distinguish between locally and globally-based automatic MPA. In local approaches (such as the majority

© A. Morsi, K. Tatsumi, A. Maezawa, T. Fujishima, and X. Serra. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** 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 *Proc. of the 24rd Int. Society for Music Information Retrieval Conf.,* Milano, Italy, 2023.

[1] https://bit.ly/3UCCiea

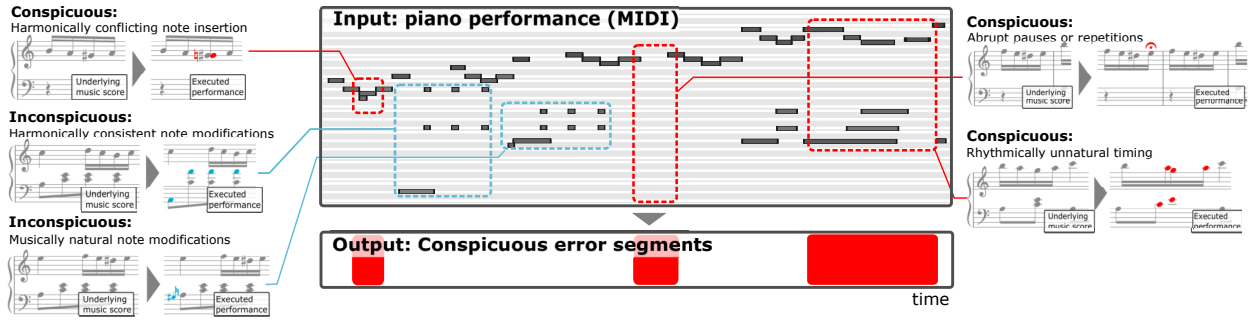


Figure 1: Illustration of our problem definition. Some errors stand out more than others in performance. Our goal is to identify segments containing conspicuous errors to the listeners, without the need for music score data.

of score-dependent performance assessment), the analysis is conducted at a note (or equivalent) level. Global approaches learn from data mapping large performance snippets (often entire performances) to overall evaluations.

Local approaches include score-based performance mistake identification, which tends to cover note-level (or equivalent) errors such as pitch [1, 2, 4, 7] and rhythm mistakes [2]. Pitch mistakes are essentially categorized as ***pitch intrusions*** (extra note) and ***pitch omissions*** (missing note), and occasionally ***pitch substitutions*** (wrong note in-place of a correct one), although the latter can be treated the joint occurrence of the former two [1]. Alignment/score comparison-based approaches for detecting deviations are locally-based by definition. Piano assessment examples of such include [4, 7, 14], which cover pitch mistakes. Not all local approaches are score-dependent, such as those which capture note-level aspects relating to the articulation or sound quality. Examples are [15] and [12], for piano (3-point scale for quality of legato or staccato) and trumpet (7-point scale) respectively.

Global approaches to performance assessment have usually been score-free, with the exception of [5] which utilizes the score as input. Usually, such approaches are based on regression models mapping features to performance-wide ratings [9, 11, 16, 17], or end-to-end approaches which learn correspondences between whole or parts of performances to performance wide ratings [5, 10, 13]. Such ratings can be discrete or continuous and can span several performance dimensions. Although the connection has not been explicitly made, we speculate that most likely they would excel in capturing conspicuous performance mistakes that manifest as consistent errors/error patterns across a performance.

Accordingly, we frame our approach as a score-independent locally based one since our goal is to return binary labels for each time point in a piano MIDI roll reflecting the presence or absence of an obvious performance mistake. Therefore, we need similarly annotated data for piano MIDI performances to train our models. Despite score deviations not necessarily indicating conspicuous errors, our desired output is closest to that of score-based performance mistake identification systems because their output can be interpreted as a binary sequence indicating the presence or absence of a score deviation albeit without perceptual relevance. However, their methods are not applicable for our problem formulation.

3. DATA

We obtain 3 sources of non-commercial, piano MIDI performance data for different playing situations:

Sight-Reading Data (SR): 103 sight-reading performances comprising mostly of piano reductions of popular classical pieces, arranged for beginner to intermediate difficulty. They are played by seven beginning to intermediate adult pianists with formal music training.

Performance Data (PF): 245 performances of approximately 3 minutes each, collected from a digital piano recording app. Not all performed pieces are known, but most of them are pop and classical, that are either read from a score, or semi-improvised. While user attributes are unknown, the performance data suggests that the skill levels range between late-beginner and early-advanced.

Burgmüller Data (BM): 50 performances from Burgmüller's 25 Etudes, Op. 100 recorded twice on a digital piano. They are played by an advanced pianist who had previously played the etudes. The pianist practiced each etude briefly before recording two takes.

The total time for the **SR**, **PF**, and **BM** are 379, 723, and 60 minutes respectively, of which 128, 176, and 3 minutes were annotated as conspicuous errors. Non-overlapping splits of **SR** and **PF** are used for training, validation, and testing, whereas **BM** is kept exclusively for testing. The annotation procedure is described in 3.1. **SR** and **PF** subsets cannot be shared, but short excerpts of them, and the full **BM** set can be found in the companion page.

3.1 Annotation Procedure

We had 2 annotators: ***Annotator 1***, who has experience as a classical piano teacher, and ***Annotator 2***, has training in music production and is also an intermediate-level pianist. We asked Annotator 1 to label the **SR** and **BM** data, and asked Annotator 2 to label the **PF** data, and to indicate (yes/no) whether they know the piece being performed. For the **SR** and **PF** subsets, annotators were given instructions to annotate *obvious* performance mistakes that can be recognized even without checking the score, and it was left open to them to decide what that entails. The an-

notation was done with Cubase [2], and they were asked to add an annotation at MIDI note 0 covering the span of the time window which they judge as pertaining to an error. Despite the potential label ambiguity due to the openness of the instructions, we wanted to observe the judgments of different people in this pilot study so that we can improve the data annotation protocol for future experiments.

The **BM** subset was treated differently because it has been played off of known music score data. First, the performances were automatically annotated with sites of score deviations using a score alignment system. Then, the annotator manually reviewed the labels by listening to the performance while looking at the corresponding sheet music, and added missing deviations from the score or removed those which do not reflect errors. The annotator simultaneously manually labeled each error as conspicuous or not.

3.2 Annotation Examples and Pitfalls

Some types of errors were labeled more consistently than others. The more common error modes, as shown in Figure 2, include insertions and deletions of notes that do not fit in musical context, abrupt pauses, and unstable rhythm coming from hesitations during playing. Annotators have shown reasonable consistency in terms of label location and span when mistakes are relatively short after which the player recovers into their playing flow, such as those of Figure 2. However, more compound deviations were labelled ambiguously. For example, sometimes after an error a player would 'sneak-in' some practice before resuming the flow of the piece. In such examples, if the short phrase being practiced sounds out of context, but in itself is coherent, an open question is where the label should be, and whether it should be one continuous label or an intermittent one.

Moreover, we also observe the presence of non-annotated conspicuous mistakes in the data, but there is an inherent ambiguity in how one would assess a "bad but acceptable" and "erroneous" performance". In a discussion with Annotator 1 after the annotations, they indicated that their mental model for deciding whether a segment should be labelled was dependent on every performance. If a region contrasts with their expectation of the music given how that performer is playing, then it was annotated. This opens the possibility that annotators have calibrated what should count as a mistake based on individual performance. Silence regions are one of the main sources of ambiguity, since silences between correct portions are non-annotated regardless of their length, but silences within or surrounding mistake portions often receive a mistake label.

3.3 Analysis of the dataset

*3.3.1 Conspicuous to total label ratio in **BM***

Although the ratio of annotated regions to total performance time is very small in the **BM** data, its annotation approach of allows us to investigate the relationship between the set of errors obtained by comparing with a score

[2] https://www.steinberg.net/cubase/

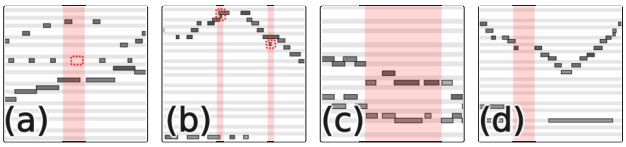


Figure 2: Examples of musical attributes that seemed to be consistently annotated as conspicuous errors (in red). (a) missed note that breaks a pattern, (b) harmonically unnatural note insertions, (c) repetition, (d) abrupt pauses.

(presumably all errors) to conspicuous errors. We found that 59% of all identified errors were perceived as conspicuous. Note that this is a very subset-specific result, because it depends on the ratio between subtle and obvious errors in the performances themselves as much as the qualities of the performer and the annotation.

3.3.2 Listening test of conspicuous errors

Through a listening test of some performance portions labeled as conspicuous errors and unlabeled areas for **PF**, we assess how different subjects agree with the annotations and among themselves. We chose **PF** because we expect it to contain a nice balance between famous and unknown pieces for each subject.

Conditions: We recruited 31 subjects, not necessarily trained musicians. 84% of the subjects had experience playing a musical instrument, and 97% of the subjects had experienced either reading or notating music scores. Each subject is asked to first listen with headphones to a snippet from the **PF** dataset, ranging from 4 to 12 seconds. The snippet is either (1) a randomly chosen conspicuous mistake segment, with 2 seconds of padding on either end, or (2) a segment that contains no error label, whose duration is the average duration of the conspicuous error segments within the piece, plus two seconds of padding. The subjects were allowed to skip questions and no constraints were given on the number of times the snippet may be listened to. The subject is then asked to choose if they hear an obvious mistake or not, along with the subject's knowledge of the piece. This procedure was repeated 15 times. Then, we scale the counts obtained when presenting non-conspicuous snippets, to provide a sensible assessment of the dataset itself. That is, the ratio of snippets containing the inconspicuous error to the conspicuous ones, ρ_0, should match the ratio between the total duration of the inconspicuous error labels to that of the conspicuous labels in the dataset, ρ_1. Thus, we scale the count of the responses obtained when presenting the inconspicuous error by ρ_1/ρ_0.

Results and discussion: A total of 462 responses were obtained (30-31 responses per snippet). The precision, recall, and the F-measure of how correctly the subjects identified the mistakes were 0.37, 0.50, and 0.43, respectively. The result suggests that the notion of conspicuous error is not so clear-cut when only presenting a short snippet surrounding an error, without providing a longer musical context. We also found that famous pieces tend to get more consistent responses. To check this, we computed for each

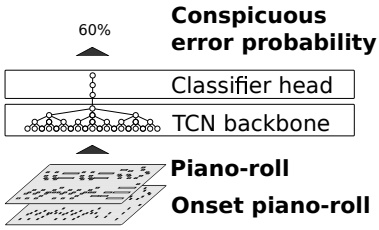


Figure 3: Our method reads a piano roll and outputs the probability of the center of a segment being a conspicuous error. It is comprised of a TCN backbone and a 1d convolution classifier head.

snippet (1) the probability that a song is unknown and (2) the entropy of the probability that a subject would identify that snippet to contain an error. The correlation between (1) and (2) was 0.63, indicating a moderate correlation between how well the piece is known among the subjects and how consistent are the labels.

4. METHODOLOGY

Given a sequence of piano note events, the goal is to infer a time sequence of binary labels that indicates the presence of conspicuous errors at a given time.

4.1 Model

Our model is a TCN-based network that receives a piano roll $\mathbf{X}$ as input and emits a binary label of conspicuous error e at each time frame of the piano roll. As shown in Figure 3, it is comprised of a feature extraction backbone followed by a classification head. We choose to assign a label at *frame-level* instead of note-level, since not only the note itself but its absence can indicate errors.

4.1.1 Piano Roll Input

Two piano rolls are extracted for a given sequence of piano note events, one for the note onset and another for the sustained portion according to the key depression. Specifically, suppose a set of I MIDI note events (start time, end time, pitch, velocity) given as $\{(s_i, e_i, p_i, v_i)\}_i^I$, and a sampling rate of R are given. Then, a 256-dimensional piano roll $X \in \mathbb{R}^{256\times T}$ is computed, such that $X(p_i, \text{round}(Rs_i)) = v_i$, and $X(128 + p_i, \text{round}(Rs)) = v_i$ for $s \in [s_i, e_i]$. Partitura [18] is used for the computation, and R is set to 16 Hz.

Notice that the sustain pedal information is ignored in the computation of the piano roll. This is necessary to prevent the piano roll of the sustained portion from smearing since a beginning pianist has a tendency to keep the pedal depressed which causes and excessive elongation of the computed note durations.

4.1.2 Conspicuous mistake detector

We model the mistake detector as a simple TCN comprising of a feature extraction backbone followed by a classification head, based on preliminary experiments exploring model architectures and inspired by the approach in [13].

Feature extraction backbone: Given the piano roll X, the feature extraction backbone computes a feature $\phi \in \mathbb{R}^{D\times T}$. We set $D = 256$ in this paper. This is realized as a 5-layer noncausal TCN with dilation of [1,2,4,8,16], and for all layers, has an output channel size of 256, kernel size of 3, uses ELU nonlinearity and has a residual connection, similar in spirit to [3].

Classification head: Given the feature ϕ, a network comprising of three layers of 1x1 convolution with output channel sizes [256,64,1] with residual connections and ELU nonlinearity followed by a sigmoid function is used to arrive at the conspicuous error posterior probability e.

4.2 Training strategies

The model is trained using RAdam with a learning rate of 10^{-3}, as to minimize the cross-entropy between the conspicuous error probability e and the posterior distribution computed from the ground-truth label. We augment the data by randomly transposing the entire MIDI file in the training data. Furthermore, when computing the cross-entropy loss, we smooth the ground-truth label to account for annotation inconsistencies in the start and end times of the conspicuous error segment. Furthermore, since it is difficult to obtain annotations of conspicuous errors, we pre-train the model as well, using the following two strategies.

4.2.1 Pretraining the feature extractor as an autoencoder

The feature extractor can be trained in an unsupervised manner, by training it as an autoencoder for a much larger collection of piano performances in the wild. Specifically, we train an auto-encoder using the feature extraction TCN introduced earlier as the encoder and a TCN with transposed 1d convolutions instead of a 1d convolution as the decoder. This way, the space of ϕ is pre-trained as to model the space of piano performances within a given receptive field of a TCN. This method could be useful if a large dataset of performances of unknown performance qualities are obtainable.

4.2.2 Pretraining the model with synthetic mistake labels

The model can also be pre-trained on performance data onto which mistakes are simulated and corresponding mistake labels are inserted to match the expected format of data in Section 3.1. Specifically, we apply systematic adjustments to a set of mistake-free performances and modify the note events, in a manner inspired by performance mistakes made by beginning adult pianists [19]. For each note event, with probability p_c we modify the note in one of the following ways:

1. With probability p_o omit a note with a probability p_o
2. With probability p_r replace a note, to the same note transposed n semitones, to simulate hitting the wrong key.
3. With probability p_i insert a note that is transposed by n semitones.

Method	Precision	Recall	F-measure
Baseline	**0.79**	**0.80**	**0.78**
SYNTH	0.65	0.76	0.69
SYNTH(FT)	0.61	0.69	0.62
AE	0.55	0.59	0.55
AE+SYNTH	0.44	0.65	0.51

(a) **SR** Data

Method	Precision	Recall	F-measure
Baseline	0.28	0.46	0.33
SYNTH	0.27	0.54	0.34
SYNTH(FT)	**0.30**	0.61	**0.38**
AE	0.28	0.52	0.34
AE+SYNTH	0.27	**0.63**	0.36

(b) **PF** Data

Method	Precision	Recall	F-measure
Baseline	0.26	0.36	0.26
SYNTH	0.26	**0.69**	0.35
SYNTH(FT)	0.26	0.49	0.32
AE	0.27	0.46	0.31
AE+SYNTH	**0.28**	0.52	**0.35**

(c) **BM** Data

Table 1: Results for different training strategies

4. With probability p_p pause the performance by a small amount distributed uniformly between 0.3 and 0.8 seconds. With probability p_{pr}, repeat the last played note.
5. With probability p_s pause the performance by a large amount distributed uniformly between 2 and 4 seconds. Repeat the last played note.

In this paper, we set $p_c = 5\%$, $p_o = 10\%$, $p_i = 39\%$, $p_r = 39\%$, $p_s = 2\%$, and $p_p = 10\%$. Furthermore, for note replacement and insertion, n is chosen so that $n = 1, 2$ are chosen with probabilities of 22% and $n = 4, 6$ by 2%. For a set of mistake-free performances, we obtained 260 hours of mostly jazz and classical MIDI piano performances. The quality and repertoire are comparable to those available from Yamaha PianoSoft [3] .

This method is useful if many performances that are known to be relatively error-free are obtainable Furthermore, this idea may possibly be used for data augmentation, at the risk of increasing false positives, since not all synthetic errors sound conspicuous, as also hinted by [1,2].

4.3 Experiment: Model Evaluation

We evaluate our model using different training strategies.

4.3.1 Experimental conditions

Our model has been trained with the following strategies:

1. Baseline - The model is trained on **SR** and **PF** data.
2. SYNTH - Same as Baseline, in addition to the inclusion of a subset of the synthetic data introduced in Section 4.2.2 during training and validation.
3. SYNTH(FT) - The model is pretrained on the synthetic data, then fine-tuned using **SR** and **PF**. This simulates a situation where a new annotated dataset becomes available after traing a model solely trained on a synthetic data.
4. AE - Train TCN autoencoder introduced in Section 4.2.1 as a pretraining step for the backbone TCN, using approximately 100,000 MIDI performances played by various users. The set of performances does not contain **SR PF** or **BM**, although it is obtained from the same source as **PF**. The model is fine-tuned on **SR** and **PF**.
5. AE+SYNTH - Use the pretrained autoencoder backbone and fine-tune using **SR**, **PF** and the synthetic data.

The trained models have been validated on **SR** and **PF**, and tested on a test split of **SR**, **PF**, and the entire **BM**.

As the metric, we have evaluated the transcription precision/recall/F1-measure using `mir_eval` [20], treating the estimated and the ground-truth annotations as note events occurring at a predefined pitch. When computing the transcription metrics, the note onset and offset tolerances have been set to 2 seconds. Furthermore, based on the validation set, the ends of the estimated segments have been padded by 0.2 seconds and overlapping segments have been merged.

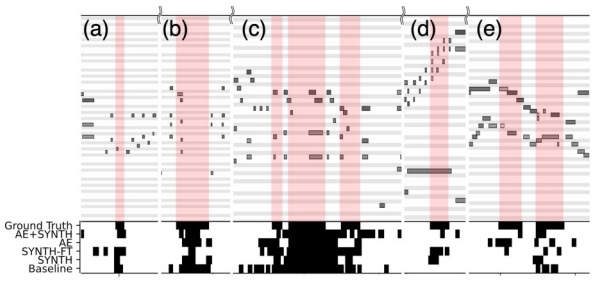
4.3.2 Results and discussion

The results are shown in Table 1. For **PF** and **BM** datasets, the augmentation strategies offer some improvements. The two strategies proposed, i.e., the use of synthetic data and autoencoder, also result in improvements. In general, both strategies tend to improve the recall rate, suggesting that they provide similar qualitative improvements, and either one can be used depending on the data available.

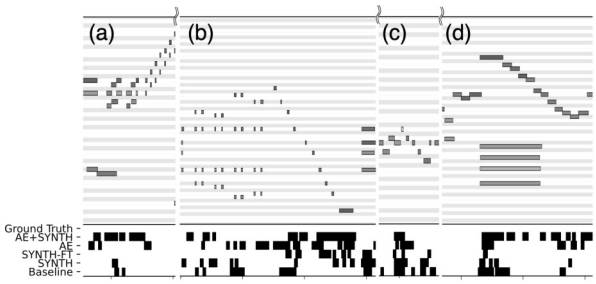
Despite the augmentation strategies, the F-measures for **PF** and **BM** data suggest future room for improvement, even taking into account the ambiguity of conspicuous errors. The **PF** and **BM** data are difficult to infer, as seen by the differences in the F-measure between the **SR** dataset and the two. As another example, the validation F-measure of the models on the synthetic dataset is about 0.60. This suggests that the model is moderately capable of pin-pointing the ground-truth labels if they are easy to classify, or generated stochastically but systematically. At the same time, however, the model has room for improvement, as the best-performing F-measure of 0.38 on the **PF** dataset falls somewhat short of the oracle F-measure of 0.43, as discussed in Section 3.3.

The method performs well for the **SR** data, perhaps because most of the mistakes are quite conspicuous in a sight-reading situation, especially compared to **PF** and **BM**, both of which contain mostly beginner-intermediate performances with occasional mistakes. The performance tends to drop as more pretraining steps are added, presumably because the pretraining data mostly contain data of the same type as the **PF** set, increasing the disparity between the training data and the test data. In sight-reading situations, the results suggest it is sufficient simply to train on a

[3] `https://shop.usa.yamaha.com/`



(a) True positives. The black band indicates the detected conspicuous error with different training strategies. The model presumably responds to (a) repetition, (b) silences, (c) slight hesitations in playing, (d) note insertions, and (e) lack of synchrony voices.



(b) False positives. The model presumably confuses (a) the repeated motives as an error, (b) rhythm with rest as abrupt pauses, (c) an audible but weak note with a note deletion, and (d) a long chord after a fast passage with hesitations.

Figure 4: Examples of typical operation and failure modes.

dataset that solely contains data from the same set, instead of pretraining or augmenting the dataset with typical amateur performances containing some conspicuous errors.

4.3.3 Qualitative insights of the estimates

Figure 4 shows some examples of true positives that are consistent across different strategies and consistent false positives. The proposed method tends to capture repetition, pauses, hesitations, and note insertions that occur in narrow pitch intervals as mistakes. At the same time, however, the very same properties arising from musical expression or composition are detected as false positives, such as repeated motifs, ornaments, and grand pauses. Even though such musical aspects are superficially performed similarly to the aforementioned mistakes, humans are capable of differentiating between genuine performance mistakes and those within musical contexts. This suggests that the model has room to improve by modeling the underlying composition better. The readers are invited to check the companion page for examples.

5. LIMITATIONS AND IMPROVEMENTS

Our work opens door to many open problems that need to be solved, some more fundamental than others.

Problem definition and annotation protocol: More work is needed to define the concept of conspicuous errors, and how the task should be evaluated from a music technology perspective. Accordingly, a more comprehensive protocol for data collection should be developed. Although we had kept the annotation instructions open to also develop an understanding of annotator behavior, it became evident that our data collection approach does not guarantee that the labels we have are for solely conspicuous errors. In [1], conspicuous errors were identified in a music performance by finding the subset of agreed-upon mistake labels between multiple listening subjects.

To define manifestations of conspicuous errors, a midpoint should be found between a rule-based approach and one learned from empirical labels. The outcome should be a set of error descriptions, some of which happen at particular time instants and some over longer windows, whether continuous windows or a longer span of intermittent labels. However, since the conspicuousness of errors is inspired by a perceptual idea, we think these errors should be defined through an empirical process albeit better defined than the one in this study to avoid the same pitfalls.

Synthetic mistakes: Synthetic data is important for improving performance, but current synthesized mistakes sound unnatural. A simple example was a case of induced pitch insertions, where it seemed impossible that someone can perform with such confidence and tempo despite the extent of out-of-context pitch insertions. We observe that beginners make mistakes and employ recovery strategies in a manner that is more complex than the presented method, so a better understanding of beginning pianists' behavior is necessary to create more natural-sounding mistakes.

Listener, player, expression, and style: Conspicuous errors are dependent on the listener's knowledge of the piece and the proficiency of the performer. Furthermore, conspicuous error and expression are two sides of the same coin. For example, hitting an adjacent key can either come across as an expressive ornament or a conspicuous error. This suggests that conspicuous error detection should inherently be conditioned on the style, the level of the listener, and the player's proficiency.

Connecting with pedagogy and edu-tainment: The impact of music education software which provides analysis solely founded on rigid note-level rhythmic and pitch correctness has been challenged [21] on the basis that users might end up too focused on playing too correctly (almost robotically) to attain the highest scores. There are many pedagogical considerations for designing useful automatic assessments [22].

6. CONCLUSION

This paper presented a study on detecting conspicuous performance mistakes for a piano solo performance of beginning to intermediate players. We (1) clarified the idea of a *conspicuous* error in line with previous research, (2) gathered locally annotated piano MIDI performance data, and (3) discussed sources of inconsistencies in our data through analysis of the annotation procedure and subjective tests. Although some of our models show an acceptable performance on the test split of the **SR** data subset, we find that the our pre-training suggestions do not provide remarkable performance improvements.

7. REFERENCES

[1] B. H. Repp, "The art of inaccuracy: Why pianists' errors are difficult to hear," *Music Perception: An Interdisciplinary Journal*, vol. 14, p. 161–183, 1996.

[2] B. Gingras, C. Palmer, P. N. Schubert, and S. McAdams, "Influence of melodic emphasis, texture, salience, and performer individuality on performance errors," *Psychology of Music*, vol. 44, p. 847–863, 2016.

[3] M. E. P. Davies and S. Böck, "Temporal convolutional networks for musical audio beat tracking," in *Proc. European Signal Processing Conference (EUSIPCO)*, September 2019.

[4] E. Nakamura, K. Yoshi, and H. Katayose, "Performance error detection and post-processing for fast and accurate symbolic music alignment," in *Proc. International Society for Music Information Retrieval Conference (ISMIR)*, October 2017, pp. 347–353.

[5] J. Huang, Y. N. Hung, K. A. Pati, S. Gururani, and A. Lerch, "Score-informed networks for music performance assessment," in *Proc. International Society for Music Information Retrieval Conference (ISMIR)*, 2020.

[6] H. Zhang and Y. a. Jiang, "Learn by referencing: Towards deep metric learning for singing assessment," in *Proc. International Society for Music Information Retrieval Conference (ISMIR)*, 2021.

[7] E. Benetos, A. Klapuri, and S. Dixon, "Score-informed transcription for automatic piano tutoring," in *Proc. European Signal Processing Conference (EUSIPCO)*, 2012, pp. 2153–2157.

[8] T. Nakano, M. Goto, and Y. Hiraga, "An automatic singing skill evaluation method for unknown melodies using pitch interval accuracy and vibrato features," in *Proc. International Conference on Spoken Language Processing (ICSLP)*, September 2006, p. 1706–1709.

[9] C. W. Wu, S. Gururani, C. Laguna, A. Pati, A. Vidwans, and A. Lerch, "Towards the objective assessment of music performances," in *Proc. International Conference on Music Perception and Cognition (ICMPC)*, July 2016.

[10] K. A. Pati, S. Gururani, and A. Lerch, "Assessment of student music performances using deep neural networks," *Journal of Applied Sciences*, vol. 8, no. 4, 2018. [Online]. Available: https://www.mdpi.com/2076-3417/8/4/507

[11] J. Abeßer, J. Hasselhorn, S. Grollmisch, C. Dittmar, and A. Lehmann, "Automatic competency assessment of rhythm performances of ninth-grade and tenth-grade pupils," in *Joint Proc. International Computer Music Conference (ICMC), and Sound and Music Computing Conference (SMC)*, September 2014, pp. 1252–1256.

[12] T. Knight, F. Uphamm, and I. Fujinaga, "The potential for automatic assessment of trumpet tone quality," in *Proc. International Society for Music Information Retrieval Conference (ISMIR)*, 2011.

[13] P. Seshadri and A. Lerch, "Improving music performance assessment with contrastive learning," in *Proc. International Society for Music Information Retrieval Conference (ISMIR)*, November 2021.

[14] T. Fukuda, Y. Ikemiya, K. Itoyama, and K. Yoshii, "A score-informed piano tutoring system with mistake detection and score simplification." Proc. Sound and Music Computing Conference (SMC), Jul 2015.

[15] V. Phanichraksaphong and W.-H. Tsai, "Automatic evaluation of piano performances for steam education," *Applied Sciences*, vol. 11, no. 24, 2021. [Online]. Available: https://www.mdpi.com/2076-3417/11/24/11783

[16] J. Abeßer, J. Hasselhorn, S. Grollmisch, C. Dittmar, and A. Lehmann, "Automatic quality assessment of vocal and instrumental performances of ninth-grade and tenth-grade pupils," in *Proc. International Symposium on Computer Music Multidisciplinary Research (CMMR)*, 2013.

[17] J. Pan, M. Li, Z. Song, X. Li, X. Liu, H. Yi, and M. Zhu, "An Audio Based Piano Performance Evaluation Method Using Deep Neural Network Based Acoustic Modeling," in *Proc. Interspeech 2017*, 2017, pp. 3088–3092.

[18] C. E. Cancino-Chacón, S. D. Peter, E. Karystinaios, F. Foscarin, M. Grachten, and G. Widmer, "Partitura: A Python Package for Symbolic Music Processing," in *Proc. Music Encoding Conference (MEC2022)*, 2022.

[19] Y. Morijiri, S. Obata, A. Maezawa, and T. Fujishima, "Understanding the challenges for adult beginners at piano practice from an analysis of errors," in *Proc. Asia-Pacific Symposium for Music Education Research (APSMER2021)*, 2021.

[20] 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. International Society for Music Information Retrieval Conference (ISMIR)*, 2014, pp. 367–372. [Online]. Available: http://dblp.uni-trier.de/db/conf/ismir/ismir2014.html#RaffelMHSNLE14

[21] A. Acquilino and G. Scavone, "Current state and future directions of technologies for music instrument pedagogy," *Frontiers in Psychology*, vol. 13:835609, 2022.

[22] V. Eremenko, A. Morsi, J. Narang, and X. Serra, "Performance assessment technologies for the support of musical instrument learning," in *Proc. International Conference on Computer Supported Education (CSEDU)*, May 2020, pp. 629–640.

VAMPNET: MUSIC GENERATION VIA MASKED ACOUSTIC TOKEN MODELING

Hugo Flores García[1,2] **Prem Seetharaman**[1] **Rithesh Kumar**[1] **Bryan Pardo**[2]

[1] Descript Inc.
[2] Northwestern University
hugofg@u.northwestern.edu

ABSTRACT

We introduce VampNet, a masked acoustic token modeling approach to music synthesis, compression, inpainting, and variation. We use a variable masking schedule during training which allows us to sample coherent music from the model by applying a variety of masking approaches (called prompts) during inference. VampNet is non-autoregressive, leveraging a bidirectional transformer architecture that attends to all tokens in a forward pass. With just 36 sampling passes, VampNet can generate coherent high-fidelity musical waveforms. We show that by prompting VampNet in various ways, we can apply it to tasks like music compression, inpainting, outpainting, continuation, and looping with variation (vamping). Appropriately prompted, VampNet is capable of maintaining style, genre, instrumentation, and other high-level aspects of the music. This flexible prompting capability makes VampNet a powerful music co-creation tool. Code [3] and audio samples [4] are available online.

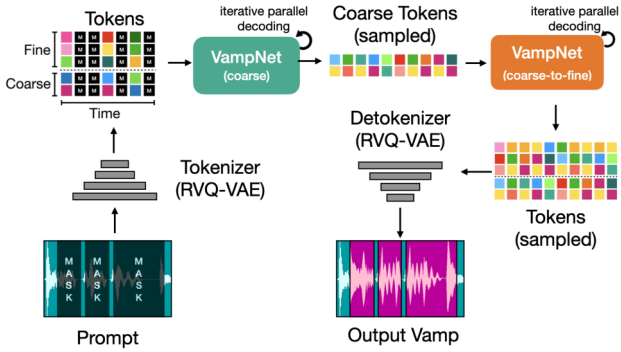


Figure 1. VampNet overview. We first convert audio into a sequence of discrete tokens using an audio tokenizer. Tokens are masked, and then passed to a masked generative model, which predicts values for masked tokens via an efficient iterative parallel decoding sampling procedure at two levels. We then decode the result back to audio.

1. INTRODUCTION

In recent years, advances in discrete acoustic token modeling have resulted in significant leaps in autoregressive generation of speech [1, 2] and music [3]. Meanwhile, approaches that use non-autoregressive parallel iterative decoding have been developed for efficient image synthesis [4, 5]. Parallel iterative decoding promises to allow faster inference than autoregressive methods and is more suited to tasks like infill, which require conditioning on both past and future sequence elements.

In this work, we combine parallel iterative decoding with acoustic token modeling, and apply them to music audio synthesis. To the best of our knowledge, ours is the first [1] extension of parallel iterative decoding to neural audio music generation. Our model, called VampNet, can be flexibly applied to a variety of applications via token-based prompting. We show that we can guide VampNet's generation with selectively masked music token sequences, asking it to fill in the blanks. The outputs of this procedure can range from a high-quality audio compression technique to variations on the original input music that match the original input music in terms of style, genre, beat and instrumentation, while varying specifics of timbre and rhythm.

Unlike auto-regressive music models [2, 3], which can only perform music continuations – using some prefix audio as a prompt, and having the model generate music that could plausibly come after it – our approach allows the prompts to be placed anywhere. We explore a variety of prompt designs, including periodic, compression, and musically informed ones (e.g. masking on the beat). We find that our model responds well to prompts to make loops and variations, thus the name VampNet [2]. We make our code open source [3] and highly encourage readers to listen to our audio samples [4].

[1] While our work was under peer review, Google released SoundStorm [6], which leverages a similar parallel iterative decoding approach to ours.

[2] To vamp is to repeat a short passage of music with variation.
[3] https://github.com/hugofloresgarcia/vampnet
[4] audio samples: https://tinyurl.com/bdfj7rdx

© H. Flores García, P. Seetharaman, R. Kumar, and B. Pardo. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** 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.

2. BACKGROUND

Two-stage approaches to generative modeling have gained traction in image [4, 5, 7, 8] and audio [2, 3, 6, 9] synthesis, largely in part due to their computational efficiency. In the first stage, a discrete vocabulary of "tokens" is learned for the domain of interest. The input is put through an encoder to obtain these tokens, which can be converted back into the input domain via a corresponding decoder. In the second stage, a model is trained to generate tokens, and is optionally given some conditioning (e.g. previous tokens, a text description, a class label) to guide generation.

2.1 Stage 1: Tokenization

In images, visual tokenization has been leveraged for state-of-the-art classification [10] and synthesis [4, 7, 8, 11]. The most popular approach is to use vector quantization on a latent space. Similar approaches have been explored for audio [12], but until recently such approaches have been restricted to low sampling rates (e.g. 16khz), or have been restricted to speech audio. The "sampling rate" of the latent space (the number of latent vectors required every second to represent audio) is a critical aspect of the tokenization scheme. The lower the sampling rate of the latent space, the easier the next stage (generation) will be to accomplish. Recently, methods based on residual vector quantization [13, 14] have been proposed for audio tokenization at high compression rates with good reconstruction quality of high-sample-rate audio.

The primary work we leverage for audio tokenization is the Descript Audio Codec (DAC) [15]. With DAC, audio is encoded into a sequence of tokens via a fully convolutional encoder. The output of this encoder is then quantized using a hierarchical sequence of vector-quantizers [11]. Each quantizer operates on the residual error of the quantizer before it. Because of this residual vector quantization, DAC is able to reconstruct audio with very high quality, at a high compression ratio. It, along with its predecessors [13, 14], are instrumental in enabling audio language models like AudioLM [2], MusicLM [3], and VALL-E [1]. While we later briefly describe our tokenizer, the key contributions of our work are applicable to the output of any audio tokenizer and our specific audio tokenizer is not the focus of this work.

2.2 Stage 2: Generation

Given audio encoded as tokens, the common approach is to use an autoregressive model [16] for generation. State-of-the-art (SOTA) audio generation approaches like AudioLM [2], MusicLM [3], and JukeBox [17] use this approach, generating each acoustic token in the sequence in a step-by-step fashion using transformer-based [18] decoder-only models. Autoregressive sampling is slow in nature due to the high number of steps required at inference time [4]. Further, autoregressive models inherently restrict downstream applications, as each generated token is only conditioned on the previous tokens. For an autoregressive model to perform tasks like inpainting ("filling in the middle"), one must re-arrange the data during training [19].

In language, masked modeling has been used extensively as a pre-training procedure for high-quality semantic representations [20]. This procedure has also been extended for representation learning in images [21] and audio [22]. Masked modeling for representation learning generally has a constant mask probability. For example, in BERT [20], tokens are masked 15% of the time during training. It has been shown that this approach is equivalent to a single-step discrete diffusion model [23], that uses masking for its noising procedure. Therefore, we can extend masked modeling to masked generative modeling by varying the probability of masking a token during training. This was done for image generation in MaskGIT [4], and in language [23]. Similar to diffusion modeling [24, 25], which seeks to synthesize data starting from random noise through a series of denoising steps, masked generative modeling seeks to synthesize data starting from completely masked data through a series of "unmasking" steps.

Key to the efficiency of MaskGIT and related approaches is a *parallel iterative decoding procedure*. In parallel iterative decoding, the model predicts every token in the output sequence in a single forward pass. However, after just one forward pass of the model, the output often does not have high quality. The output of the first sampling step is re-masked, with a lower masking probability, and then put through the model again. In this way, masked generative models can efficiently refine their output, resulting in high quality generation.

In unconditional generation tasks, the model is asked to generate a realistic sample from the target data distribution from scratch, without any guidance. This is a difficult problem, as many target data distributions are highly multimodal. Unconditional generative models are susceptible to mode collapse [26], blurry samples, mode averaging, and other issues [27]. Therefore, some conditioning is helpful as it provides some signal for the model to resolve the multimodality. Conditioning is also a commonly used method to guide the output of the system towards desired content.

Conditioning can take the form of a class label, a genre tag or lyrics [17], or an associated text description [3,8,28]. Conditioning can also be applied at every timestep, like the semantic tokens of AudioLM [2], or aligned text or phonemes for text-to-speech generation [1].

In this work,we adopt a masked generative modeling approach with a parallel iterative decoding procedure, inspired by work in vision such as *MaskGIT* [4] and *Paella* [5], as illustrated in Figure 1. We do not apply any conditioning beyond that provided by the unmasked tokens in our encoded audio. As we show later, different approaches to masking, applied at inference time, can be used to steer generation in useful and artistic ways.

In training, tokens are masked randomly throughout the sequence. The model is then asked to predict the value of each of the masked tokens in a single forward pass, but it is conditioned on all of the unmasked tokens, both in the future as well as in the past. We vary the number of tokens

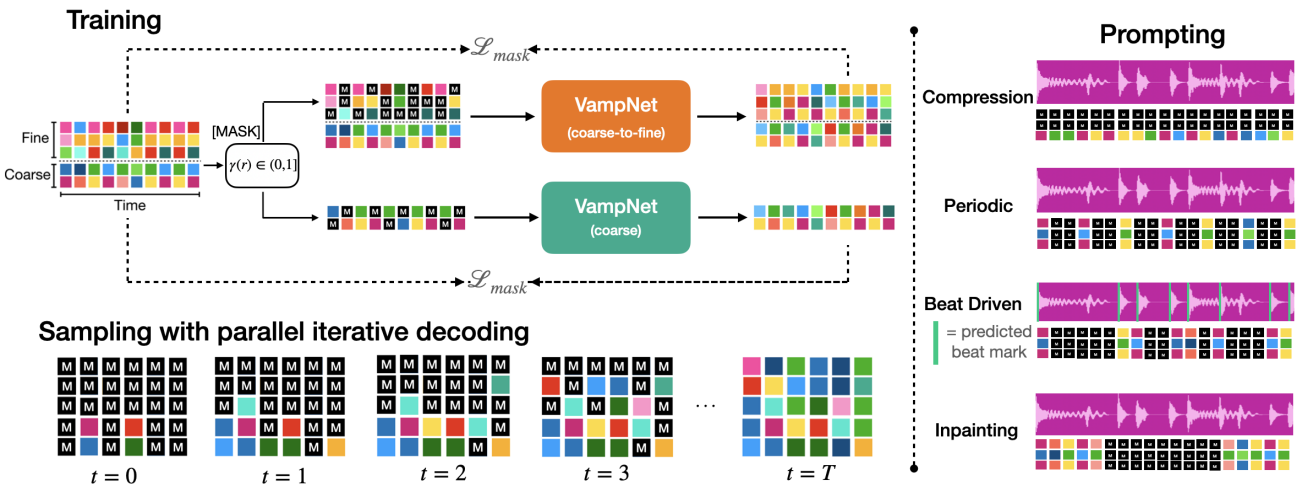


Figure 2. Training, sampling, and prompting VampNet. **Training**: we train VampNet using Masked Acoustic Token Modeling, where we randomly mask a portion of a set of input acoustic tokens and learn to predict the masked set of tokens, using a variable masking schedule. Coarse model training masks coarse tokens. Coarse-to-fine training only masks fine tokens. **Sampling**: we sample new sequences of acoustic tokens from VampNet using parallel iterative decoding, where we sample a subset of the most confident predicted tokens each iteration. **Prompting**: VampNet can be prompted in a number of ways to generate music. For example, it can be prompted periodically, where every Pth timestep in an input sequence is unmasked, or in a beat-driven fashion, where the timesteps around beat markings in a song are unmasked.

that are masked during training, allowing us to generate audio at inference time through a sampling procedure. We now describe our method in more detail.

3. METHOD

We adapt the procedure of *Masked Visual Token Modeling*, proposed in MaskGIT [4] to audio, accounting for several key differences between the vision and audio domain. We call our approach *Masked Acoustic Token Modeling*.

3.1 Masked Acoustic Token Modeling

We first train an audio tokenizer based on the techniques described in DAC [15]. Unlike the visual tokens of MaskGIT, our acoustic tokens are hierarchical in nature due to residual vector quantization. As a first step, the audio signal x is encoded at each time step t as a a D dimensional latent vector Z. We then quantize Z using N vector quantizers. Quantizer 1 produces $\hat{Z}_1$, a quantized approximation of Z that has residual error $R_1 = Z - \hat{Z}_1$. Thereafter, the residual from each quantizer i is passed to the next quantizer $i+1$, which produces a quantized approximation of the remaining residual error: $R_i \approx \hat{Z_{i+1}}$. Vector Z is reconstructed by summing the output of the N quantizers: $Z = \sum_{i=1}^{N} \hat{Z}_i$.

Since the encoded signal is represented as a quantized vector of N discrete tokens at each timestep, we have N tokens that can be masked or unmasked at each timestep. Rather than attempt to generate all tokens at once, we instead split the N tokens into N_c "coarse" tokens, and N_f "fine" tokens, as in AudioLM. We then train two generative models: one that generates the fine tokens given the coarse tokens as conditioning, and one that generates the coarse tokens given a sequence of coarse tokens. To generate a sample (Figure 1), we chain the two models together. First, we apply the coarse model to generate a sequence of coarse tokens. Then, we apply the coarse-to-fine model to generate the fine tokens. We decode the tokens to a 44.1khz waveform using the decoder of our audio tokenizer.

3.2 Training procedure

Let $\mathbf{Y} \in \mathbb{R}^{T \times N}$ be a matrix representing the output of the encoder for some audio segment. Each element $y_{t,n}$ in $\mathbf{Y}$ is a token from the nth level codebook at timestep t. Let $\mathbf{Y}_M$ be the set of all masked tokens in $\mathbf{Y}$ and $\mathbf{Y}_U$ be the set of all unmasked tokens in $\mathbf{Y}$. The model generates a probability distribution over the set of possible codebook values for each token $y \in \mathbf{Y}_M$, given the unmasked tokens and the model parameters θ. The training objective is to maximize the probability of the true tokens. This corresponds to minimizing the negative log likelihood.

$$\mathcal{L} = -\sum_{\forall y \in \mathbf{Y}_M} \log p(y|\mathbf{Y}_U, \theta) \tag{1}$$

To predict the masked tokens, we use a multi-layer bidirectional transformer, which predicts the probabilities of each possible token at every timestep, for every quantizer. If each quantizer has a codebook size of C possible values, and there are N quantizers, then the last layer of the network will be a fully connected layer of shape (E, CN), where E is the dimensionality of the output of the last layer. We then reshape this output into (EN, C), and compute the cross-entropy loss between the ground-truth one-hot token and the predicted token. Because the transformer is bidirectional, it can attend to all tokens in the input sequence to optimize the loss for each token.

For the coarse-to-fine generative model, the input sequence always contains N_c coarse tokens, and the masking operation is restricted to the N_f fine tokens. The last layer of this network only predicts masked fine tokens. Otherwise, the training procedure for both models is identical.

3.3 Sampling

We follow the same iterative confidence-based sampling approach used in MaskGIT. More concretely, given Y_M as the set of masked tokens and Y_U as the set of unmasked tokens, do:

1. **Estimate.** For each masked token y in Y_M, estimate the conditional probability distribution over its vocabulary of codebook values V.
2. **Sample.** For each masked token, sample from the distribution to generate an associated token estimate $\hat{y} \in V$. We don't use any sampling tricks in this step, sampling from the categorical probability distribution for each token as-is.
3. **Rank by Confidence.** Compute a confidence measure for each of the sampled tokens by taking their prediction log-probabilities and adding temperature-annealed Gumbel noise to them:

$$confidence(\hat{y}_t) = log(p(\hat{y}_t)) + temp \cdot g_t \quad (2)$$

   where $\hat{y}_t$ is a token estimate at timestep t, g_t is an i.i.d sample drawn from Gumbel(0,1) [29], and $temp$ is a hyperparameter that is linearly annealed to 0 over the number of sampling iterations. Then, sort the set of sampled token estimates by the confidence computed above. We find that high temperature values (e.g. > 6.0) result in higher quality samples.
4. **Select.** Pick the number of tokens to mask at the next sampling iteration, k, according to the masking schedule [5]. Take the k lowest confidence estimates and toss them out, re-masking their tokens. Place the remaining high-confidence token estimates in Y_U, removing their tokens from Y_M.
5. **Repeat** Return to step 1 until the number of iterations has been reached.

3.4 Prompting

Interactive music editing can be enabled by incorporating human guidance in the sampling procedure through the conditioning prompt of unmasked tokens. Because our approach isn't conditioned on any signal other than the input audio itself, we find that various types of prompts are useful for obtaining coherent samples, as they lower the amount of multimodality when sampling from the model. Like AudioLM, we can prompt our model with prefix audio of some duration (usually between 1 and 4 seconds), and it will provide a continuation of that audio. Unlike AudioLM, and other auto-regressive approaches, we can also prompt our model with suffix audio, and it will generate audio that leads up into that suffix. We can provide prefix and suffix audio, and the model will generate the remaining audio, such that it is appropriate, giventhe specified prefix and suffix.

We can also apply a "periodic" prompt, where all but every Pth timestep are masked.The lower P is, the more the generated audio will sound like the original, as the model is highly conditioned. For example if $P = 2$, then the model is essentially behaving like a upsampler, imputing the tokens for every other timestep. As P increases, the model shifts from behaving in a *compression* mode to a *generative* mode, creating variations that match the style of the original.

Another useful style of prompt are "compression" prompts, where all codebooks other than the most coarse-grained are masked. This gives the model strong conditioning on every timestep, so the model is likely to produce audio that closely matches the original. We can combine this prompt with a periodic prompt with low P for even more extreme compression ratios. Given the bitrate of the codec B , which has number of codebooks N, a downsampling rate P for the periodic prompt, and a number of kept codebooks N_k, we can achieve a bitrate of $B/P(N - N_k)$.

Finally, we can design music-specific prompts, which exploit knowledge about the structure of the music. More concretely, we explore beat-driven prompting, where timesteps that fall on or around the beat are left unmasked. The model is left to create music between these beats, resulting in interesting variations on the original music. These prompts can all be combined to create a very useful music creation tool. In concert with a well designed user interface, VampNet shows promise as the basis for a next-generation music editing and creation suite.

4. EXPERIMENTS

Our experiments aim to evaluate VampNet's capability to both compress and generate music, given the various prompting strategies described in Section 3.4. For our objective audio quality measures, we use a multiscale mel reconstruction error and the Fréchet Audio Distance (FAD). Mel-reconstruction error is defined as the $L1$ distance between log-mel spectrograms at various time-scales,

$$\mathrm{D}_{F,M} = ||\hat{S}_{F,M} - S_{F,M}||_1 \quad (3)$$

where F is the FFT size of each spectrogram, and M is the number of mel-frequency bins. We use $F \in [2048, 512]$ and $M \in [150, 80]$, with a hop size of $\frac{1}{4}$ the FFT size. Mel-reconstruction is valuable as a metric for compression quality, but not for generation quality, since it is likely that models produce audio that does not match one to one with the original target audio. For generation quality, we use FAD, which measures the overlap between distributions of real and generated audio. Unlike mel-reconstruction, FAD is geared more towards evaluating if sample quality falls within the data distribution of the real audio, and can be used to evaluate generation quality.

[5] $k = \gamma(\frac{t}{t_T})D$, where t is the current iteration, t_T is the total number of iterations, and D the total number of tokens in the sequence. The scheduling function γ is a cosine schedule.

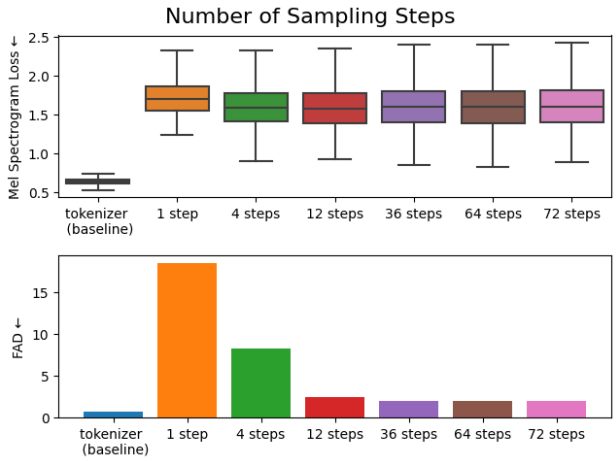


Figure 3. Mel reconstruction error (top) and Fréchet Audio Distance (FAD, bottom) for VampNet samples taken with varying numbers of sampling steps, taken using a periodic prompt of $P = 16$. The samples were generated by de-compressing tokens at an extremely low bitrate (50 bps), effectively generating variations of the input signals.

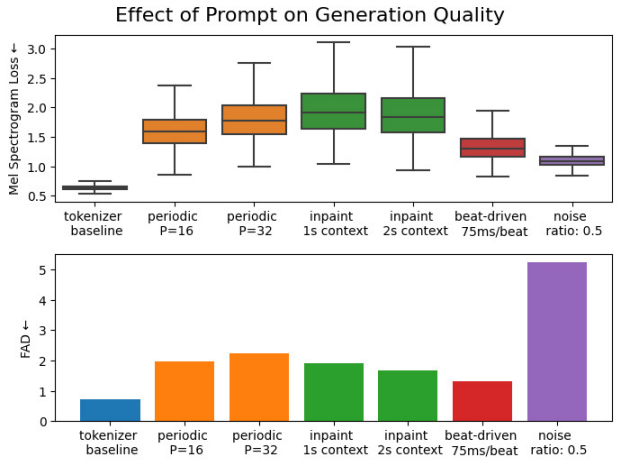


Figure 4. Multiscale Mel-spectrogram error (top) and Fréchet Audio Distance (FAD, bottom) for VampNet 10s samples taken with a different types of prompts.

4.1 Dataset

Similar to JukeBox [17], we collect a large dataset of popular music recordings. Our dataset consists of 797k tracks, with a sampling rate of 32 khz. These tracks are resampled to 44.1kHz to make compatible with our tokenizer. Our dataset contains music from thousands of artists across genres described in Echo Nest's Every Noise at Once [6].

We use a subset of 2k tracks for validation, and another subset of 2k tracks for testing. We ensure that there is no artist overlap between train, validation, and test tracks. In addition, we collect a set of music and non-music data (speech, environmental sound), which we used to train our tokenizer, using the datasets described in DAC [15]. All audio is normalized to -24dbFS. We do not use any metadata about these files during training, as our model is trained unconditionally.

4.2 Network Architecture and Hyperparameters

The audio tokenizer model we use takes as input 44.1kHz audio, and compresses it to a bitrate of 8kbps using 14 codebooks, with a downsampling rate of 768x. The latent space therefore is at 57Hz, with 14 tokens to predict at every timestep. We designate 4 of these tokens as the coarse tokens, and the remaining 10 as the fine tokens. Refer to the Descript Audio Codec [15] for details on the tokenizer architecture. We train the tokenizer for 250k steps.

The VampNet architecture (for both coarse and coarse-to-fine models) consists of a bidirectional transformer [18] with relative attention [30] and an embedding dimension of 1280 and 20 attention heads. The coarse model has 20 attention layers, while the coarse-to-fine model has 16. We train the coarse and coarse-to-fine model for 1M and 500k steps, respectively. We train with the AdamW optimizer [31] with β_1 and β_2 set to 0.9 and 0.999, respectively. We use the learning rate scheduler introduced by Vaswani et al [18] with a target learning rate of 0.001 and 10k warmup steps. We use a dropout of 0.1, and a batch size of 25, with a GPU memory budget of 72GB.

4.3 Efficiency of VampNet

We first validate that VampNet can generate realistic music audio in a low number of steps. To do this, we run VampNet using one of our prompts (the periodic prompt, with $P = 16$) on our test set, on 10-second excerpts. We vary the number of sampling steps in $[1, 4, 8, 12, 36, 64, 72]$, and report metrics for each sampling step.

4.4 Effect of prompts

We seek to understand how VampNet responds to different prompts, as discussed in Section 3.4. The prompts range from "compression" prompts, which compress music to a low bitrate, to more creative "generative" prompts. We examine whether compression and generative prompts exist on a continuum, and whether decompression from low bitrates results in generative behavior.

We draw 2000 10-second examples from our evaluation dataset, encode them into token streams with our audio tokenizer, and manipulate the token streams in four ways:

1. Compression prompt: C codebooks are left unmasked, starting from the coarsest codebook. All other tokens are masked. We set $N_k = 1$.
2. Periodic prompt: every Pth timestep is left unmasked. In an unmasked timestep, tokens from every codebook are unmasked. All other tokens (e.g. tokens in timesteps that do not correspond to the period P) are masked. We set $P \in [8, 16, 32]$.
3. Prefix and suffix (inpaint) prompts: a segment at the beginning and at the end of the sequence is left unmasked. All other tokens are masked. This prompt is parameterized by a context length in seconds. We set the context to be either 1 second or 2 seconds, which corresponds to 57 or 114 timesteps.

[6] https://everynoise.com/engenremap.html

4. Beat-driven prompt: we first process the audio waveform with a beat tracker [32]. Then, around each detected beat, we unmask timesteps to the right of the beat. We examine a 75ms unmasked section around each beat, which is about 4 timesteps per beat.

After manipulating the input token streams with our prompts, we generate new musical signals from these masked token streams using VampNet, and compute FAD and mel-reconstruction error between the generated signals and the input signals from our music dataset. We include a noisy token stream baseline, where a portion (as dictated by mask ratio r) of the tokens in the input token stream are replaced with random tokens. We also include as baseline the codec by itself, as well as the coarse-to-fine model.

Finally, we examine how these prompts can be combined - specifically the compression and periodic prompts. By manipulating the hyperparameters of these prompts (C and P), we can shift the model behavior from compression to generation. As more timesteps are masked, the model must generate plausible musical excerpts that connect the unmasked timesteps, that may not match the input music.

5. RESULTS AND DISCUSSION

Results for our experiment varying the number of sampling steps used to generate samples with VampNet are shown on Figure 3. We find that VampNet achieves the lowest FAD with 36 sampling steps, although 12 sampling steps achieves comparable performance. In practice, we find that samples taken with 24 steps achieve a fair tradeoff between generation quality and compute speed, with 10-second samples taking around 6 seconds to sample on an NVIDIA RTX3090. In contrast, to generate 10 seconds of audio with an autoregressive model would require 574 steps, which would take around 1 min to generate 10 seconds of audio, given an autoregressive model with the same number of parameters as ours, and the same tokenizer.

Results for our study on the effect of each prompt are shown in Figure 4. First, we note that while the noisy token baseline has comparable mel reconstruction to all prompts, it performs very poorly in terms of FAD. This indicates that while our prompting strategies may result in audio that is not a perfect match to the original input audio, it still falls inside the distribution of plausible music.

Of our proposed prompts, we find that beat-driven prompts perform best, achieving the lowest FAD of all prompts. A notable result here is that the periodic prompt with $P = 16$ (35 conditioning timesteps) performs on par with inpainting with 1 second of context (57 conditioning timesteps). Therefore, prompt techniques that spread out the conditioning tokens throughout the sequence (periodic prompts) are able to use fewer conditioning timesteps to generate samples of comparable quality to those generated by sampling techniques that place all of the conditioning tokens at the start and end of the sequences (inpainting).

Qualitatively, we also find that beat-driven prompts can keep a steadier tempo than other prompts, though their outputs tend to resemble the original music closer than periodic prompts. In practice, a mix of beat-driven, periodic, and inpainting prompts can be employed to steer of VampNet in creative ways. To illustrate, we highly encourage the reader to listen to the accompanying sound samples [7].

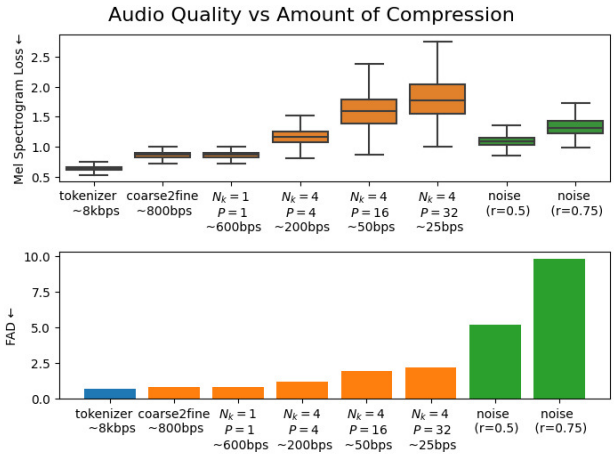


Figure 5. Mel-spectrogram error (top) and Fréchet Audio Distance (FAD) (bottom) for VampNet samples at varying bitrates. A baseline is provided by replacing tokens in the input sequence with random tokens, per noise ratio r.

We then combined periodic and compression prompting to show how the model's behavior shifts between reconstruction and generation tasks, as more tokens are masked away. Results for this experiment are shown in Figure 5. At higher bitrates, (600 bps and above), VampNet is able to accurately reconstruct the original music signal, achieving low mel-spectrogram error and FAD values with respect to the evaluation music audio. At bitrates of 200bps and below, VampNet has comparable reconstruction quality to the noisy token baselines, indicating that the sampled VampNet signals no longer resemble the input audio in terms of fine-grained spectral structure. However, the FAD for VampNet samples at low bitrates is much lower than the FAD for noisy baselines. This indicates that even though VampNet isn't able to reconstruct the input music signal at low bitrates, it is still able to generate coherent audio signals with musical structure, that are closer to the distribution of "real music" than our noisy baseline.

6. CONCLUSION

We introduced VampNet, a masked acoustic token modeling approach to music generation. VampNet is bidirectional, and can be prompted a variety of ways using an input audio file. Through different prompting techniques, VampNet can operate in a continuum between music compression and generation, and is an excellent tool for generating variations on a piece of music. With VampNet, a musician could record a short loop, feed it into VampNet, and have VampNet create musical variations on the recorded idea every time the looped region repeats. In future work, we hope to investigate the interactive music co-creation potential of VampNet and its prompting techniques, as well as explore the representation learning capabilities of masked acoustic token modeling.

[7] audio samples: `https://tinyurl.com/bdfj7rdx`

7. REFERENCES

[1] C. Wang, S. Chen, Y. Wu, Z. Zhang, L. Zhou, S. Liu, Z. Chen, Y. Liu, H. Wang, J. Li *et al.*, "Neural codec language models are zero-shot text to speech synthesizers," *arXiv preprint arXiv:2301.02111*, 2023.

[2] Z. Borsos, R. Marinier, D. Vincent, E. Kharitonov, O. Pietquin, M. Sharifi, O. Teboul, D. Grangier, M. Tagliasacchi, and N. Zeghidour, "Audiolm: a language modeling approach to audio generation," *arXiv preprint arXiv:2209.03143*, 2022.

[3] 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.

[4] 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.

[5] D. Rampas, P. Pernias, E. Zhong, and M. Aubreville, "Fast text-conditional discrete denoising on vector-quantized latent spaces," *arXiv preprint arXiv:2211.07292*, 2022.

[6] Z. Borsos, M. Sharifi, D. Vincent, E. Kharitonov, N. Zeghidour, and M. Tagliasacchi, "Soundstorm: Efficient parallel audio generation," 2023.

[7] P. Esser, R. Rombach, and B. Ommer, "Taming transformers for high-resolution image synthesis," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 12 873–12 883.

[8] 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.

[9] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Défossez, "Simple and controllable music generation," 2023.

[10] 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," *arXiv preprint arXiv:2010.11929*, 2020.

[11] A. Van Den Oord, O. Vinyals *et al.*, "Neural discrete representation learning," *Advances in neural information processing systems*, vol. 30, 2017.

[12] C. Gârbacea, A. van den Oord, Y. Li, F. S. Lim, A. Luebs, O. Vinyals, and T. C. Walters, "Low bit-rate speech coding with vq-vae and a wavenet decoder," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 735–739.

[13] 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.

[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," 2023.

[16] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever *et al.*, "Improving language understanding by generative pre-training," 2018.

[17] 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.

[18] 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.

[19] M. Bavarian, H. Jun, N. Tezak, J. Schulman, C. McLeavey, J. Tworek, and M. Chen, "Efficient training of language models to fill in the middle," *arXiv preprint arXiv:2207.14255*, 2022.

[20] 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*, 2018.

[21] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 16 000–16 009.

[22] 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.

[23] J. Austin, D. D. Johnson, J. Ho, D. Tarlow, and R. van den Berg, "Structured denoising diffusion models in discrete state-spaces," *Advances in Neural Information Processing Systems*, vol. 34, pp. 17 981–17 993, 2021.

[24] Y. Song and S. Ermon, "Generative modeling by estimating gradients of the data distribution," *Advances in neural information processing systems*, vol. 32, 2019.

[25] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in Neural Information Processing Systems*, vol. 33, pp. 6840–6851, 2020.

[26] A. Srivastava, L. Valkov, C. Russell, M. U. Gutmann, and C. Sutton, "Veegan: Reducing mode collapse in gans using implicit variational learning," *Advances in neural information processing systems*, vol. 30, 2017.

[27] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," *Advances in neural information processing systems*, vol. 29, 2016.

[28] 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.

[29] E. J. Gumbel, "Statistical theory of extreme values and some practical applications; a series of lectures." Washington, 1954.

[30] P. Shaw, J. Uszkoreit, and A. Vaswani, "Self-attention with relative position representations," *arXiv preprint arXiv:1803.02155*, 2018.

[31] I. Loshchilov and F. Hutter, "Fixing weight decay regularization in adam," *CoRR*, vol. abs/1711.05101, 2017. [Online]. Available: http://arxiv.org/abs/1711.05101

[32] C. J. Steinmetz and J. D. Reiss, "WaveBeat: End-to-end beat and downbeat tracking in the time domain," in *151st AES Convention*, 2021.

EXPERT AND NOVICE EVALUATIONS OF PIANO PERFORMANCES: CRITERIA FOR COMPUTER-AIDED FEEDBACK

Yucong Jiang
University of Richmond
yjiang3@richmond.edu

ABSTRACT

Learning an instrument can be rewarding, but is unavoidably a huge undertaking. Receiving constructive feedback on one's playing is crucial for improvement. However, personal feedback from an expert instructor is seldom available on demand. The goal motivating this project is to build software that will provide comparably useful feedback to beginners, in order to supplement feedback from human instructors. To lay the groundwork for that, in this paper we investigate performance assessment criteria from both quantitative and qualitative perspectives. We gathered 83 piano performances from 21 players. Each recording was evaluated by both expert piano instructors and novice players. This dataset is unique in that the novice evaluators are also players, and that both quantitative and qualitative evaluations are collected. Our analysis of the evaluations indicates that the kind of specific, concrete piano techniques that are most elusive to novice evaluators are precisely the kind of characteristics that can be detected, measured, and visualized for learners by a well-designed software tool.

1. INTRODUCTION

Learning to play a musical instrument can be rewarding, but is also unavoidably a huge undertaking. Receiving feedback on one's playing is crucial for improvement. However, personal feedback from an expert instructor is seldom available on demand; it is typically available (if at all) only in weekly music lessons. Our long-term goal in this project is to build software that will provide comparably useful feedback to beginners, as needed, in order to supplement insights from human instructors. The modes of computer-generated feedback could involve textual or visual indicators, or a mix of both. However, determining what kinds of feedback are especially helpful for beginners (among those that are feasible for computers to generate) is not trivial and should not be based on assumptions. To lay the groundwork for meaningful computer-aided feedback, therefore, in this paper we gather information on how experts and novices assess piano performances and what criteria they tend to rely on in such assessments.

Recent years have seen rapid growth in commercial products for computer-assisted instrumental learning. Unfortunately, most applications cannot deal with performances involving expressive timing: they expect users to play at a fixed tempo throughout a piece, even though such performances in real life are often perceived as boring and far short of the full expressive potential of music. For example, Yousician [1] and Simply Piano [2] color correctly played notes as the user progresses through a song at a preset tempo. While platforms like this have their own purposes and values, such oversimplified music playing experiences can mislead some learners to think that making music is all about playing the correct notes (rather than better sounding notes). Moreover, real-time feedback could distract players from listening to themselves, and as Percival et al. [3] point out, "computer analysis and interaction should occur *after* a student has finished playing".

Therefore, for our purposes, it makes more sense to envision software that can analyze a complete performance recording before providing feedback. Given such a recording, we would like to investigate what additional evaluation criteria (beyond note accuracy) should be incorporated into the feedback. In fact, even beginner-level players can usually tell when they've hit wrong notes, as the music won't sound right, but they often lack the ability to make more sophisticated judgments about the quality of their playing: articulation, tempo control, dynamics, and interpretation or expressiveness. Therefore, in this paper we focus on analyzing performances that are relatively "correct" in terms of wrong notes, so that they are ready for more nuanced aspects to be evaluated.

We have gathered 83 such piano performances from 21 players, each of whom chose from among seven beginner pieces. Each recording was evaluated by four expert piano instructors, and also by 17 peers from among the players themselves, with both numerical ratings and written comments. In this paper we examine (1) whether instructors and players evaluate performances differently, (2) whether better players are also better evaluators, and (3) what objective indicators can be detected and measured by computers that would reflect comparable evaluation criteria. This dataset is unique in that the peer evaluators are also players, and both quantitative and qualitative evaluations are collected. Each performance has also been aligned to its score, making it possible in the future to derive addi-

© Y. Jiang. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Y. Jiang, "Expert and Novice Evaluations of Piano Performances: Criteria for Computer-Aided Feedback", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

tional objective measurements (e.g., tempo variations), and to support further analyses relating performances to their scores (e.g, inter-song performance analysis).

2. RELATED WORK

A recent review paper [4] offers a comprehensive discussion of computer-aided instrument learning. The authors emphasize the differences between systems that are designed to measure competence and those designed to enhance learning, as the former only need to provide a rating, but the latter need to provide descriptive evidence justifying the evaluation. Two other review papers [5] [6] discuss the potentials of utilizing MIR techniques in music education. Example work on piano music tutor systems include [7] [8] [9] [10] [11].

Music performance analysis (MPA) is a broader topic, encompassing other purposes and uses beyond assisting learners, but a recent review paper [12] does discuss its application potential and challenges in regard to music education. Another closely related topic is modeling expressive music performance [13] [14] , which focuses on more abstract and higher-level aspects of a performance.

Related to examining performance evaluations, [15] discusses subjectivity in music performance assessment, [16] investigates how individual raters differ in their rating scale structure, and [17] provides insights on the benefit of peer assessment of music performance.

3. DATASET DESCRIPTION

Our dataset includes three components: 83 piano performance recordings in the WAV format, spanning seven different musical pieces; 803 evaluations of these performances, with players' metadata; and 83 audio-to-score alignments (with seven MusicXML score files and 83 alignment text files) indicating the starting time in the audio of each musical note in each score. This dataset is publicly available at facultystaff.richmond.edu/∼yjiang3/papers/ismir23/.

3.1 Performance Recordings

We recruited 21 participants from a local college, using flyers and campus-wide email announcements. These participants represent a range of piano experience, from a low of three months to a high of 16 years. Each participant completed a short questionnaire before recording a performance for the project. Except for one music major and one music minor, the participants play piano as a hobby. More than one participant recounted the story that they took piano lessons growing up, played on-and-off throughout the years, and recently came back to practicing it in college. When asked to self-identify their piano skill levels, nine of the 21 described their skills as "advanced", eight as "intermediate", and four as "beginner". (None chose the "professional" category from our prompt.)

We selected seven pieces from a popular score book for adult group piano classes [18], and asked each player to play however many pieces they felt like from these. (This flexibility helped recruit lower-level players who might otherwise be intimidated by this task.) The sheet music was shared with them weeks in advance to allow time for practice and preparation. Table 1 provides the names of these pieces and the number of performances of each. The players were advised to warm up before a recording session, and when recording, were offered the option either to be left alone in the piano room (to decrease nervousness) or to have the researcher present. They were allowed to re-record multiple times until satisfied with their own playing (e.g., with the preponderance of notes played correctly).

Piece Name	#Measures	#Recordings
Careless Love	16	11
Cielito Lindo	16	6
Lavender's Blue	16	17
Over the Waves	32	11
She Wore a Yellow Ribbon	34	13
The Blues	16	17
The Entertainer	40	8

Table 1. Summary of performance recordings.

3.2 Performance Evaluations

To evaluate the quality of these recordings, we recruited four professional piano instructors and 17 out of the 21 players (the other four were unfortunately not available for this stage). The instructors all have doctoral degrees and at least two decades of teaching experience. Each performance was evaluated by all four instructors and at least five (sometimes six) randomly chosen peers, resulting in 803 evaluations in total. The evaluators were asked to provide a numerical rating from one (poor) to five (excellent) for each recording, and also to briefly explain the basis for their rating, describing what criteria they considered. We collected the evaluations through a web-based form where users can play the recordings (grouped by score), enter evaluations, and save their progress. The sheet music is also linked from the form. It took between two to three hours for each instructor to evaluate all 83 performances, and 50 minutes on average for each peer player to evaluate 27 or 28 performances.

3.3 Audio-to-Score Alignment and Web Interface

Based on the sheet music, we created seven digital scores in the MusicXML format, and aligned each performance to its score. The alignment was achieved by the hidden Markov model proposed in [19], with occasional manual corrections. Each of the 83 alignment files contains two columns of values: a musical time in the score, and its played time in the recording. The alignment can also support future analyses relating performance attributes to elements in scores. For example, one could easily investigate whether players tend to slow down at a particular measure.

To make it more convenient to explore this dataset, we have built a demonstration web-based interface where a user can select and play any of these performances, while looking at the sheet music with the currently

played notes highlighted. The evaluations of the selected performance are also shown on the same page. This interface can help anyone interested in this dataset to find connections among performances, scores, and evaluations. This website is publicly available at facultystaff.richmond.edu/∼yjiang3/papers/ismir23/.

4. QUANTITATIVE ANALYSIS

4.1 Self-Identified Piano Levels

To verify the accuracy of the performers' piano skill levels, we separate the performances into three groups according to their self-reported levels, and compare the evaluators' ratings of those performances in each group. Figure 1 compares three box plots, one for each group of performance ratings, where each performance rating in a group represents the average among the four instructors. Although the median rating increases with the skill levels, the three distributions overlap with each other. To test whether the difference between the advanced group's ratings and the intermediate group's is statistically significant, we conduct a one-sided Welch's t-test and get a p-value of 0.1826 (α = .05)—so we cannot say the former played better than the latter. (The beginner group's n is too small for statistical tests.) Although "beginner", "intermediate", and "advanced" categories are common labels applied to self-study courses and musical scores for amateur musicians, this result indicates that the ambiguity and subjectivity inherent in defining these categories make self-identified skill levels unreliable.

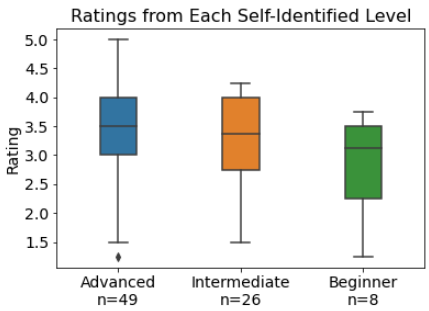


Figure 1. Comparing (averaged instructor) ratings among three self-identified groups.

4.2 Expert Evaluations

To examine how each instructor distributes their rating levels, we count the ratings at each level and compare their frequencies, as shown in Figure 2. It is clear that Instructors 1 and 2 tend to give high ratings more often than Instructors 3 and 4; the former also avoid giving the lowest rating almost completely. This indicates that the absolute rating values may be subjective and skewed.

Therefore, to measure how similarly these instructors rate, it makes more sense to compare ratings according to relative rather than absolute values. For this purpose we use Kendall's τ coefficient, which focuses on the rank correlation and can handle ties (with the tau-b version). Table

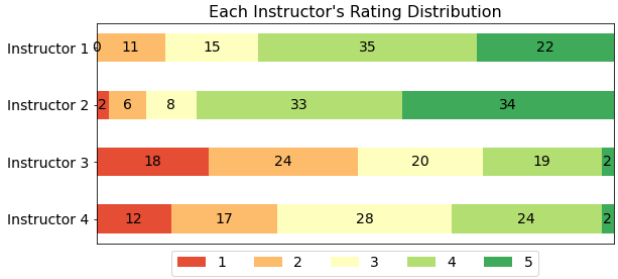


Figure 2. Comparing instructor rating distributions.

2 shows how each pair of instructors' ratings are associated with each other, with correlations sorted in descending order; all p-values are close to zero, statistically significant at the α = .01 level. These instructors show strong correlations (> .5) with one another, especially Instructor 3 and Instructor 4.

I3 & I4	I1 & I2	I2 & I4	I1 & I4	I1 & I3	I2 & I3
.806	.595	.563	.521	.514	.508

Table 2. Kendall's τ correlations (I=Instructor).

4.3 Peer Evaluations

As described in Section 3.2, 17 of the players also provided peer evaluations of the performances. To measure how the players' ratings compared to the instructors', we calculate the Kendall's τ correlation between ratings provided by each player and the average rating for the same recording subset provided by the instructors. Let's define k_p as the correlation for the pth player, where $p = 1, 2, \ldots, 17$. All k_p end up ranging between .401 and .741 (p-values < .01), with a mean of .542.

If we use k_p to represent the degree of "accuracy" of the pth player 's ratings, we can investigate the question of *whether better players are also better evaluators*. Let's define r_p as the average rating received by the pth player from all four instructors (for all pieces by this player). We use Spearman's ρ to measure how r_p and k_p are monotonically related, and the result is:

$$\rho_{r,k} = 0.152$$
$$p\text{-value} = 0.56$$

Although the correlation is positive, the large p-value prevents us from rejecting the null hypothesis that no relationship exists between how well individuals play and how accurately they rate performances.

5. QUALITATIVE ANALYSIS

5.1 Content Analysis and Annotation

To understand the evaluation criteria used by the instructors and the (novice) peer evaluators, we conduct a content analysis of their written comments [20]. The process involves first building an annotation model representing various evaluation criteria that appear in the text, and then

using this model to annotate each evaluation comment. (These two steps are iterative, as described later, in keeping with best practices for textual analysis [21].) For example, one of the comments—"The player didn't play the staccato notes in the left hand. No dynamic changes. A wrong note was played."—is annotated with *staccato*, *dynamic contrast*, and *wrong note* (terms that are then categorized under Articulation, Dynamics, and Note Accuracy respectively).

We use specialized text analytics software (QDA Miner, from Provalis Research) to construct the annotation model and to annotate each comment. To define appropriate annotations, we find recurring words and phrases (frequency ≥ 3), and look at each original comment in context to understand the intended meaning. For example, one of the most frequent phrases is "left hand", and one of its recurring contexts is that the left hand notes were played too loudly; therefore we create an annotation called *left hand loudness*. Other key words often associated with this aspect include "bass", as in "... I would like the bass [to] sound softer". By searching for related key words (e.g., "balance"), we have found similar contexts describing *right hand loudness* or just the *balance in general*. We group these conceptually related annotations under the same category called *Balance*. As we examine the contexts of these frequent words and phrases one-by-one, we create new annotations (and categories), and use them to annotate evaluator comments.

Building annotations and annotating comments is an iterative process: while examining the comments, we have discovered infrequent but useful key words like "8va" and "cresc" that we should search for. We sometimes carve out a new annotation from existing ones when observing enough cases to form a pattern (e.g., we have created a separate *tempo steadiness* annotation from *good tempo* and *inaccurate tempo*.) We have also spot-checked individual comments to make sure all evaluation criteria are sufficiently represented in our annotation model.

5.2 The Annotation Model

Figure 3 shows the annotation model developed from our dataset, containing 47 annotation terms arranged in 11 categories (and two subcategories). Many of these annotations can represent both positive and negative aspects of a performance: for example, *tempo steadiness* can be used to annotate both steady tempo and unsteady tempo. This is harmless, as our goal is to identify evaluation criteria, not the valence of the evaluations *per se*. A small handful of annotation terms exist only in the instructors' comments or only in the peers' comments, and these are mostly annotations in the Styles category: four styles are mentioned only by the instructors and five styles only by the peers. In addition, *dynamic shaping* (20 instances), *melodic shaping* (9 instances), and *rubato* (11 instances) only exist in the instructors' comments.

The annotation model derived from this dataset represents a diverse set of criteria, and it serves as a pool from which computers can select and generate measurements.

Tempo and timing
- inaccurate tempo
- good tempo
- tempo steadiness
- tempo contrast
- ritardando
- rubato
- pause

Note accuracy
- correct note
- wrong note
- missed note
- wrong octave

Phrasings
- phrasing
- melodic shaping

Pedal
- inaccurate pedal
- good pedal

Dynamics
- accurate dynamics
- inaccurate dynamics
- dynamic contrast
- dynamic shaping

Balance (between hands)
- balance in general
- left hand loudness
- right hand loudness

Articulation
- articulation in general
- legato
- staccato
- accent

Confidence
- confident or hesitant

Note connection
- choppy
- connectedness

Styles
- smooth
- heavy
- light
- abrupt
- crisp
- character
- lively
- flow
- lyrical
- mechanical
- style
- bland
- with emotion

Rhythm
- correct rhythm
- incorrect rhythm
- ***Notes too short***
  - shortened note
  - left hand too short
- ***Notes too long***
  - note too long

Figure 3. The annotation model. Lower case: annotations; bold: categories; italic: subcategories.

Many of the criteria are objective in nature: e.g., tempo change, note accuracy, and rhythm. These have low ambiguity and thus computational methods can detect them in a fairly straightforward manner; in fact, many traditional MIR techniques can be used for measuring these criteria. For example, we can easily track tempo changes based on audio-to-score alignment results (although deriving *perceived* tempo involves a few more parameters [22]). At the other end of the spectrum, however, criteria like confidence and style are very abstract, and thus are extremely hard for computers to detect. The rest of the criteria fall in the middle. Dynamics, phrasing, and articulation, for example, are directly linked to measurable features of the audio signal, but they involve many other parameters and can be subjective. Some literature addresses this duality (e.g., [23] on articulation and [24] on dynamic shaping), but there is no consensus on how to model such features, and attempts are scarcer than the more traditional MIR work mentioned above. Such aspects are almost never considered in computer-aided instrument learning applications.

5.3 Frequency of Evaluation Criteria

In the end, we have a total of 885 annotation instances for the instructors' comments, and 1015 for the peers' comments, averaging 2.7 and 2.2 annotations per comment respectively. We count the number of annotation instances under each annotation category separately for the instructors and for the peers, and calculate the frequency with which each annotation category was used by the two groups respectively. These percentages are shown in Figure 4. For both the instructors and the peers, Tempo, Note Accuracy, and Rhythm are the top three categories, accounting for just over 60% of the total annotations (although the peers describe Tempo more frequently and Rhythm less frequently than the instructors). For the remaining categories, Balance, Styles, and Confidence show the most difference between the two groups.

Balance is the fourth most common evaluation criterion found among the instructors' comments, accounting for 8.4% of the annotations, versus only 2.4% of the peers'. Balance between hands is a well-known challenge for piano beginners, and it can be difficult to notice on one's own; the percentage discrepancy between experts and novices suggests a promising opportunity for computer assistance. Meanwhile, Styles is the fourth most common criterion for the peer evaluators (9.4%), but it is only the eighth for the instructors (4.4%). Confidence accounts for 7.3% of peer annotations, but only 2.7% of instructor annotations. Styles and Confidence are both abstract concepts. For the instructors, these two are both ranked after Balance, Dynamics, Articulation, and Pedal, which represent concrete piano techniques, and they occupy 28.2% in total. In contrast, all these four categories have lower percentages for the peers, and they occupy only 18.2% in total. This discrepancy implies that as compared to experts, novices might be more likely to judge a performance using abstract concepts, while experts tend to point out specific piano techniques.

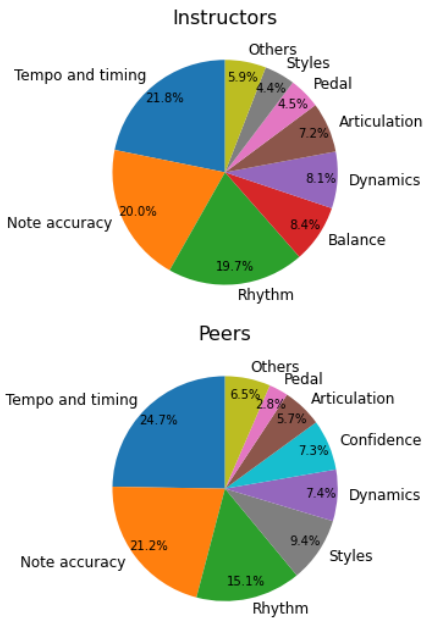


Figure 4. Comparing annotation category percentages.

We further investigate the usage of the four "technique" criteria and the two "abstract" criteria just described by comparing three groups among the peer evaluators: the four peer players whose performances received the lowest average ratings, the four peer players whose performances received the highest average ratings, and the four ranked in the middle. For each group, we calculate the usage percentages as above, and focus on comparing the six criteria. Figure 5 shows the comparison among the three groups, as well as how they compare to the instructors. The left bars indicate a consistent positive association between piano skill levels and the usage of piano technique criteria. Although the (opposite) trend of the right bars is less consistent, as the middle group used abstract criteria less frequently than the higher-skilled group, the lower-skilled group indeed used a significantly higher percentage of abstract criteria than the average of all 17 peer evaluators (16.7%). This suggests that lower-skilled piano players lack the ability to pin down specific piano techniques involved in a performance, and their evaluation criteria tend to be correspondingly more general and abstract, e.g., "There is some hesitancy in the chords. The bass clef chords are a bit abrupt".

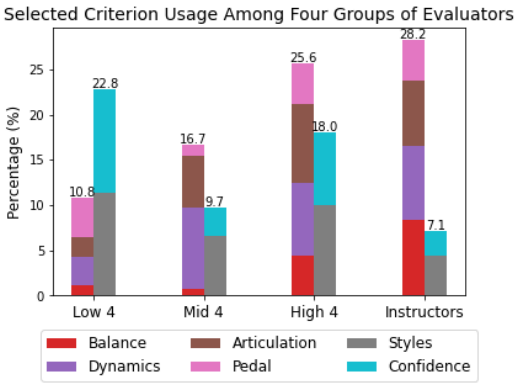


Figure 5. Comparing piano technique criteria (left) and abstract criteria (right) usage among three groups of peers and the instructors.

6. DISCUSSION

6.1 Do Instructors and Players Evaluate Performances Differently?

In terms of numerical ratings, the consistent high correlations between the peer evaluations and the average of the instructor evaluations (Section 4.3) suggest that even novices have a reliable sense of what good or poor performance is like. In terms of evaluation criteria, both the peers and the instructors use Tempo, Note Accuracy, and Rhythm the most, accounting for a little over 60% of the total comment annotations from both groups. However, beyond these top three criteria, the two groups exhibit different patterns: the peers tend to use more abstract and general criteria like Confidence while the instructors use more concrete and specific piano techniques like Balance (between hands). It is unsurprising that instructors would use more technical criteria, given their own training and teaching experience, and computer-generated feedback based on these criteria could be particularly illuminating to individuals seeking to improve their playing.

6.2 Are Better Players Also Better Evaluators?

For the sake of this discussion, we define good evaluations as evaluations similar to the ones done by the instructors. In terms of *ratings*, we do not find enough evidence confirming better player are also better evaluators—even players of poor performances can provide accurate ratings. However, in terms of evaluation *criteria*, we have found evidence that higher-skilled players tend to provide better

evaluations. Specifically, they are more capable of judging a performance based on piano techniques, which are in line with piano instructors' evaluations.

6.3 Computer-Aided Feedback

These results suggest that there is a consistent standard of good and poor performances—at least for beginner pieces. However, numerical ratings have limited value for helping students learn. In fact, students take private music lessons not to be given a rating, but to seek specific formative feedback for improvement. Building a machine that can provide comparable feedback would offer much greater value to end users.

Much of the terminology in the annotation model is score-dependent, verifying whether or not the performer has followed elements in the sheet music. The basic elements are notes, rhythm, and tempo/timing, which also correspond with the top three evaluation criteria. Relevant MIR tasks for detecting such errors include music transcription [25], source separation [26], and audio-to-score alignment [27]. Other typical elements in the sheet music are dynamics, articulation, and pedaling, and some attempts (such as [23] [28] [29]) have been made at modeling and detecting them.

However, not every element or aspect worth evaluating is explicitly indicated in the score. For example, pedaling and balance between hands are often only implied in the score, and can also be up to personal interpretation by the performer. In such instances, text-based feedback can be of limited utility, and what a computer may be able to do more effectively is provide visualized feedback. The value of such feedback lies in making implicit aspects of a performance explicit to the player, rather than instructing the player what to do. For example, the computer could show a tempo curve indicating (intentional or unintentional) tempo changes. Such visualizations can be especially helpful to beginners, who might not be able to notice such aspects easily.

6.4 Peer Evaluation for Education

At the end of each peer evaluation session, we asked the evaluator two open-ended questions: "How do you feel after listening to so many recordings in a row?" and "How do you feel about this process compared to how you evaluate your own playing?" Most evaluators expressed that it was a positive experience, with words like "fun", "very interesting", and "enjoyed it". A couple of them mentioned they were able to pay more attention to the elements in the sheet music when evaluating others. Four of them indicated that the process of comparing multiple recordings helped them judge their own playing better. This overall positive response suggests that there is educational potential for peer evaluation platforms where piano learners could anonymously give each other feedback.

6.5 Limitations

There are some inherent limitations in this project. First, the pieces we focus on are all at the beginner level, meaning they are relatively short and involve relatively few sophisticated piano techniques. It is possible that our findings might not apply to performances (or evaluations) of more advanced pieces. Second, the recording process might involve some bias against more advanced players who felt confident and could sight read, and thus did not prepare as much as the beginners. Third, the size of the dataset is relatively small. This facilitated our process of careful, manual content analysis, but imposes some limits on the statistical analysis.

7. CONCLUSIONS AND FUTURE WORK

In this paper, we contribute a unique dataset of amateur piano performance recordings and corresponding expert and peer evaluations. This dataset allows for interesting multifaceted analysis of nuances in the peer evaluations, because the peer evaluators are also players, and both quantitative and qualitative evaluations are recorded. Through the initial analyses presented in this paper, we find that even novices exhibit reliable judgement at distinguishing good performances from poor ones, but higher-skilled novices tend to base their judgement on piano techniques (as experts do), while lower-skilled novices rely on more subjective and/or abstract impressions. Most evaluation criteria used by experts are concrete, and are therefore precisely the kind that can be detected and measured by software evaluating an audio signal and its relationship to the score. Visualizing these aspects could provide valuable assistance to beginners seeking constructive insights on their playing. Despite some limitations to the generalizability of its findings, this paper lays the groundwork for building more advanced computer-aided instrument learning software. In future work, we plan to combine the audio-to-score alignments in this dataset with other MIR techniques to derive specific measurements reflecting experts' evaluation criteria. Once that is achieved, we then plan to compare those computer-generated evaluations of these recordings (including measurements and/or visualizations) to the human annotations.

8. ACKNOWLEDGMENTS

We extend our sincere thanks to multiple individuals for their contributions to this project. First, we thank the piano players and the piano instructors for participating in our experiments. Second, we thank Joon Han and Liz Smith for helping record the piano performances. Third, we thank Caitlin Sales for creating the demo website. We would also like to thank the reviewers of this paper for their detailed and constructive feedback.

9. REFERENCES

[1] Yousician website. https://yousician.com/.

[2] Simply Piano website. https://www.hellosimply.com/simply-piano.

[3] G. Percival, Y. Wang, and G. Tzanetakis, "Effective use of multimedia for computer-assisted musical instrument tutoring," in *Proceedings of the International Workshop on Educational Multimedia and Multimedia Education*, 2007, pp. 67–76.

[4] V. Eremenko, A. Morsi, J. Narang, and X. Serra, "Performance assessment technologies for the support of musical instrument learning," in *Proceedings of the 12th International Conference on Computer Supported Education (CSEDU)*, 2020, pp. 629–640.

[5] C. Dittmar, E. Cano, J. Abeßer, and S. Grollmisch, "Music information retrieval meets music education," in *Dagstuhl Follow-Ups*, vol. 3. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2012.

[6] P. Kasák, R. Jarina, and M. Chmulík, "Music information retrieval for educational purposes-an overview," in *2020 18th International Conference on Emerging eLearning Technologies and Applications (ICETA)*. IEEE, 2020, pp. 296–304.

[7] R. B. Dannenberg, M. Sanchez, A. Joseph, P. Capell, R. Joseph, and R. Saul, "A computer-based multimedia tutor for beginning piano students," *Interface*, vol. 19, no. 2-3, pp. 155–173, 1990.

[8] A. Arzt, S. Böck, S. Flossmann, H. Frostel, M. Gasser, and G. Widmer, "The complete classical music companion v0. 9," in *Proceedings of the AES International Conference on Semantic Audio, London, UK*, 2014, pp. 18–20.

[9] F. Tsubasa, Y. Ikemiya, K. Itoyama, and K. Yoshii, "A score-informed piano tutoring system with mistake detection and score simplification," in *Sound and Music Computing Conference*, 2015.

[10] S. Ewert, S. Wang, M. Müller, and M. Sandler, "Score-informed identification of missing and extra notes in piano recordings," in *Proceedings of the 17th International Society for Music Information Retrieval (ISMIR) Conference*, 2016.

[11] B. F. Yuksel, K. B. Oleson, L. Harrison, E. M. Peck, D. Afergan, R. Chang, and R. J. Jacob, "Learn piano with bach: An adaptive learning interface that adjusts task difficulty based on brain state," in *Proceedings of the CHI conference on human factors in computing systems*, 2016, pp. 5372–5384.

[12] 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, 2020.

[13] C. E. Cancino-Chacón, M. Grachten, W. Goebl, and G. Widmer, "Computational models of expressive music performance: A comprehensive and critical review," *Frontiers in Digital Humanities*, vol. 5, p. 25, 2018.

[14] C. Cancino-Chacón, S. Peter, S. Chowdhury, A. Aljanaki, and G. Widmer, "On the characterization of expressive performance in classical music: First results of the con espressione game," *arXiv preprint arXiv:2008.02194*, 2020.

[15] S. Thompson and A. Williamon, "Evaluating evaluation: Musical performance assessment as a research tool," *Music Perception*, vol. 21, no. 1, pp. 21–41, 2003.

[16] B. C. Wesolowski, S. A. Wind, and G. Engelhard Jr, "Examining rater precision in music performance assessment: An analysis of rating scale structure using the multifaceted rasch partial credit model," *Music Perception: An Interdisciplinary Journal*, vol. 33, no. 5, pp. 662–678, 2016.

[17] D. Blom and K. Poole, "Peer assessment of tertiary music performance: Opportunities for understanding performance assessment and performing through experience and self-reflection," *British Journal of Music Education*, vol. 21, no. 1, pp. 111–125, 2004.

[18] J. W. Bastien, *The older beginner piano course*. Kjos West, 1977.

[19] C. Raphael, "Automatic segmentation of acoustic musical signals using hidden Markov models," *IEEE transactions on pattern analysis and machine intelligence*, vol. 21, no. 4, pp. 360–370, 1999.

[20] T. G. Harwood and T. Garry, "An overview of content analysis," *The marketing review*, vol. 3, no. 4, pp. 479–498, 2003.

[21] Q. Deng, M. J. Hine, S. Ji, and S. Sur, "Inside the black box of dictionary building for text analytics: a design science approach," *Journal of international technology and information management*, vol. 27, no. 3, pp. 119–159, 2019.

[22] K. Seyerlehner, G. Widmer, and D. Schnitzer, "From rhythm patterns to perceived tempo," in *Proceedings of the 8th International Society for Music Information Retrieval (ISMIR) Conference*, 2007.

[23] 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.

[24] G. Widmer and A. Tobudic, "Playing Mozart by analogy: Learning multi-level timing and dynamics strategies," *Journal of New Music Research*, vol. 32, no. 3, pp. 259–268, 2003.

[25] E. Benetos, A. Klapuri, and S. Dixon, "Score-informed transcription for automatic piano tutoring," in *2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*. IEEE, 2012, pp. 2153–2157.

[26] S. Ewert and M. Müller, "Score-informed source separation for music signals," in *Dagstuhl Follow-Ups*, vol. 3. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2012.

[27] R. Agrawal and S. Dixon, "A hybrid approach to audio-to-score alignment," *arXiv preprint arXiv:2007.14333*, 2020.

[28] K. Kosta, "Computational modelling and quantitative analysis of dynamics in performed music," Ph.D. dissertation, Queen Mary University of London, 2017.

[29] B. Liang, G. Fazekas, and M. B. Sandler, "Detection of piano pedaling techniques on the sustain pedal," in *Audio Engineering Society Convention 143*. Audio Engineering Society, 2017.

CONTRASTIVE LEARNING FOR CROSS-MODAL ARTIST RETRIEVAL

Andres Ferraro **Jaehun Kim** **Sergio Oramas**
Andreas Ehmann **Fabien Gouyon**
Pandora-SiriusXM, Oakland
andres.ferraro@siriusxm.com

ABSTRACT

Music retrieval and recommendation applications often rely on content features encoded as embeddings, which provide vector representations of items in a music dataset. Numerous complementary embeddings can be derived from processing items originally represented in several modalities, e.g., audio signals, user interaction data, or editorial data. However, data of any given modality might not be available for all items in any music dataset. In this work, we propose a method based on contrastive learning to combine embeddings from multiple modalities and explore the impact of the presence or absence of embeddings from diverse modalities in an artist similarity task. Experiments on two datasets suggest that our contrastive method outperforms single-modality embeddings and baseline algorithms for combining modalities, both in terms of artist retrieval accuracy and coverage. Improvements with respect to other methods are particularly significant for less popular query artists. We demonstrate our method successfully combines complementary information from diverse modalities, and is more robust to missing modality data (i.e., it better handles the retrieval of artists with different modality embeddings than the query artist's).

1. INTRODUCTION AND RELATED WORK

The MIR community has dedicated significant effort to defining and computing music similarity in the last 20 years. Music similarity can be used in multiple downstream tasks, from playlist continuation, music visualization/navigation, music categorization for organizing catalogs, or for personalized recommendations. The notion of similarity is subjective and there is no consensus on how to define and evaluate it [1]. To evaluate the performance of a music similarity algorithm, some previous works either focus on content-based aspects, such as melody or harmony. Other works measure similarity based on *cultural aspects*, such as based on the co-occurrence of items in playlists or on editorial data –this is the approach of our work.

© A. Ferraro, J. Kim, S. Oramas, A. Ehmann and F. Gouyon. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** A. Ferraro, J. Kim, S. Oramas, A. Ehmann and F. Gouyon, "Contrastive learning for cross-modal artist retrieval", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

Multiple methods have been proposed to compute music similarity based on a variety of data types related to the music, e.g., based on audio descriptors [2], document similarity [3], or graphs of musical connections [4, 5]. Some relatively recent works propose ways to produce embeddings –that can be used to compute music similarity– in a supervised or unsupervised way, by training models on large amounts of data (such as audio, text or image). Such pre-trained models, which are often released publicly, may produce feature representations –i.e. embeddings– that are effective for previously unseen tasks. Such embeddings can be computed from diverse types of modalities related to music such as audio [6–8], tags [9], album covers images [10], or biographies [4]. The multiple modalities of data that can describe a music item –such as audio, tags, or listening interactions– may contain *complementary* information. For example, the quality and scale of audio vs collaborative data has been shown to have significant influence in autotagging tasks [11]. It therefore appears beneficial to combine diverse complementary modalities to obtain a more informative representation of music items. In fact, recent research identifies the combination of diverse sources of data as specially promising for mitigating limitations and issues in music recommendation research [12].

Another aspect to take into account is that in any given music dataset, data of diverse modalities might be available for different subsets of items. Therefore, when querying with an item represented in a given modality, the maximum coverage for retrieval is limited to items for which that same modality is available, leaving out a potentially significant –and relevant– part of the dataset. For example, the availability of listening interactions or users' explicit feedback is highly dependent on item popularity. Therefore, for artists with very little listening and user feedback, it may not be possible to obtain embeddings from that modality. Embeddings from other modalities may suffer from the same issue, either because there is no data available to produce an embedding or because the quality of the available information is very low. For instance in the case of a model trained on tag annotations to produce artist embeddings, where the output embedding may not be very informative for those artists that have a single or few tag annotations. Such issues are particularly common and problematic emerging or more underground artists, for which the available information is more limited.

In order to mitigate the issue of availability of some modalities, it is important to combine and take full ad-

vantage of all information available so that when querying with an artist that has only one modality available, we can also retrieve artists for which we have a different modality information. Therefore, the focus of this work is to combine diverse modalities into a common shared space that is beneficial for 1) leveraging each modality information from the artists, and 2) allowing to operate on a single space that covers the full population of artists, ensuring that whether or not an artist is retrieved for another does not depend on the number of modalities available.

The problem of combining embeddings from diverse modalities in a shared representation has received some attention in the last few years. In the music domain, there have been some works on combining embeddings by simple concatenation [13] or predicting one modality from another [14]. Contrastive learning techniques go beyond simple concatenation or prediction, trying to learn a shared representation between embeddings from different modalities. Some examples of research related to multimodal contrastive learning can be found in [10], where embeddings from a shared multimodal space are used as additional features for classification, or in [15, 16] where, e.g., music audio can be retrieved from natural language descriptions. In this work, we propose to apply a contrastive learning method that maps embeddings from diverse modalities to a shared embedding space, extending the advantages of multiple modalities to populations that would not be covered otherwise.

In summary, in this work we propose an approach to combine the multiple encoders of a contrastive learning method, showcasing several improvements over baselines and single-modality approaches in an artist similarity task. We show under two different contexts –using an open and an in-house dataset– that our proposed approach:

- achieves higher performance in terms of accuracy and coverage of retrieved artists (§ 3.1),
- successfully combines complementary information from diverse modalities (§ 3.2),
- is more robust to missing modality data (§ 3.3),
- particularly increases the performance for less popular query artists (§ 3.4).

2. METHODOLOGY

2.1 Single-Modality Embeddings and Contrastive Method

In this work, we use three modalities, namely: tags, user-listening interactions (i.e. collaborative filtering data, referred to as CF), and audio information. In all cases, we use pre-trained models to obtain embeddings for each of the modalities. We evaluate artist similarity performance using the embeddings from the pre-trained models directly, and compare to the performance when using the embeddings produced by our contrastive method which is trained with the same embeddings from pre-trained models.

In these experiments we apply a contrastive learning loss based on InfoNCE [17]. Specifically, we define the contrastive loss between two modalities, $\boldsymbol{\psi}_a$ and $\boldsymbol{\psi}_b$, as: $\mathcal{L}_{\boldsymbol{\psi}_a,\boldsymbol{\psi}_b} = \sum_{i=1}^{M} -\log \frac{\Xi(\boldsymbol{\psi}_a^i,\boldsymbol{\psi}_b^i,\tau)}{\sum_{k=1}^{2M} \mathbb{1}_{[k\neq i]}\Xi(\boldsymbol{\psi}_a^i,\boldsymbol{\zeta}^k,\tau)}$, where M is the batch size and τ is the temperature parameter. We define $\Xi(\mathbf{a},\mathbf{b},\tau) = \exp(\cos(\mathbf{a},\mathbf{b})\tau^{-1})$, based on the cosine similarity. $\boldsymbol{\zeta}^k$ is defined as $\boldsymbol{\psi}_a^k$, if $k \leq M$ and else $\boldsymbol{\psi}_b^{k-M}$. This loss function attempts to minimize the distance between the modalities of the same artist while maximizing the distance with any modality from other artists.

We use three encoders –one for each modality– that will produce three representations in our shared space for each artist. During training [1] we minimize the sum of the pairwise losses between each of the modalities as in [18]: $\mathcal{L}_{\text{tot}} = \mathcal{L}_{\text{Audio-Tag}} + \mathcal{L}_{\text{Audio-CF}} + \mathcal{L}_{\text{Tag-CF}}$

Once the model is trained with the contrastive method and we want to use it for inference, for a given artist, we aggregate the output of each internal encoder by averaging all available information.

2.2 Training Data

In order to investigate the effectiveness of our contrastive method under different situations, we train our model using two independent datasets: We use a dataset based on public data to facilitate the reproducibility of some of the results. And we also use an in-house dataset that contains multimodal information for a larger set of artists.

Training our model requires *full coverage of the three modalities for all artists* –tag-based embeddings, CF embeddings, and audio embeddings. For the public dataset, we use the Million Song Dataset (MSD) [19] and its connections with other datasets to collect tags, audio and CF embeddings. We collected audio track embeddings using the public unsupervised model from [6] to extract embeddings from MSD audio previews, then we averaged all audio tracks embeddings for each artist. The tagging data was collected from the MSD500 dataset [11] and embeddings were computed using PMI factorization [13] of 500 tags. The CF embeddings were obtained using weighted matrix factorization [20] based on the Echonest Profile dataset, [2] with Gaussian process-based Bayesian hyperparameter tuning [21]. We gathered information from the three modalities for $17,478$ artists.

For the in-house dataset (hereafter, OWN) we collected tags, CF, and audio information for $38,301$ artists. This dataset is larger than MSD and includes what we believe is *higher-quality tags and CF data*, which allows us to compare the performance of our approach in a different setting. The CF information is computed from very large amounts of user-listening interactions on a streaming platform. The audio embeddings are computed using the supervised model [3] described in [6]. The tag embeddings are

[1] For both datasets we use Adam optimization with a learning rate of 0.0001 and temperature of 0.1. We use a fully connected layer of 256 for the CF encoder, two layers with 512 and 256 for the Audio encoder and 4 attention heads of 256 for the tag encoder. The learned space has 200 dimentions. Batch size for C_{OWN} is 2048 and for C_{MSD} is 128.

[2] Specifically, we aggregated the per-song listening counts corresponding artists such that we obtain the 'user-artist' listening matrix.

[3] i.e. a *different* model for audio embeddings than when training on MSD.

computed using PMI factorization from a total of $6,421$ different tags, which are a combination of manual and automatic annotations. Since our pre-trained models for audio and CF are at the track level, we compute artist embeddings by averaging over artist track embeddings.

In the remainder of this work, we refer to the model trained with the contrastive method with in-house data as C_{OWN} and the model trained with public data as C_{MSD}.

2.3 Evaluation Dataset

The ground truth for artist similarity is defined herein by the OLGA public dataset [22], containing artist similarity information collected from AllMusic. Our evaluations are therefore based on a *cultural* ground-truth, following [5].

We collected data from the MSD dataset for the original $17,646$ artists in OLGA. We obtained tag data from the MSD500 for $10,971$ (62%) artists, user interaction data from the Echonest Profile dataset for $15,389$ (87%) artists, and audio embeddings using MSD audio previews for 100% of the artists. [4]

We also create a subset of OLGA where *all* artists contain complete tags, user interaction, and audio information from MSD. We refer to this subset as OLGA Full Modality Coverage (FMC), which contains $9,474$ artists and it is also mapped to our internal dataset. The OLGA FMC subset is used to compare the results of multiple methods pre-trained on different and independent datasets.

2.4 Evaluation Conditions

In order to provide insights on the performance of the contrastive method, we conduct analyses under 3 different situations, varying the degree of availability of the different modalities in the evaluation data:

Raw evaluation dataset: In one condition, we compare the methods using all the artists in the OLGA dataset. In this case, we are interested to understand performance in a scenario of a real –uncontrolled– evaluation dataset, accounting for some organic imbalance of the availability of data in different modalities.

Full Modality Coverage: In another condition, we use the OLGA FMC subset where all artists contain CF, tags, and audio embeddings in both MSD and OWN datasets. In this case, we want to understand performance while factoring out the potential influence of one or another modality being only partially available in evaluation.

Systematic variation of modality coverage: We also perform multiple comparisons by grouping artists from OLGA depending on how many modalities are available. Here, we want to look at how much the contrastive method and the baselines are capable of doing cross-modality retrieval when using different modalities as input. In particular, we want to see whether or not they are capable of retrieving artists that have different modality information available compared to the query artists. Therefore, in this part, we create 7 groups of artists –at random– of equal size with each group containing one, two, or three modalities (namely, CF, audio, tag, CF+audio, CF+tag, audio+tag, audio+CF+tag). We refer to these groups as 'Modality Groups'. It is important to highlight the artificiality of this setting. We are considering an extreme case only to evaluate cross-modality retrieval capabilities of the methods. We are not considering here the accuracy of these results since it is already evaluated in the other analyses.

2.5 Baseline multimodal approaches

For multi-modal baselines, we employ two conventional models: PCA, and Gaussian random projection [23, 24] (which we refer to as Rand). [5] For fitting these models, we consider artists who have access to all modalities. Their multimodal embeddings are concatenated and treated as a single feature vector. It yields a dimensionality of $2,063$ for the MSD dataset, and $2,528$ for the OWN dataset. We set the reduced dimensionality to 200, which is the same size as the embeddings of the contrastive model. If an artist has a missing modality in the prediction phase, we employ the global mean embedding of the missing modality. [6]

2.6 Metrics

Accuracy: We consider nDCG@200 to measure how accurate the retrieved artists are compared to the ground truth while taking into account the position in the ranking of the retrieved artists, a metric considered robust to missing relevance information [26]. [7]

Distribution: We also compute the Gini@200 index, measuring the distribution of the top 200 retrieved artists in each experimental condition across the whole set of artists. A lower value of Gini indicates that the recommendations across artists are more uniformly distributed –covering more artists retrieved– while a higher value of Gini indicates that the recommendations are focused on only the few same artists.

We compute the confidence interval using the bootstrap method [27] on the evaluation artist population. We report them in Figure 1 at 95% confidence level.

Expected Contrastive Loss: We propose an additional metric that we named Expected Contrastive Loss (ECL). We use this measure to analyze to what extent an artist is coherent with respect to their multimodal representations. From how we defined the loss in Section 2.1, a high loss value implies that the artist is relatively difficult to be distinguished from other artists. Once the training is reasonably progressed, we employ ECL to quantify how "coherent" the artist is with respect to their internal representations obtained from the different modalities, which is defined as: $ECL(i,u,v) = d_{ii}^{uv} - \mathbb{E}_{j \setminus i}[d_{ij}^{uv}]$,

[4] Note that we don't control for artist separation between MSD, OWN and OLGA. But even if some artists may be present in both train and test sets, the artist similarity information from OLGA is *only* used for evaluation, and is never used during the training of the single-modality embeddings nor the contrastive models on either MSD or OWN.

[5] For both algorithms, we employ the standard implementation provided from `scikit-learn` [25].

[6] This does not happen in FMC

[7] We focus on nDCG@200 in this work, as we experimentally observed high correlation with other retrieval metrics such as precision, recall, and R-Precision.

where i and j denote artist index, while u and v refer to the modality index. d_{ij}^{uv} means the cosine distance between artist i from modality u and artists j from modality v. Taking expectation over all the possible modality pairs leads to the final coherency measure for artist i: $ECL(i) = \mathbb{E}_{u,v \setminus u}[ECL(i, u, v)]$.

Clustering: We further analyze the multimodal embedding space of the contrastive model, by investigating how well the artist embeddings are clustered. The contrastive method essentially can be seen as a "supervised" clustering task, where we minimize the distance among "positive points" (i.e., multimodal embeddings from an artist) and maximize the distance between those to the "negative points" (i.e., embeddings belonging to the other artists). It implies that an artist will get a higher training loss when the embeddings are dispersed and overlapped with the embedding cluster of other artists, while the opposite cases will get lower values. The model will fit the multimodal embedding space such that the artist embeddings poorly clustered initially have more concentrated and distant clusters. While the contrastive learning implements this naturally by its loss function, there are other well-known measures for the validation of the clustering methods, such as *intra-cluster distance (CD_{intra})* indicating how an artist embeddings are well clustered together, and *inter-cluster distance (CD_{inter})* indicating how an artist-specific embedding cluster is far and distinct from others'. [8]

3. RESULTS

3.1 Performance comparison of contrastive method

We now look at the performance of the contrastive method when some modality information is missing in the evaluation dataset (using the raw OLGA dataset) and when all modalities are available for each artist (FMC subset). We also compare the performance of the contrastive method to the baseline methods and to single-modality embeddings.

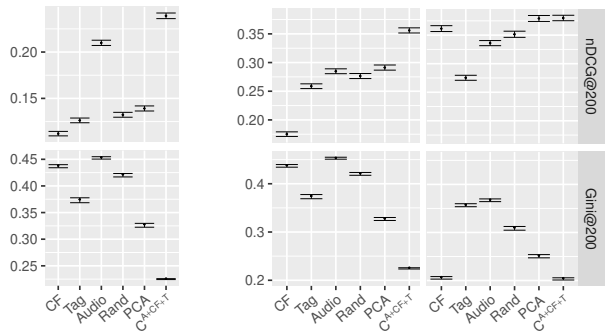
3.1.1 Performance with incomplete modality information

Focusing on the different combinations of input modalities to the contrastive method, we can see in Table 1 that the highest nDCG result is obtained when combining all modalities as input. We therefore focus only on this model for the remainder of the work.

Figure 1a shows the results for all artists in OLGA. We can see that when using features from MSD, the contrastive method outperforms the baselines and the original embeddings in all the metrics. The contrastive method always gives a better Gini compared to the other methods –which means that the distribution of retrieved artists is more uniform– while outperforming the other models in nDCG. [9]

[8] we compute CD_{intra} as the mean cosine distance between multimodal embeddings of an artist to their centroid in the multimodal space of learned contrastive model. CD_{inter} is computed as the mean distance between the centroids of target artist and of all the other artists.

[9] OWN-OGLA is omitted since we observe a similar behaviour.



(a) MSD-OLGA (b) MSD-FMC (c) OWN-FMC

Figure 1: Performance comparison between contrastive and other methods. Training with MSD (a and b) or with OWN (c), Evaluation on OLGA (a) or on FMC (b and c).

	OLGA		FMC	
	nDCG@200	Gini	nDCG@200	Gini
C^{A+CF+T}	**0.2387**	0.2264	**0.3560**	0.1666
C^{A+T}	0.2282	0.2035	0.3407	0.1559
C^{A+CF}	0.1381	0.3425	0.2319	0.1873
C^{CF+T}	0.1781	0.3467	0.3082	0.1917
C^{A}	0.2338	**0.1857**	0.3471	**0.1353**
C^{T}	0.1232	0.4939	0.2554	0.1745
C^{CF}	0.1381	0.3425	0.2319	0.1873

Table 1: Evaluation of the contrastive method trained with MSD data using all combinations of modalities for OLGA dataset and FMC subset.

3.1.2 Performance with complete modality information

When we look at the results with Full Modality Coverage (Figures 1b and 1c), the contrastive method outperforms the baselines and the pre-trained models in all the metrics both when trained with MSD data or with OWN data.

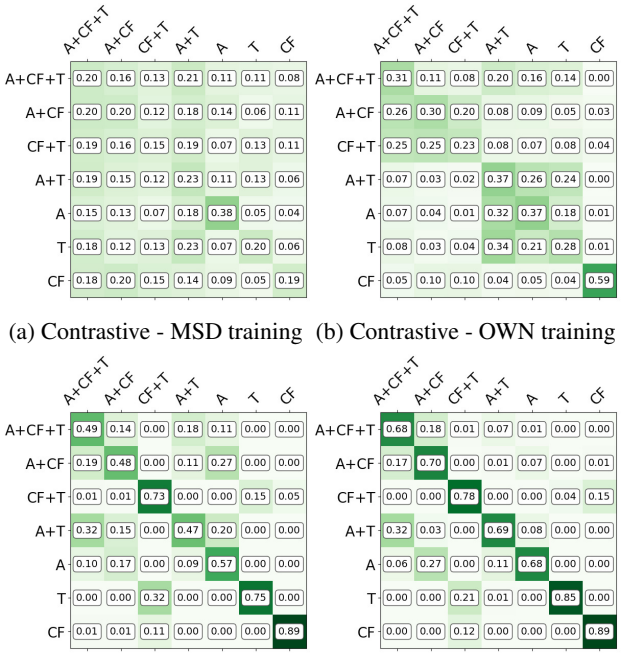
When looking at baseline performance between OLGA and FMC (Figures 1a and 1b), we can see that in the latter, baselines are relatively close to the best single-modality embeddings, but in the former (i.e. with incomplete modality information) their performance drops significantly lower than the best single-modality embeddings. This is something we do not observe with the contrastive method, which suggests that the baseline models are more limited in the capabilities of retrieving artists that miss some of the modalities from the query artist, while our contrastive method may be more robust to missing modality information. We investigate this further in Section 3.3.

3.2 Combining complementary modality information

If we focus only on the single-modality approaches, and MSD pre-training, Audio gives the best single-modality performance in both OLGA and FMC (Figure 1a and 1b). On the other hand, when pre-trained with OWN, CF is slightly better than Audio and Tag (Figure 1c). These results suggest that performance is highly dependent on the quality of the data used to pre-train the single-modality embeddings. Results from Figure 1b and 1c also sug-

	Audio	CF	Tag	Rand	PCA	C_{MSD}	C_{OWN}
Entropy	0.76	0.79	0.79	0.73	0.96	**1.86**	1.59

Table 2: Entropy of each model for Modality Groups. Higher values indicate better distributed retrieved artists.



(a) Contrastive - MSD training (b) Contrastive - OWN training

(c) PCA - MSD training (d) PCA - OWN training

Figure 2: Analysis of modality-group dependency ratio when restricting the information available for each group to one, two, or three modalities. Rows indicate the groups used to make the queries and the columns are the groups of retrieved artists. Darker green indicates a higher concentration of the retrieved artists in that cell. The color scale is normalized across all figures. Groups of artists are randomized, so an ideal situation is a homogeneous color in the full matrix.

gest that, whichever single-modality embedding is best, our contrastive method is able to successfully build on top of it and still gain in performance by combining complementary information from other embeddings.

3.3 Robustness to missing modality data

In this subsection, we further analyze how the contrastive method would be able to retrieve artists depending on the available information for the query artists and the candidates for retrieval. In Figure 2, we can see how artists are retrieved from each of the Modality Groups when only considering the top 5 results for each query artist. Typically we see that with the contrastive method, the same group used for query comprises between 15-38% of the retrieved artists. We see however an exception for the CF group which obtains a larger portion of the retrieved artists (59%) when using OWN data to train the models.

When we do a similar comparison for the PCA baseline method, we see in Figure 2 that there are higher percentages in the diagonal of the matrix. This indicates that most of the retrieved artists are concentrated in the same modality group used to make the query. Therefore, these results highlight the difficulty for the PCA baseline method to retrieve artists beyond the query artist's modality.

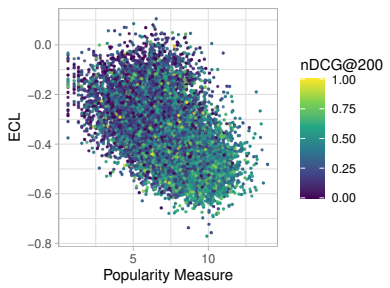


Figure 3: Scatter plot of artists based on the popularity proxy measure and the ECL. Each point represents an artist, where the color brightness represents the per-artist retrieval performance (nDCG@200). It is computed on the FMC subset with MSD data.

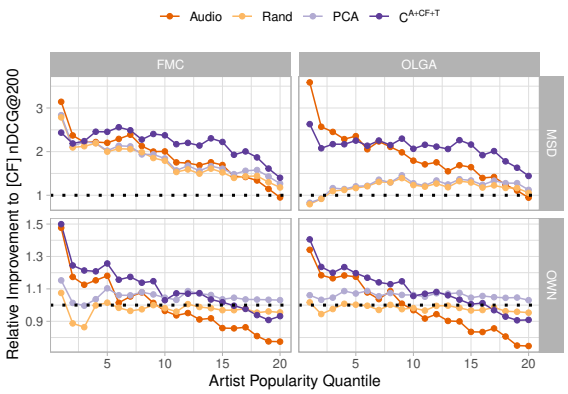


Figure 4: Relative retrieval improvement against CF modality. The x axis represents the grouped popularity quantile in 20 levels, meaning the first group includes artists whose popularity is under 5% percentile, while the top 5% popular artists belongs to the last group. The y axis is proportional improvement of nDCG@200 compared to the CF embedding model. The dotted horizontal line indicates the retrieval performance of CF modality. FMC and OLGA are evaluation datasets. MSD and OWN are training conditions.

In Table 2 we compare the entropy of each model for the Modality Groups. A higher entropy indicates that retrieved artists are better distributed across the different modality groups, i.e. that retrieval is less biased by the query modality –or more robust to partial modality data in the query. We can see that the contrastive model is more robust to missing modality data than the single-modality embeddings and the baseline approaches to combining modalities. This is true when trained with MSD or with OWN.

3.4 Effect of Popularity

Artist popularity may be a deterministic factor in artist retrieval, both for training and evaluation. Intuitively, we likely have more data about popular artists, which implies more multimodal data is available for training. At the same time, the scale of evaluation metric themselves can be inflated as more popular artists would have more ground truths (annotated as 'similar artists'). To confirm this, we compute a proxy measure for the artist popularity (POP) as $\text{POP}(\text{artist}) = \log(\#\text{listen} + 1)$, [10] and then further compare it to other training and evaluation measures.

Firstly, we compare POP with ECL and the retrieval performance. Figure 3 shows that there is correlation among POP, ECL, and nDCG. In particular, ECL has a negative correlation with nDCG. This is a desirable outcome as a model that minimizes the contrastive loss recommends "similar" artists even though such a model is not being explicitly shown artist-relatedness ground truth during training. Meanwhile, POP also correlates with nDCG, which demonstrates the confounding effect of popularity to the task itself.

Further, we investigate how multimodal models interact with artists with different popularities. One of the benefits of employing multiple modalities is the potential mitigation of the information void for "cold-start" artists from their music audio data. For MIR applications, audio is likely accessible even when some of the other modalities are not readily available. For instance, the CF modality is not available before artists' songs are consumed by the listeners. To confirm whether the audio and further multimodal embedding models would benefit less popular artists via multimodality, we divided the artists in 20 groups by popularity quantiles. For each group, we further compute the relative improvement of retrieval performance (nDCG) compared to the CF single modality model.

Figure 4 suggests that the original audio embedding achieves better performance for the less popular artists in all training and evaluation conditions. The contrastive model shows improvements for the majority of the groups compared to the audio, while it may have smaller or no improvement over audio in the least popular group for the MSD dataset. In the OWN dataset, a similar trend is observed where the contrastive model shows a small decline for the most popular groups compared to the original CF embeddings. The two baseline models indicate relatively flat results except in the case of the MSD-FMC subset, which implies that their prediction may be more reliant on the CF modality. For the MSD-FMC subset, both baselines follow similar trends to the audio and contrastive model.

3.5 Multimodal Embedding Space Analysis

We conduct a correlation study of multiple measures where, for each artist, we compute clustering measures and other key indicators such as contrastive loss ECL, retrieval performance (nDCG@200), and finally the popularity measure. In this way, we expect to obtain a better understanding of what contrastive learning achieves in terms of clustering of embeddings, and how they are connected to retrieval performance and popularity.

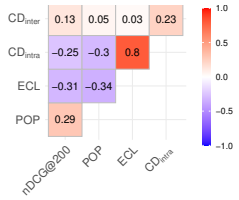


Figure 5: Correlation (Kendall's τ) among variables of interest. Each cell indicates value of τ between two associated variables. POP denotes the popularity measure.

The result of the correlation study can be found in Figure 5. We see that the ECL is highly correlated to CD_{intra}, while almost independent to CD_{inter}. Notably, in terms of magnitude, all other measures (ECL, CD_{intra}, and POP) are relatively more correlated to nDCG compared to CD_{inter}, and also correlated to each other. [11]

These relations suggest that our contrastive learning method aims at producing an artist embedding space where the diverse modalities of an artist occupy a coherent region, but not necessarily a region that is unique to the artist. CD_{inter} shows lower correlation with most of the other measures, which confirms its relatively small connection to the contrastive learning and the artist retrieval downstream task. We hypothesize that this is because the maximization of CD_{inter} is constrained by the artist similarity inherent in the multimodal information and ultimately preserved. This is desirable if the ultimate goal is a representation that can measure artist similarity.

4. CONCLUSION AND FUTURE WORK

In this work, we propose a method based on contrastive learning to combine multiple artist modalities into a single representation. In an artist similarity task, we show our method yields clear improvements over other methods in terms of retrieval accuracy and coverage, and successfully combines complementary information from diverse modalities. In particular, we investigate retrieval bias towards the query's modality. Although our method exhibits a slight bias towards retrieving artists with similar modality to the query, we show it handles cross-modal retrieval better than other methods. Future work may be dedicated to further mitigate this bias. Additionally, we show that our method is particularly beneficial for less popular artists.

Our method appears to generate an artist representation space with high local coherence for intra-artist modalities, but at the cost of inter-artist separation. Depending on the final application, this is a property that could perhaps be managed by iterating on the contrastive learning method, for instance, by adapting the loss function or by adapting the size of the training sample batch as suggested in [28].

[10] #listen denotes the total listening count of the artist, computed from the MSD-Echonest Profile dataset.

[11] We focus on the magnitude, as the goal of this study is to investigate the degree to which some of the key indicators are associated with clustering quality measures in absolute manner

5. ACKNOWLEDGEMENT

We would like to express special thanks to Matt McCallum for the help collecting audio features and Sam Sandberg for his valuable comments.

6. REFERENCES

[1] D. P. Ellis, B. Whitman, A. Berenzweig, and S. Lawrence, "The quest for ground truth in musical artist similarity," in *ISMIR*, 2002.

[2] T. Pohle, D. Schnitzer, M. Schedl, P. Knees, and G. Widmer, "On rhythm and general music similarity." in *ISMIR*, 2009, pp. 525–530.

[3] M. Schedl, D. Hauger, and J. Urbano, "Harvesting microblogs for contextual music similarity estimation: a co-occurrence-based framework," *Multimedia Systems*, vol. 20, pp. 693–705, 2014.

[4] S. Oramas, M. Sordo, L. Espinosa-Anke, and X. Serra, "A semantic-based approach for artist similarity," in *ISMIR*, 2015.

[5] F. Korzeniowski, S. Oramas, and F. Gouyon, "Artist similarity for everyone: A graph neural network approach," *Transactions of the International Society for Music Information Retrieval*, vol. 5, no. 1, 2022.

[6] M. C. McCallum, F. Korzeniowski, S. Oramas, F. Gouyon, and A. F. Ehmann, "Supervised and unsupervised learning of audio representations for music understanding," in *ISMIR*, 2022.

[7] P. Alonso-Jiménez, D. Bogdanov, J. Pons, and X. Serra, "Tensorflow audio models in Essentia," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 266–270.

[8] P. Alonso-Jiménez, X. Serra, and D. Bogdanov, "Music representation learning based on editorial metadata from Discogs," in *ISMIR*, 2022.

[9] S. Dieleman, P. Brakel, and B. Schrauwen, "Audio-based music classification with a pretrained convolutional network," in *ISMIR*, 2011, pp. 669–674.

[10] 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. 2018; 1 (1): 4-21.*, 2018.

[11] 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)*, 2021, pp. 591–595.

[12] A. Ferraro, "Music cold-start and long-tail recommendation: Bias in deep representations," in *Proceedings of the 13th ACM Conference on Recommender Systems*, 2019, p. 586–590. [Online]. Available: https://doi.org/10.1145/3298689.3347052

[13] S. Oramas, O. Nieto, F. Barbieri, and X. Serra, "Multi-label music genre classification from audio, text, and images using deep features," in *ISMIR*, 2017.

[14] A. Van den Oord, S. Dieleman, and B. Schrauwen, "Deep content-based music recommendation," *Advances in neural information processing systems*, vol. 26, 2013.

[15] 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 *ISMIR*, 2022.

[16] I. Manco, E. Benetos, E. Quinton, and G. Fazekas, "Contrastive audio-language learning for music," in *ISMIR*, 2022.

[17] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[18] A. Ferraro, X. Favory, K. Drossos, Y. Kim, and D. Bogdanov, "Enriched music representations with multiple cross-modal contrastive learning," *IEEE Signal Processing Letters*, vol. 28, pp. 733–737, 2021.

[19] T. Bertin-Mahieux, D. P. Ellis, B. Whitman, and P. Lamere, "The million song dataset," in *ISMIR*, 2011.

[20] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative filtering for implicit feedback datasets," in *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008)*, 2008, pp. 263–272.

[21] T. Head, MechCoder, G. Louppe, I. Shcherbatyi, fcharras, Z. Vinícius, cmmalone, C. Schröder, nel215, N. Campos, T. Young, S. Cereda, T. Fan, rene rex, K. K. Shi, J. Schwabedal, carlosdanielcsantos, Hvass-Labs, M. Pak, SoManyUsernamesTaken, F. Callaway, L. Estève, L. Besson, M. Cherti, K. Pfannschmidt, F. Linzberger, C. Cauet, A. Gut, A. Mueller, and A. Fabisch, "scikit-optimize/scikit-optimize: v0.5.2," Mar. 2018. [Online]. Available: https://doi.org/10.5281/zenodo.1207017

[22] F. Korzeniowski, S. Oramas, and F. Gouyon, "Artist similarity with graph neural networks," in *ISMIR*, 2021.

[23] E. Bingham and H. Mannila, "Random projection in dimensionality reduction: applications to image and text data," in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, 2001, pp. 245–250.

[24] W. B. Johnson and J. Lindenstrauss, "Extensions of Lipschitz mappings into a Hilbert space," *Contemporary Mathematics*, vol. 26, pp. 189–206, 1984. [Online]. Available: https://doi.org/10.1090/conm/026/737400

[25] 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 E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[26] D. Valcarce, A. Bellogín, J. Parapar, and P. Castells, "Assessing ranking metrics in top-n recommendation," *Information Retrieval Journal*, vol. 23, pp. 411–448, 2020.

[27] B. Efron and R. Tibshirani, *An Introduction to the Bootstrap*. Springer, 1993. [Online]. Available: https://doi.org/10.1007/978-1-4899-4541-9

[28] C. Chen, J. Zhang, Y. Xu, L. Chen, J. Duan, Y. Chen, S. Tran, B. Zeng, and T. Chilimbi, "Why do we need large batchsizes in contrastive learning? a gradient-bias perspective," in *Advances in Neural Information Processing Systems*, vol. 35, 2022, pp. 33 860–33 875.

REPETITION-STRUCTURE INFERENCE WITH FORMAL PROTOTYPES

Christoph Finkensiep[1,2] **Matthieu Haeberle**[1] **Friedrich Eisenbrand**[1]
Markus Neuwirth[3] **Martin Rohrmeier**[1]
[1] École Polytechnique Fédérale de Lausanne, Switzerland
[2] University of Amsterdam, The Netherlands (corresponding author: c.finkensiep@uva.nl)
[3] Anton Bruckner Privatuniversität Linz, Austria

ABSTRACT

The concept of form in music encompasses a wide range of musical aspects, such as phrases and (hierarchical) segmentation, formal functions, cadences and voice-leading schemata, form templates, and repetition structure. In an effort towards a unified model of form, this paper proposes an integration of repetition structure (i.e., which segments of a piece occur several times) and formal templates (such as AABA). While repetition structure can be modeled using context-free grammars, most prior approaches allow for arbitrary grammar rules. Constraining the structure of the inferred rules to conform to a small set of templates (meta-rules) not only reduces the space of possible rules that need to be considered but also ensures that the resulting repetition grammar remains interpretable in the context of musical form. The resulting formalism can be extended to cases of varied repetition and thus constitutes a building block for a larger model of form.

1. INTRODUCTION

Repetition is one of the most central aspects of music [1] and constitutes a constant across almost all cultures, styles and genres. The repetition of material is one of the major compositional devices for the arrangement of parts in overarching musical form [2, 3, 4], be it a folksong, a minuet, a sonata, a jazz standard, or a pop song. In general, musical form could be characterized in terms of exhaustive segmentation, hierarchical grouping structure, rhythmic-hypermetrical structuring, the form functionality of segments [3], and repetition structure. For the purpose of this paper, three aspects of form are considered: a hierarchical organization [5], which is also reflected in hierarchical harmonic structure [6]; repetition of formal constituents, which is one of the most prominent and salient features of form perception in human music cognition [1]; and prototypes of formal organization (such as AABA) which can characterize classical forms [3] but are also common structures in pop, jazz, and folk songs.

© C. Finkensiep, M. Haeberle, F. Eisenbrand, M. Neuwirth, M. Rohrmeier. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** C. Finkensiep, M. Haeberle, F. Eisenbrand, M. Neuwirth, M. Rohrmeier, "Repetition-Structure Inference with Formal Prototypes", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

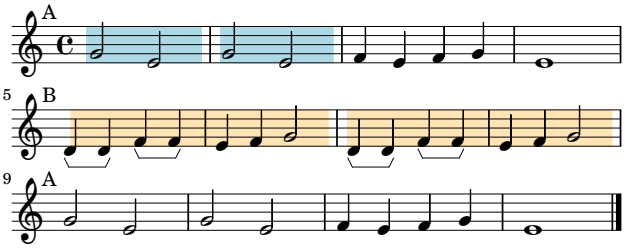


Figure 1: A German 19th-century folksong melody on the lyrics "Stille, stille, kein Geräusch gemacht" or "Bier her, Bier her, oder ich fall um."

The GTTM [5] defines *grouping structure* in terms of a tree of hierarchical containment relations that provides an exhaustive segmentation of the piece. GTTM's preference rules for grouping structure include Gestalt principles [7] as well as repetition. In addition to grouping structure, repetition structure is defined as a hierarchical grouping tree that captures (optimal) reuse of material (exact or in variation) in terms of groups of musical units and recursive groups of groups. Repetition structure provides a full grouping of a piece, however, it may potentially result in a different tree than what is obtained by a general formal analysis of a piece (see Figure 6). For human judgement of form in general, repetition is not the only factor, as features of (hyper-)metrical structure, form functions, or harmony may play a role as well (see also below in section 4.2). Accordingly, the objective of a computational model of repetition structure as an aspect of musical form may ultimately require to take such aspects into account as well.

Repetition structure plays a role within a single piece as well as over a corpus of pieces since abstract repetition patterns generalize over a whole dataset or style. The melody shown in Figure 1, for example, exhibits repetition of parts on several levels: On the highest level, the melody follows an ABA form, as the first four measures are literally repeated at the end (mm. 9-12). The B part (mm. 5-8) itself consists of a repetition of a two-measure phrase (yellow). Similarly, the first measure of the A part is repeated in the second measure (blue). Even on the level of individual notes, the direct repetition of a note is a prominent feature of mm. 5 and 7.

In the context of form, repetition structure refers to the re-occurrence of formal constituents (such as phrases and sections) that form a hierarchical segmentation structure,

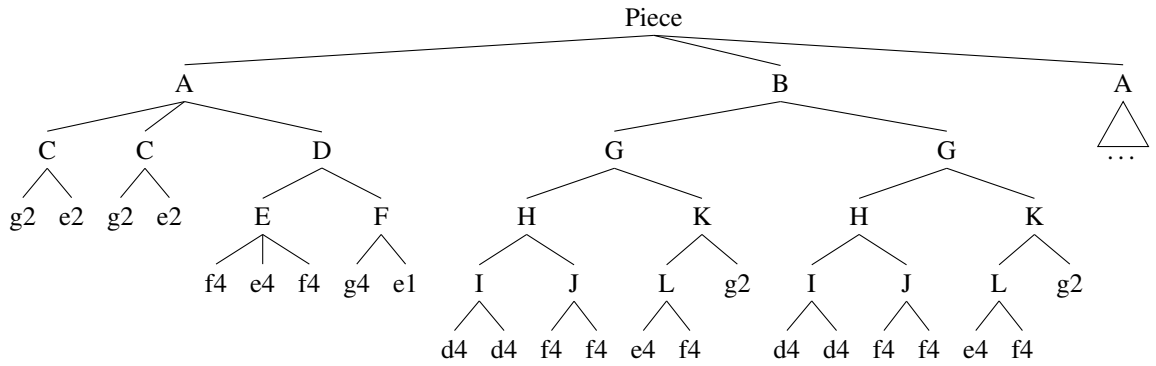


Figure 2: A possible repetition tree of the melody in Figure 1. The leaves of the tree encode pitch and duration of the melody notes. The second occurence of part A is identical to the first (not shown here).

as opposed to motivic or thematic material, for example. An example of such a segmentation structure for the example piece (Figure 1) is shown in Figure 2. Every formal segment of the piece corresponds to a subtree in the repetition tree. Note that all occurrences of the same segment share the same label and have exactly the same subtree structure. For this reason, the repetition structure shown in the tree can be more compactly described as a restricted context-free grammar (CFG) which has one non-terminal symbol for each segment (with terminal symbols for the notes of the piece) and exactly one rule for each symbol, encoding the decomposition of the corresponding segment. This relationship between the repetition structure of a piece and a compact representation of the piece as a CFG has been utilized for compression-based pattern discovery algorithms such as SEQUITUR [8].

Another aspect of the repetition tree shown in Figure 2 is that its rules use a limited set of formal prototypes, such as $\alpha\beta\alpha$ (e.g., Piece $\longrightarrow$ ABA), $\alpha\alpha\beta$ (A $\longrightarrow$ CCD), or $\alpha\alpha$ (B $\longrightarrow$ GG). In order to avoid confusion with the letters for specific form parts, we denote these form templates with greek letters, e.g., $\alpha\beta\alpha$, $\alpha\alpha\beta$, or $\alpha\alpha$. A concrete instance of a form template is denoted by applying the template to specific segments: $\mathrm{CCD} = \alpha\alpha\beta(\mathrm{C}, \mathrm{D})$. While the rules in a piece's repetition grammar are specific to a particular segment in that particular piece, the form templates establish a relation between different rules with the same shape, within the same piece or across different pieces. We therefore call them *meta rules*.

This paper is a contribution towards an integrated computational model of musical form, combining two important aspects of form: repetition and formal prototypes. The model characterizes the relationship between meta rules and hierarchical repetition structure and provides a *proof of concept* algorithm and evaluation for repetition structure inference based on *minimal description length* [9].

2. RELATED WORK

Identification of repetition structure is closely related to compression, as identification of redundant information is important to achieve shorter encodings. An early example of grammar-based compression is SEQUITUR, an algorithm that infers a (not globally optimal) grammar for a given sequence in linear time [8, 10]. For an overview of approximate grammar-based compression, see [11, 12]. Besides inference of segmentation structure, grammar-based compression algorithms have been used for tasks such as error detection and tune classification [12, 13]. The principle of minimum description length has also been used outside of grammar-based approaches, e.g., in combination with hidden Markov models [14]. The approach presented in this paper differs from previous smallest-grammar approaches in two ways: the shape of the grammar rules is not arbitrary but constrained to a set of formal prototypes, and this constrained model is evaluated by inferring the global optimum instead of an approximation, which is generally NP-hard and thus only feasible for short sequences.

The segmentation structure of a piece can also be inferred based on criteria other than repetition. The GTTM [5] defines grouping structure based on a set of well-formedness and preference rules for recursively combining events into larger segments. In the MIR community, the analysis of musical form is known as *music structure analysis* (MSA) [15, 16, 17, 18, 19, 20, 21, 22]. MSA comes in a variety of tasks, involving boundary detection, (hierarchical) segmentation, the identification of segment labels and relations, and combinations of these tasks. While MSA uses a wide spectrum of supervised and unsupervised methods, from matrix factorization to deep learning, the definition of musical form in this context is usually given implicitly in the form of a dataset (e.g., [21, 22]) on which the model may be trained, and on which it is evaluated. The present paper, in contrast, presents a theoretical contribution towards an explicit definition of musical form, and the resulting model is not intended as a solution to a computational problem, such as performing a general segmentation and labeling task. As a consequence, our evaluation focuses on exploring the characteristic properties of the model.

3. METHODS AND DATA

3.1 Problem Description

A specific repetition structure for a given piece can be characterized through a piece-specific context-free gram-

$m_1 : \alpha\alpha$	$m_2 : \alpha\beta$
$m_3 : \alpha\alpha\alpha$	$m_4 : \alpha\beta\alpha$
$m_5 : \alpha\alpha\beta$	$m_6 : \alpha\beta\beta$
$m_7 : \alpha\alpha\beta\alpha$	$m_8 : \alpha\beta\beta\alpha$

Table 1: The set of meta rules used in this paper.

mar that generates exactly one string — the piece. We call such a grammar a *local grammar* for the piece. It consists of:

- a set of terminal symbols T, corresponding to the unique atomic segments of the piece; [1]
- a set of non-terminal symbols N, corresponding to the unique composite segments of the piece;
- a starting symbol P that stands for the full piece;
- a set of production rules R.

Since each non-terminal symbol stands for a specific segment, R contains exactly one rule for each non-terminal symbol, signifying the decomposition of the segment and enforcing that all occurrences of the segment are decomposed identically. As a consequence, the rules are not allowed to be (mutually) recursive since a segment cannot contain itself as a proper subsegment. [2]

In order to establish a relation between local repetition grammars and general formal prototypes, the right-hand side (RHS) of each rule must be an instance of a *meta rule*. Meta rules are generally of the shape $\{\alpha, \beta, \gamma, \ldots\}^+$ and are instantiated by creating a bijective mapping between letters and specific non-terminal symbols. For example, the meta rule $\alpha\alpha\beta\alpha$ encodes the formal prototype AABA and can be instantiated as

$$\alpha\alpha\beta\alpha(S,T) = \alpha\alpha\beta\alpha\{\alpha \mapsto S, \beta \mapsto T\} = SSTS \quad (1)$$

where $S \neq T$. Thus, a local grammar can express that a piece has an overarching AABA structure by using a rule

$$P \longrightarrow \alpha\alpha\beta\alpha(S,T) \quad (2)$$

that takes the starting symbol P to an instance of $\alpha\alpha\beta\alpha$ with $\alpha = S$ and $\beta = T$. The set of meta rules can be chosen freely to encode a set of typical formal prototypes. The meta rules used in the following experiments are shown in Table 1.

The goal of repetition structure inference is to find a local grammar for a given piece according to some optimality criterion, such as musical plausibility, probability, or description length (DL). In accordance with prior approaches that use repetition grammars, our proof-of-concept implementation searches for local grammars with minimum description length, defined in analogy to [13] by counting the symbols needed to encode the grammar:

$$DL(R) = \sum_{r \in R} 2 + |\text{params}(r)| \quad (3)$$

where params(r) denotes the parameters of the meta rule on the RHS of rule r. That is, for each rule we count one symbol for the meta rule, one symbol for each parameter of the meta rule, and one separator symbol [3] marking the end of the rule. For example, the rule in Equation 2 has a description length of 4: one meta-rule symbol ($\alpha\alpha\beta\alpha$ or m_7) [4], two parameters (S and T) and the separator. It is not necessary to encode the left-hand side (LHS) of a rule since there exists a canonical order of rules, starting with the rule for P and then listing the rules in the order in which their LHS symbols are introduced on the RHS of other rules.

3.2 Algorithm

The minimal grammar for a given piece is found in a two-stage process. First, a set of possible rules for each unique segment of the piece is computed. Second, a set of rules is selected from these candidates, ensuring that the resulting grammar is consistent and minimizing the cost of the selected rules.

Algorithm 1 Enumerating all rule candidates.

```
1:  function PARSE(input)
2:      subs ← uniqueSubsequences(input)
3:      chart ← {}
4:      for seq ∈ sortByLength(subs) do
5:          for s from 1 to |seq| − 1 do
6:              cs ← COMPLETE(seq[: s], seq[s + 1 :])
7:              chart[seq] ← cs
8:      return chart

9:  function COMPLETE(left, right)
10:     il ← chart[left]_incomplete
11:     ir ← chart[right]_incomplete
12:     bs ← binaryRules(left, right)
13:     is ← incompleteConstituents(left, right, il, ir)
14:     ns ← nAryRules(left, right, il, ir)
15:     return (complete = bs ∪ ns, incomplete = is)
```

The first stage (Algorithm 1) begins with collecting all unique subsegments of the piece. For each of these subsegments, all possible decompositions according to the meta rules are computed using dynamic programming, analogous to the CYK algorithm: The segment is split at every possible split point (l. 5), generating two subsegments left and right of the split point. For binary meta rules ($\alpha\alpha$ and $\alpha\beta$), an instance of the rule can be identified directly by comparing the subsegments (l. 12). Meta rules

[1] In the case of melodies, these atomic segments correspond to notes and rests, but they could also correspond to polyphonic events (slices), or previously annotated elementary phrases.

[2] A unary identity rule (e.g. $X \longrightarrow X$) is not permitted. Other unary rules are not possible because of the one-to-one correspondence between segments and grammar symbols.

[3] The separator is not strictly necessary since the length of the rule is known from the meta rule, but it is included here to stay as close as possible to [13].

[4] $\alpha\alpha\beta\alpha$ is counted as one symbol since the set of meta rules is assumed to be fixed and cannot be freely extended.

of higher arity are decomposed into binary parts. For example, the meta rule $\alpha\beta\alpha$ can be decomposed into an incomplete constituent $\alpha\beta^*$ and another α. When a segment $S = S_1S_2$ has a decomposition into $\alpha\beta(S_1, S_2)$, it additionally stores an $\alpha\beta^*(S_1, S_2)$ item (l. 13). At a later point, a larger segment $T = SS_3 = S_1S_2S_3$ retrieves the item $S \longrightarrow \alpha\beta^*(S_1, S_2)$, checks whether $S_3 = S_1$, and accordingly stores a rule $T \longrightarrow \alpha\beta\alpha(S_1, S_2)$. Similarly, all larger meta rules are constructed from incomplete constituents such as $\alpha\beta^*$ and $\alpha\alpha^*$ (l. 14).

Once the possible decompositions of each subsequence are known, the second stage of the algorithm converts the set of possible rules into an integer linear program (ILP) which then extracts a set of rules with minimal cost. Each subsequence of the input of lenght ≥ 2 corresponds to a potential non-terminal symbol, so one binary indicator variable s_{symb} for each symbol $symb$ encodes the inclusion of the symbol in the grammar. Similarly, the inclusion of each candidate rule is indicated by a binary variable s_{rule}. The optimization problem is then given by

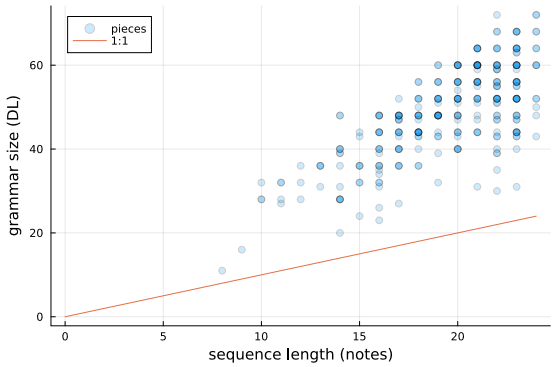
$$\begin{aligned} \min_{\substack{r\in\{0,1\}^{|rules|}\\ s\in\{0,1\}^{|symbols|}}} \quad & \sum_{rule\in rules} r_{rule}\cdot DL(rule) \\ \text{s.t.} \quad & s_{symb} = \sum_{\substack{rule\in rules\\ \text{LHS}(rule)=symb}} r_{rule} \\ & r_{rule} \leq \sum_{symb\in\text{RHS}(rule)} \frac{s_{symb}}{|\text{RHS}(rule)|} \\ & s_{start} = 1. \end{aligned} \tag{4}$$

Two constraints define the relationship between symbols and rules: each included symbol requires exactly one corresponding rule; and each included rule requires the symbols on its right-hand side. [5] A third constraint requires the presence of the starting symbol which corresponds to the full input sequence. The rules and symbols are then selected by minimizing the total cost of the included rules as defined in Equation 3.

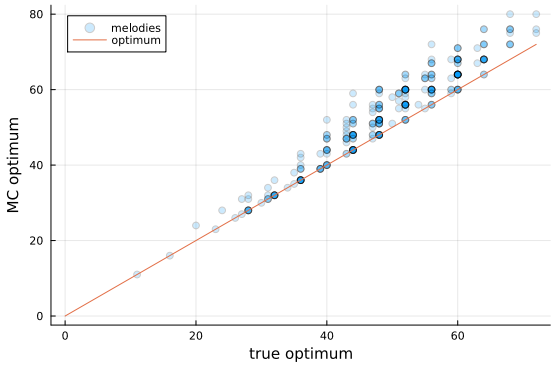
4. RESULTS AND DISCUSSION

4.1 Quantitative Evaluation on a Dataset

For evaluating the above approach, we infer the minimal grammars (under the meta rules from Table 1) for the 298 shortest melodies from the Essen folksong collection [23], with a length of 8 to 24 notes. The melodies are represented as sequences of notes (including rests), consisting of pitch (or a rest symbol) and duration. Other aspects, such as the position of a note in a measure, are not taken into account. The minimization algorithm is implemented in Julia and is available online. [6] For ILP optimization we use the JuMP framework [24] together with the Gurobi solver backend. [7]



(a) Grammar size vs. input length.



(b) Optimal grammar size vs. Monte-Carlo minimum.

Figure 3: Comparison of the description length of the minimal local grammars to (a) the input sequence length and (b) local grammars obtained through Monte-Carlo minimization.

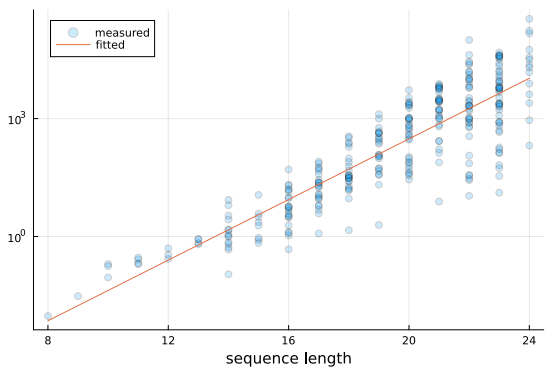
Since the local repetition grammar formalism is not designed to obtain optimal compression of the input sequence (but uses description length as a rather arbitrary proxy for the plausibility of a specific segmentation), we cannot expect very good compression rates. Indeed, when comparing the length of the input sequences to the total description length of the corresponding minimal grammar, the grammars are usually larger than the original piece (Figure 3a) with a average ratio of 2.47 (geometric mean). This indicates that restricting the grammars to a small set of meta rules is not sufficient to achieve an actual compression of the dataset, at least when only considering exact repetition.

Since finding the global minimum is expensive (see below), most grammar-based compression algorithms only attempt to approximate the global optimum [8, 12, 13]. We estimate the payoff of inferring the global optimum by comparing the optimal description lengths to approximate solutions obtained by a Monte-Carlo minimization process: Beginning with the start symbol, the rule for each required symbol is chosen randomly from the set of possible rules, and the corresponding RHS symbols are added to the list of required symbols. This process is repeated until all required symbols are covered. Out of 10,000 randomly sampled grammars for each piece, the smallest grammar is selected. The results are shown in Figure 3b. For the

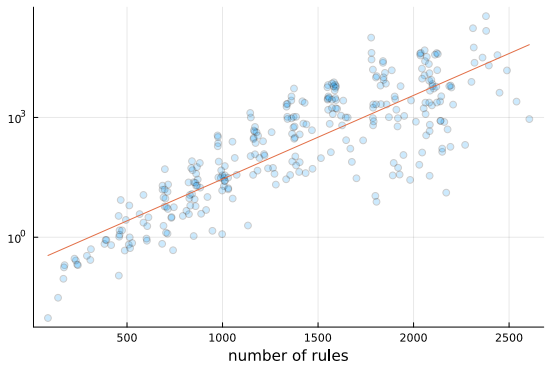
[5] The logical conjunction of the RHS symbols is expressed as a normalized sum instead of a product in order to maintain linear relationships between the variables in the program.

[6] https://github.com/DCMLab/form-repetition-ismir23

[7] Gurobi requires a license, which is provided freely for academic purposes. Alternatively, the JuMP framework supports using different solver backends.

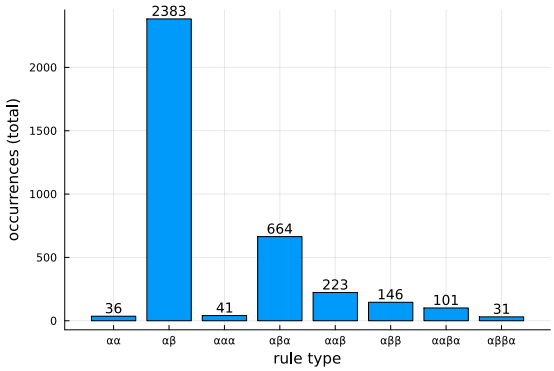


(a) Runtime relative to input length.

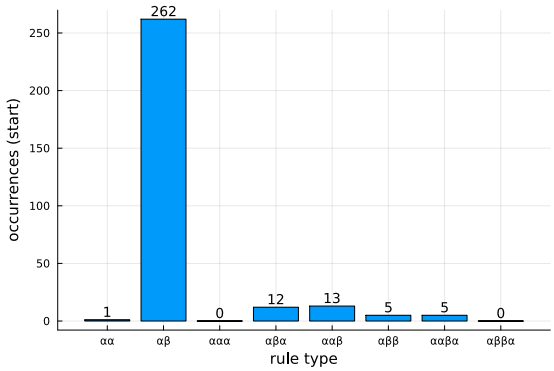


(b) Runtime relative to possible rules.

Figure 4: The measured runtime of the optimization step relative to (a) the length of the input sequence and (b) the number of possible rules for the sequence. Note that in both cases, the time axis is scaled logarithmically, so the fitted exponential curves appear as straight lines.



(a) Overall meta rule usage.



(b) Meta rules used at the top of the form tree.

Figure 5: The meta rules used in the inferred minimal grammars for the melodies in the dataset.

given dataset, the Monte-Carlo minimum is on average 1.08 times longer than the true grammar. In many cases, the Monte-Carlo process finds a true optimum, since the sample size of 10,000 is large enough to find an optimal solution by chance. However, with growing input size, the range of possible grammars grows exponentially, in the worst case.[8] So, while a Monte-Carlo estimate can be a useful approximation on short sequences, it cannot keep up with the size of the search space for longer sequences, unless the sample size is increased exponentially as well.

The runtime behavior of the optimization problem is shown in Figure 4. The problem of finding an unrestricted minimal CFG is known to be NP-hard [13]. The runtime for the restricted case relative to the input length is shown in Figure 4a with logarithmic scaling. Since the actual size of the optimization problem depends not only on the input length but also on the amount of redundancy within the sequence, Figure 4b shows the runtime relative to the number of possible rules obtained in the first stage of the algorithm. In both cases, the runtime grows approximately exponentially with the number of rules. This is supported by an exponential regression in both figures, fit as a linear function in logarithmic space which minimizes the squared *ratio* between measured and predicted runtime instead of the squared difference.

The distribution of meta rules in the inferred grammars is shown in Figure 5a. By far the most common rule type is $\alpha\beta$, which is not surprising since it is the only rule type that does not require any form of repetition. The rule type $\alpha\alpha$ is used very infrequently, which may seem surprising due to its simplicity. However, all other rules (except for $\alpha\beta\beta\alpha$ and $\alpha\alpha\alpha$ which are similarly rare) have one part that does not need to be repeated and are thus applicable to a wider range of situations. The distribution of starting-rule types is shown in Figure 5b. These rule types correspond to the overarching form of the melody in terms of exact repetition. The even stronger prevalence of $\alpha\beta$ in this case indicates that there is very little exact repetition on the highest form level in the given dataset of melodies, which might be biased due to the focus on short melodies. On the other hand, this lack of repetition indicates that a model of formal segmentation cannot be exclusively based on repetition but needs to take into account at least the possibility of varied repetition, as well as other markers of form such as cadences and meter.

4.2 Qualitative Evaluation on an Example Melody

Table 2 displays the minimal grammar for the example piece in Figure 1. Compared to the overall distribution of meta rules, the grammar uses many repeating rules, which reveals that the piece features an unusual amount of inter-

[8] The number of subsequences grows quadratically, and the number of possible grammars is a product over all substrings.

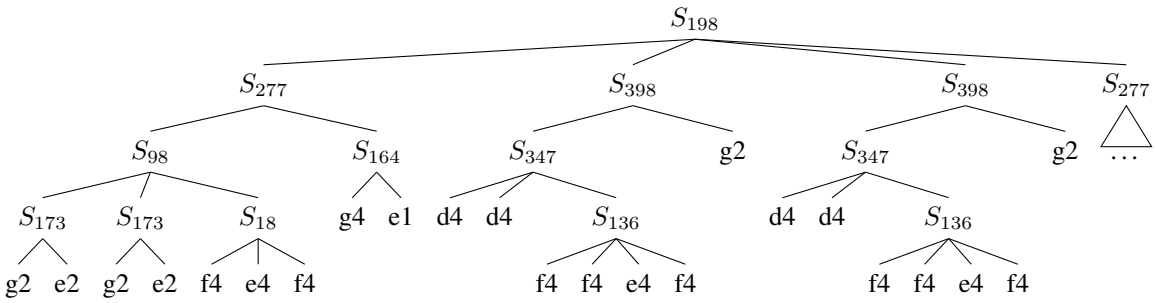


Figure 6: The minimal tree for the example piece in Figure 1.

rule		meta rule	cost
r_1	$S_{198} \longrightarrow S_{277}\ S_{398}$	$\alpha\beta\beta\alpha(S_{277}, S_{398})$	4
r_2	$S_{277} \longrightarrow S_{98}\ S_{164}$	$\alpha\beta(S_{98}, S_{164})$	4
r_3	$S_{398} \longrightarrow S_{347}\ g2$	$\alpha\beta(S_{347}, g2)$	4
r_4	$S_{98} \longrightarrow S_{173}\ S_{18}$	$\alpha\alpha\beta(S_{173}, S_{18})$	4
r_5	$S_{164} \longrightarrow g4\ e1$	$\alpha\beta(g4, e1)$	4
r_6	$S_{347} \longrightarrow d4\ S_{136}$	$\alpha\alpha\beta(d4, S_{136})$	4
r_7	$S_{173} \longrightarrow g2\ e2$	$\alpha\beta(g2, e2)$	4
r_8	$S_{18} \longrightarrow f4\ e4$	$\alpha\beta\alpha(f4, e4)$	4
r_9	$S_{136} \longrightarrow f4\ e4$	$\alpha\alpha\beta\alpha(f4, e4)$	4

Table 2: The minimal grammar for the example piece in Figure 1.

nal repetition. As the derivation tree in Figure 6 shows, the optimal solution found by the algorithm captures many aspects of the human intuition. Similar to the hand-annotated segmentation in Figure 2, the minimal tree captures the overarching repetition of mm. 1-4 and mm. 9-12 as well as mm. 5-6 and mm. 7-8. However, whereas the human intuition groups the single note repetitions together and splits non-repeating segments according to bar units (mm. 5, 7), the algorithm finds that other groupings provide an even more economic description length in terms of rule usage, which leads to a somewhat counter-intuitive dangling half note g at the end of the phrase. This illustrates that human decisions in terms of repetition structure do not purely optimize repetition, but that they take rhythmic-metric boundaries into account. Therefore, the objective of a model of repetition structure that captures or comes close to the human intuition needs to be further developed to also incorporate such features. A candidate model may be the hierarchical model of rhythmic structure as a formal grammar [25].

5. CONCLUSION

In this paper we have presented a computational model of musical repetition structure as an aspect of musical form. Since repetition structure is an aspect of human music cognition, the overarching objective of our approach is to approach human listening. The model captures repetition structure with a special form of context-free grammar, in which the rewrite of each category is only defined once such that it captures a unique repeating fragment of a given piece. A set of meta-rules defines the generic types of repetition patterns that could occur within a piece. The model is very generic and can also be applied to more complex textures as well as music of all styles and cultures, as long as a representation as a sequence of symbols is meaningful.

Inferring the optimal grammar with respect to a suitable objective criterion (such as description length) is able to effectively capture the repetition structure in a piece. Other objective criteria (e.g., prior probabilities of meta rules) can be used in a similar way since the algorithm does not depend on a fixed cost function. On the other hand, maximizing the redundancy that is captured by a segmentation does not ensure that the segmentation is a good analysis of the form of a piece. For one, not all repetition and reuse of material is exact, which is evident from the low proportion of repeating meta rules used in the example dataset. Simple forms of varied repetition could be integrated in our model relatively easily: given a suitable measure of similarity, not only identical segments are grouped together but also sufficiently similar segments. More sophisticated versions of this model could capture how variations are produced through the generative process of the grammar (e.g., by making different decisions in different subtrees), or how only certain aspects of a segment are repeated while others change (e.g., using the same rhythm with a different melodic contour). Furthermore, even when all repetitions are exact (as in the example piece), capturing repetition is not the only criterion for grouping tokens into formal segments, as other criteria such as cadences, rhythm and meter, formal function, or harmonic and contrapuntal schemata interact with grouping as well.

The runtime complexity of finding the smallest grammar for a given piece is generally exponential. For sufficiently short pieces, exact inference can be approximated probabilistically, but there is no guarantee that the resulting suboptimal grammars resemble the true optimum. For larger inputs, the search space grows exponentially, so naive Monte-Carlo approximation can become arbitrarily bad. This indicates that further research is required to find plausible estimates of formal structure, integrating the technical aspect of optimization with the musical problem of defining what constitutes a plausible analysis.

6. ACKNOWLEDGEMENTS

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme under grant agreement No 760081 – PMSB. This research was supported by the Swiss National Science Foundation within the project "Distant Listening – The Development of Harmony over Three Centuries (1700–2000)" (Grant no. 182811). The authors thank Mr. Claude Latour for generously supporting this research.

7. REFERENCES

[1] E. H. Margulis. *On Repeat: How Music Plays the Mind*. Oxford, New York: Oxford University Press, Feb. 6, 2014. 224 pp.

[2] F. Diergarten and M. Neuwirth. *Formenlehre. Ein Lese- Und Arbeitsbuch Zur Instrumentalmusik Des 18. Und 19. Jahrhunderts*. 2nd ed. Laaber-Verlag, 2020.

[3] W. E. Caplin. *Classical Form: A Theory of Formal Functions for the Instrumental Music of Haydn, Mozart, and Beethoven*. Oxford University Press, May 14, 1998. 320 pp.

[4] Y. Greenberg. *How Sonata Forms: A Bottom-Up Approach to Musical Form*. Oxford Studies in Music Theory. Oxford, New York: Oxford University Press, June 10, 2022. 264 pp.

[5] F. Lerdahl and R. Jackendoff. *A Generative Theory of Tonal Music*. MIT press, 1983.

[6] M. Rohrmeier. "The Syntax of Jazz Harmony: Diatonic Tonality, Phrase Structure, and Form". In: *Music Theory and Analysis (MTA)* 7.1 (Apr. 30, 2020), pp. 1–63. DOI: 10.11116/MTA.7.1.1.

[7] M. Wertheimer. "Laws of Organization in Perceptual Forms". In: *A source book of Gestalt Psychology* 1 (1923).

[8] C. G. Nevill-Manning and I. H. Witten. "Identifying Hierarchical Structure in Sequences: A Linear-Time Algorithm". In: *Journal of Artificial Intelligence Research* 7 (Sept. 1, 1997), pp. 67–82. DOI: 10.1613/jair.374.

[9] D. J. C. MacKay. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, Sept. 25, 2003. 694 pp.

[10] C. G. Nevill-Manning and I. H. Witten. "Compression and Explanation Using Hierarchical Grammars". In: *The Computer Journal* 40 (2_and_3 Jan. 1997), pp. 103–116. DOI: 10.1093/comjnl/40.2_and_3.103.

[11] E. Lehman and A. Shelat. "Approximation Algorithms for Grammar-Based Compression". In: *Proceedings of the Thirteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. Society for Industrial and Applied Mathematics., 2002, pp. 205–212.

[12] K. A. Sidorov, A. Jones, and A. D. Marshall. "Music Analysis as a Smallest Grammar Problem." In: *ISMIR*. 2014, pp. 301–306.

[13] D. Humphreys, K. Sidorov, A. Jones, and D. Marshall. "An Investigation of Music Analysis by the Application of Grammar-Based Compressors". In: *Journal of New Music Research* 50.4 (Aug. 8, 2021), pp. 312–341. DOI: 10.1080/09298215.2021.1978505.

[14] P. Mavromatis. "Minimum Description Length Modelling of Musical Structure". In: *Journal of Mathematics and Music* 3.3 (Nov. 1, 2009), pp. 117–136. DOI: 10.1080/17459730903313122.

[15] M. Buisson, B. Mcfee, S. Essid, and H.-C. Crayencour. "Learning Multi-Level Representations for Hierarchical Music Structure Analysis". In: International Society for Music Information Retrieval (ISMIR). Dec. 4, 2022.

[16] B. McFee. "Audio-Based Music Structure Analysis: Current Trends, Open Challenges, and Applications". In: 3.1 (1 Dec. 11, 2020), pp. 246–263. DOI: 10.5334/tismir.54.

[17] B. McFee and D. Ellis. "Analyzing Song Structure with Spectral Clustering." In: *ISMIR*. Citeseer, 2014, pp. 405–410.

[18] F. Kaiser and T. Sikora. "Music Structure Discovery in Popular Music Using Non-negative Matrix Factorization." In: *ISMIR*. 2010, pp. 429–434.

[19] M. Levy and M. Sandler. "Structural Segmentation of Musical Audio by Constrained Clustering". In: *IEEE Transactions on Audio, Speech, and Language Processing* 16.2 (Feb. 2008), pp. 318–326. DOI: 10.1109/TASL.2007.910781.

[20] M. C. McCallum. "Unsupervised Learning of Deep Features for Music Segmentation". In: *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). May 2019, pp. 346–350. DOI: 10.1109/ICASSP.2019.8683407.

[21] 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*. Vol. 11. Miami, FL, 2011, pp. 555–560.

[22] C.-i. Wang, G. J. Mysore, and S. Dubnov. "Re-Visiting the Music Segmentation Problem with Crowdsourcing." In: *ISMIR*. 2017, pp. 738–744.

[23] H. Schaffrath. "The Essen folksong collection in the Humdrum Kern Format (D. Huron, Ed.)" In: *Menlo Park, CA: Center for Computer Assisted Research in the Humanities* (1995).

[24] M. Lubin, O. Dowson, J. D. Garcia, J. Huchette, B. Legat, and J. P. Vielma. "JuMP 1.0: Recent improvements to a modeling language for mathematical optimization". In: *Mathematical Programming Computation* (2023). In press.

[25] M. Rohrmeier. "Towards a Formalization of Musical Rhythm". In: *Proceedings of the 21th International Society for Music Information Retrieval Conference, ISMIR 2020, Montreal, Canada, October 11-16, 2020*. Ed. by J. Cumming, J. H. Lee, B. McFee, M. Schedl, J. Devaney, C. McKay, E. Zangerle, and T. de Reuse. 2020, pp. 621–629.

ALGORITHMIC HARMONIZATION OF TONAL MELODIES USING WEIGHTED PITCH CONTEXT VECTORS

Peter van Kranenburg
Meertens Institute, Utrecht University
peter.van.kranenburg@meertens.knaw.nl

Eoin Kearns
Meertens Institute
eoin.kearns@meertens.knaw.nl

ABSTRACT

Most melodies from the Western common practice period have a harmonic background, i.e., a succession of chords that fit the melody. In this paper we provide a novel approach to infer this harmonic background from the score notation of a melody. We first construct a pitch context vector for each note in the melody. This vector summarises the pitches that are in the preceding and following contexts of the note. Next, we use these pitch context vectors to generate a list of candidate chords for each note. The candidate chords fit the pitch context of a given note each with a computed strength. Finally, we find an optimal path through the chord candidates, employing a score function for the fitness of a given candidate chord. The algorithm chooses one chord for each note, optimizing the total score. A set of heuristics is incorporated in the score function. The system is heavily parameterised, extremely flexible, and does not need training. This creates a framework to experiment with harmonization of melodies. The output is evaluated by an expert survey, which yields convincing and positive results.

1. INTRODUCTION

One of the essential aspects of Western folk music is that it is in oral circulation among practitioners regardless of formal music training. As such, the transmitted music is expected to conform to melodic patterns which belong to Western music traditions. This is most tangible in the perception of rules of tonality, including the perception of stable scale tones, modes, and key centres [1]. These factors dictate the implied harmonic movement within the melody. Detecting this implied harmony is an integral part of the accompaniment of folk music. With this knowledge, it is possible to create musically meaningful harmonic progressions, using symbolic chord representations to accompany a melody.

In this paper, our aim is to explicitly design a model of how to generate a sequence of accompanying chords for a given melody, such that e.g., a guitarist could play along. Most of the recent work on this task involves machine learning in which a model is trained on a set of examples. In contrast, an essential aspect of our approach is to explicitly incorporate musical expert knowledge into the model. Our model is heavily parameterized. This has the advantage of allowing the user to have full control over the process. A disadvantage could be that the resulting model lacks the flexibility to handle various situations, which often is a reason to train a neural network instead of handcrafting a model. Our results show, however, that our current model is capable of generating convincing chord sequences for a given melody.

The model can be employed in a wide range of applications. It allows a musician to quickly obtain a suitable accompaniment for a given melody. This can be accomplished by using the default parameter settings that are established in this paper, but the model also allows to tune the parameters to get a certain desired effect. In Section 5 of this paper we provide an example in which the number of generated chords greatly varies, while each generated chord sequence is acceptable to accompany the melody. Thus, the generated harmony can be adjusted to various levels of mastering an instrument.

From a music theory perspective, our model can be considered an experimental framework to explore general principles of harmonization. In this approach, the model is used to better understand these principles. It is extremely instructive to add a heuristic to the model, or to adjust a parameter, and to examine the cases in which this leads to strong chord sequences, but even more so to examine the cases that are not acceptable. These are conditions under which the general rule apparently fails. In the current paper, we do not elaborate on this use of our model, but it is an important affordance that we do not want to be left unmentioned.

We also can imagine the system being used in an artistic way, rather than to just generate an accompaniment for practical use. In the current implementation, we incorporate well-established principles of harmonization, but it is very well possible to include other heuristics that generate chord sequences that, although not adhering to the general principles of Western tonality, could be considered an artistic contribution, or an inspiration for a new composition.

Finally, we mention the possible educational use of the model. By exploring the generated chord sequences, students can get ideas to improve or enrich their own compositions or improvisation.

© P. van Kranenburg and E. Kearns. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** P. van Kranenburg and E. Kearns, "Algorithmic Harmonization of Tonal Melodies using Weighted Pitch Context Vectors", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

2. RELATED WORK

Multiple approaches have been taken for the task of automatic harmonization [2]. Early applications of formal grammars and rule-based algorithms for automatic harmonization [3, 4] mainly sought to compose chorales in the style of Bach. This was achieved by harmonizing the soprano melody line using a set of rules to ascertain harmonic choices. These rules and heuristics are informed by observation of chorales, and enhanced by rules found in treatises. The resulting systems output successful harmonizations of existing melodies, as well as new compositions.

Context free grammars have found use for this task. Koops [5] adopts this approach in his HarmTrace and FHarm models to derive the harmonic function of a chord in its tonal context according to a set of predefined rules.

Temperley [6] proposed a rule-based algorithm to harmonize a melody by dividing the piece temporally into segments (chord spans). All possible roots are then assigned a score according to a set of four rules. The model prefers root relations which best conform to the circle of fifths. The model also predominantly chooses chord spans which begin on the metrical downbeat, and identifies and prefers ornamental dissonances which can be resolved in the subsequent chord span. While approach is related to Temperley's algorithm, it is more flexible as it does not hard-code one musical model, but instead allows basically any kind of musical preference by redefining the chord transition scoring function. Our approach is also more practical since it not only generates a sequence of root notes, but also the chord qualities.

Most of the more recent approaches are based on some form of machine learning, sometimes explicitly stating the aim to include a "minimal use of music knowledge" [7]. These approaches include Statistical Grammar learning [7], Hidden Markov Models [8–10], and neural networks [11, 12]. [13] presents a hybrid approach based on Markov chains, combining a music theoretic framework with learning from data. Our approach is distinct in that it does not require learning at all, and thus allows for full control over the process of generating the sequences.

3. DATA

The algorithm is evaluated using MTC-FS-INST-2.0, which forms part of the Meertens Tune Collections [14]. The data set consists of c. eighteen thousand melodies, both vocal and instrumental, collected from Dutch sources. The melodies have a variety of time signatures and modes. We use the pre-computed features as distributed in MTCFeatures. [1] Since our model relies on notated meter, we only use the melodies with a meter.

3.1 Music Representation

We represent the melodies as sequences of feature values, one value per note. In this paper, we use three features as provided by MTCFeatures, namely `pitch40`, `beatstrength`, and `beatinsong`, which gives onset times in units of the beat. The base-40 representation of pitch preserves the pitch spelling [15]. It includes 40 values per octave representing 40 possible pitches starting with C𝄫 and ending with B×. We map all pitch values into one octave. We use the encoding as designed by Hewlett with one adaptation: we give the first pitch (C𝄫) index 0 instead of index 1, which has a practical advantage when doing the implementation in Python.

[1] https://zenodo.org/record/3551003

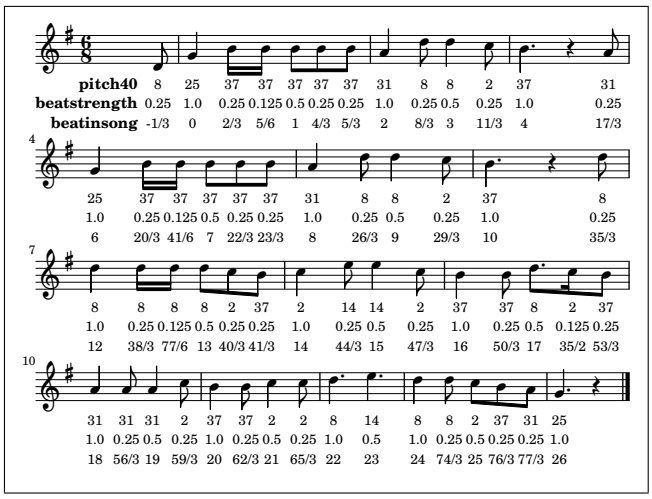


Figure 1. Example melody with the values for `pitch40`, `beatstrength`, and `beatinsong` per note.

We use the beatstrength as computed by the music21 meter model [16, 17]. Music21 is a Python library for processing symbolic musical scores. We heavily use this library. In the meter model of music21, a beatstrength is computed for each note, which indicates the metric weight at the moment of onset of the note. The main accent in the measure gets value 1.0, secondary accents get value 0.5, lower metric positions get 0.25, 0.125, etc. Figure 1 shows an example.

4. METHOD

Our approach to generate a sequence of chords for a given melody consists of three stages: First, we construct for each note a vector summarising the pitch context of that note. Second, we generate for each note a list of potential chords from the pitch context vector. Each chord gets a score indicating the extent to which it fits the pitch context. Finally, for each note, we choose one of the candidate chords, based on its score, and on a chord transition score, such that the sum of all transition scores across the sequence of chords is maximized.

The evaluation also consists of several steps. First, we tune the various parameters on a randomly chosen set of melodies. Next, we use the best parameter setting to generate chord sequences for an independent, disjoint set of melodies. We then provide six music experts with the results and to provide us with a rating of each harmonization on a five-level rating scale. Finally, we use statistics to explore and summarise the responses.

In the following of this section, we will explain each of these steps in detail.

4.1 Weighted Pitch Context Vectors

For a given note, which we indicate as the *focus note*, we consider both a preceding and a following context. These consist of the sequences of notes that are preceding, and respectively following the focus note. For both the preceding and the following context we construct a *weighted pitch context vector*. Each of these vectors has 40 elements corresponding to the 40 pitches in base-40 representation. The value of each of the elements represents the "amount" of the corresponding pitch that is present in the context of the focus note. The full context vector is a 80-dimensional vector which is the concatenation of the preceding and following context vectors.

4.1.1 Length of Contexts

Choosing the length of the contexts is not straightforward. It is, in fact, an important parameter in our model. The music21 meter model provides metric information for each note, notably concerning the *beat* and the *beatstrength* of a note (as explained in Section 3.1). This allows us to express the length of the context as a number of beats. This seems a good approach since the beat is a perceptually meaningful unit. Alternatives would be a fixed number of notes or a certain amount of score time. We did not explore these for the current study.

We experimented with different values for the context length, as well as with a variable context length based on the beatstrengths of the surrounding notes. We found that the latter approach, with variable length, yielded the best results in terms of acceptable chord sequences. In our resulting implementation the context length is computed as follows. We start with the focus note. For the preceding context, we consider the notes before the focus note in reversed order, starting with the note directly before the focus note, and we keep adding the notes to the context until (and including) we reach a note with beatstrength 1.0. For the following context, the same procedure is followed, except that the first encountered note with beatstrength of 1.0 is not included in the context. In our implementation, there is also a parameter whether to include the focus note itself into the context or not. Since the current aim is to generate a chord for the focus note, we always add the focus note to the contexts.

The consequence of this procedure is that for a note on the main accent of the measure (i.e., the first note), the preceding context is the entire previous measure, and the following context is the remainder of the measure of the focus note. In contrast, for notes that are not on the main accent, the preceding context includes all the previous notes in the same measure, while the following context includes the remaining notes in the measure. To a certain extent, this accounts for harmonic progression at different metric levels.

4.1.2 Weighting of Context Notes

The contribution of each context note to the value of the corresponding pitch in the pitch context vector is determined by two components: the metric weight (beatstrength) of the context note, and the distance to the focus note.

Intuitively, the duration of a context note has an impact on its importance in the context of the corresponding focus note. Therefore, we do not simply take the beatstrength of the moment of onset of the context note as weighting factor. Instead, we compute a metric grid, which is a succession of evenly spaced moments in score time. The basic unit of the grid, i.e., the distance between two subsequent positions in the grid, is the greatest common divisor of all note durations. Therefore, each note of the melody starts at a position in the grid, and the "span" of the note mostly includes several grid positions. The metric weighting factor of a context note is the sum of metric weights (beatstrengths) of all positions of the metric grid that are in the "span" of the note. Thus, the duration of the note, as well as the metric importance of the note are incorporated in the weighting. This approach also accounts for syncopation. During the span of a syncopated note, a grid-position with higher metric weight than the metric weight at the start of the note occurs. This is included in the sum.

Also intuitively, the further a note is away from the focus note, the lower the importance in the context of the focus note. In our model, we use a linearly decreasing windowing function. The metric weighting factor of a given context note is multiplied by the value of this window function at the position of the onset of the context note. The value of the window function during the span of the focus note is 1.0, and is linearly decreasing towards the end of the context. The value at the end of the context is a parameter in our model. We set this to a value slightly higher than 0.0 in order to have some influence from the notes that are at the outer boundaries of the contexts.

4.2 Generating Candidate Chords

Once we have computed a pitch context vector consisting of a preceding and following pitch context for each of the notes in a melody, we use these vectors to generate a set of candidate chords for each of the notes in a melody.

In our current implementation, we consider four types of chords: diminished triad, minor triad, major triad, and dominant seventh chord, and we consider three types of context: preceding context, following context, and full context. The full context just is a superposition, i.e., an element-wise sum, of the preceding and following contexts. Discerning these three types of contexts is a crucial element in our model. It allows the method to determine the position of a chord change. If the set of chords that is implied by the preceding context is sufficiently different from the set of chords that is implied by the following context, a chord change is likely, while the presence of a chord that sufficiently fits the full context likely results in a continuation.

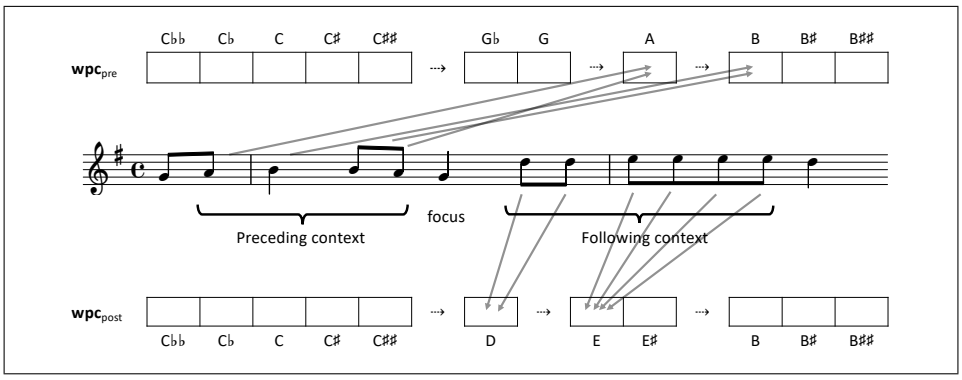


Figure 2. Example of a Pitch Context Vector. The vector consists of two parts, $\mathbf{wpc}_{pre}$ and $\mathbf{wpc}_{post}$, which represent respectively the preceding and the following context. The full Pitch Context Vector is a concatenation of the two parts. Each element in the vector gets a value representing the 'amount' of the corresponding pitch that is present in the context.

Thus, considering 40 possible root notes, we have 160 (40*4) possible chords for each of the preceding, following, and full contexts.

4.2.1 Candidate Chord Score and Strength

For each focus note, we construct a 120*4 matrix, containing a score for each possible chord in each possible context.

The score of a chord with respect to a context vector is determined by two factors: first, the extent to which the chord pitches match the pitches in the context vector, and, second, whether the root note of the chord is present in the local scale. We will explain these two factors in the following.

For each chord quality (diminished, minor, major, and dominant), a chord mask is defined. This is a 40-dimensional binary vector with ones at the positions of the corresponding chord tones. E.g., for a major chord on C♭♭ the positions 0, 12, and 23, corresponding with C♭♭, E♭♭ and G♭♭ are assigned value 1, while other positions get value 0. To compute the score of this chord for a given context, we multiply the mask element-wise with the pitch context vector, and we sum the resulting values. The resulting value represents the overlap between the context and the chord.

To compute the scores for all possible root notes, we subsequently rotate the mask over all possible 40 shifts, and compute for each shift the sum of products. We do this for the preceding context vector, the following context vector, and the full context vector. For the repertoire we have, we do not perform all 40 shifts, we only take into account natural root notes, root notes with one flat, and root notes with one sharp.

Next to these scores, we also compute a *strength* value for each of the possible chords. The strength takes a value between 0 and 1, and is computed as the ratio of the sum of the pitch context values for the chord tones (as determined by the mask) and the sum of all pitch context values. E.g, if a pitch context vector has some weight for C, E, G, and A, a C major chord would get a high score, but a strength lower than 1.0, because there is also weight for the A, which is not a chord tone.

To obtain a single score for each chord candidate, we simply multiply the score with the corresponding strength. This implies a penalty for non-chord tones within the pitch context.

We normalize the score matrix for a given focus note by dividing all scores by the highest score. Thus, the best fitting candidate always has a score of 1.0.

4.2.2 Local Scale

A second factor that determines the possible candidate chords is the local scale. As with the chord mask, we define a scale as a 40-dimensional binary vector. The elements with value 1 are the scale tones. For each note in the melody we derive a local scale vector. This records the alterations of the stemtones that are 'in use' at that position in the melody. For each stemtone $\in \{A, B, C, D, E, F, G\}$, we look for the occurrence closest to the focus note, accepting all possible alterations, and we record the alteration in the scale vector. This accounts for modulations. E.g., if in a melody in D major a G♯ occurs, which is eventually cancelled back to a G, the notes that are closer to the G♯ have a 1 at position 26 in the local scale vector (the base40 representation of G♯) while the notes closer to the G have a 1 at position 25.

One problem is posed if a stemtone is missing altogether in a melody. For example, the melody in Figure 1 lacks the note F. The key signature suggests a F♯, but that is not available to our algorithm. In these cases, we add the tone with the most likely alteration to the scale vector. For sharps, we find this by following the circle of fifths upwards from the missing tone and check the alteration of the next tone. For example, if a C♯ is present in the local scale, we infer that the scale should have a F♯, and not a F natural. For flats, we do the same, but we inspect the circle of fifths in reversed order. For edge cases, we include both the natural and altered tone in the scale. E.g., if stemtone G is missing throughout the melody, and the scale does have a C♯ and a D natural, we include both the G natural and the G♯ as possible scale tones in the local scale vector.

We use the local scale for a given focus note to eliminate those chord candidates that have a root which is not in the scale, by setting its score to 0.0. E.g., a C♯ diminished chord fits a context vector with weight for pitches E and

G, but we eliminate this candidate for a melody in C major because C♯ is not in the scale (except when sometime during the melody the C is temporarily raised).

4.3 The Chord Transition Score

The result of the procedure as described in the previous section is a sequence of matrices, one for each melody note, containing a score for each possible chord for that melody note. The next challenge is to choose one chord for each melody note out of these 120*4 possibilities. For that, we employ a chord transition scoring function (TRS), which computes a score for a given succession of chords, c_1 and c_2 for two subsequent melody notes, n_1 and n_2.

This transition scoring function can be considered a model of what would be a good chord transition. We implement this function as a series of heuristics, each penalizing the score if an aspect of the transition is undesired. We discern two kinds of penalty which could be described as a "total ban" and a "discouragement" respectively. For a total ban, we assign a very low score (-10 in our implementation), which forbids the transition in almost all cases. For a discouragement, we multiply the score by multiplier $\in [0, 1]$. The lower the multiplier, the higher the penalty for the undesired aspect of the chord transition. In our model we include the following heuristics.

- The initial transition score is the candidate score of c_2 for note n_2, as computed in the previous step.
- If the root note of c_2 differs from the root of c_1, multiply with 0.8. This stimulates continuation of a chord.
- If the root notes of both chords are the same, but the chord qualities differ, multiply with 0.1. Except for a change from major to dominant.
- For a root movement other than a prime, a fourth, or a fifth, multiply by 0.75. Root movements of fourths and fifths generally account for good harmonic progression.
- If c_1 is a dominant chord and the root of c_2 is not a fourth higher, multiply by 0.1. We strongly want a V-I relation after a dominant chord.
- If the root of c_2 is a fourth up, and c_1 is not major or dominant, multiply by 0.8.
- If c_1 is diminished, and the root of c_2 is not a semitone up, multiply by 0.1. We strongly want a VII-I relation after a diminished chord.
- If the beatstrength of n_2 is below a threshold, do not allow a chord change (score -10), except for a transition from major to dominant with the same root. The threshold is determined by the meter. For 2/4, 2/8, and 2/2 meter we take 0.25, for all other meters 0.5.
- Do not allow a chord change (score -10) if n_2 is not a chord tone in c_2, and if the beatstrength of n_2 is 0.5 or higher. The seventh of a dominant chord is not considered a chord tone. On strong metric positions, we want chord tones in the melody.
- As an exception to the previous rule, do always allow a chord change to c_2 if the next note after n_2 is a chord tone of c_2, and has a lower beatstrength than n_2. This allows for appoggiaturas.
- Do not allow (score -10) a chord that starts at a low beatstrength (<1.0) to continue past a note with higher beatstrength. Except for a chord that starts on an upbeat. This prevents chord syncopation.
- If the final root change is not a fourth up, or a fifth down, multiply with 0.1.
- If the final root change is a fifth up (a plagal cadence), multiply with 0.8.
- Only allow the root or the third of c_2 as melody note if n_2 is the final note of the melody. If this is not the case assign score -10. If the final note is the third, multiply with 0.75.

4.4 Finding the Optimal Sequence

We designed an algorithm that optimizes the score for a sequence of chord transitions. It takes the sequence of chord score matrices as input and uses the chord transition scoring function. Algorithm 1 shows the pseudo code of our algorithm. We fill a matrix, Score, which contains for each note, and for each possible chord, the total score of the chord sequence up until that note and that chord. In parallel, we fill a traceback matrix, Trace, which for each note, and for each chord, points to the chord of the previous note which is the previous chord in the sequence (i.e., maximises the total score of the chord sequence). After both the Score and Trace matrices are filled, we find the chord sequence by finding the chord with the maximal score for the final note, and following the trace back according to the pointers in the Trace matrix.

Algorithm 1 Algorithm to find the optimal sequence of chord transitions, in which l is the length of the melody in number of notes, Cand is the sequence of matrices with scores for the chord candidates, and TRS is the Chord Transition Scoring Function as defined in Section 4.3.

```
Require: Cand : ARRAY[l][120][4] of float
  function HARMSCORE(Cand)
    declare Score : ARRAY[l][120][4] of float
    declare Trace : ARRAY[l][120][4][2] of int
    Score[0] ← Cand[0]
    for n in {1, 2, ..., l − 1} do
      ixs1 ← indices of cells in Cand[n − 1] > 0
      ixs2 ← indices of cells in Cand[n] >0
      for (p2, c2) in ixs2 do
        declare S : ARRAY[120][4] of float
        for (p1, c1) in ixs1 do
          trs ← TRS(Cand, p1, c1, p2, c2)
          S[p1][c1] ← Score[n − 1][p1][c1] +trs
        end for
        (pm, cm) ← argmax(S)
        Score[n][p2][c2] ← max(S)
        Trace[n][p2][c2] ← (pm, cm)
      end for
    end for
    return Score, Trace
  end function
```

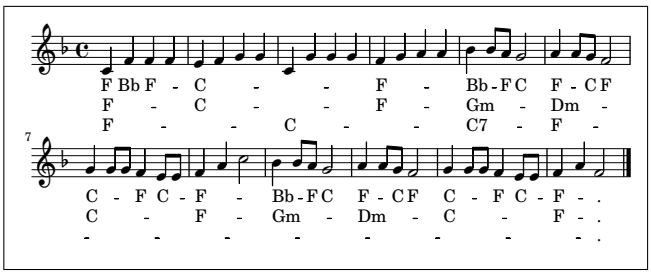


Figure 3. Example of harmonic progressions at various levels of abstraction.

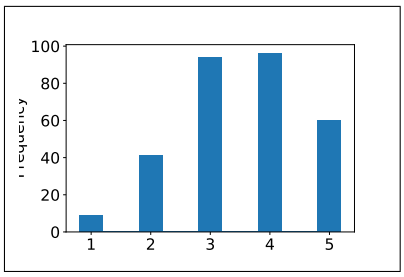


Figure 4. Distribution of ratings.

4.5 Evaluation

To evaluate our algorithm, we first tuned the various parameters ourselves by inspecting the parameter space and chose settings which seemed to yield good results. The values as reported in Section 4.3 are the result of this process.

Next, we randomly chose another unrelated set of 50 melodies and computed chord sequences for these using the parameter values from the previous step. We then asked six music experts to evaluate each harmonization. All evaluators are practicing musicians on a professional level, and have extensive experience in musical analysis. They were given a five level scale and a set of directions in order to rate the harmonizations:

1. Bad. Numerous basic mistakes.
2. Somethings are good but contains a number of incorrect chord choices.
3. Largely okay, small number of incorrect chord choices.
4. Acceptable harmonization.
5. Excellent harmonization. No improvements to be made.

Evaluators were also given a set of directions on how to rate the harmonizations. They were asked to judge to what extent the chords fit the melody, with an emphasis on the correctness of chords with regards to the local context, as opposed to creativity. They were not to take voice leading into consideration for the chord correctness, as the bass line is not modelled in this version of the algorithm.

We then use these ratings to compute inter-rater agreement and explore the extremes.

5. RESULTS

5.1 Parameter Exploration

Exploring the parameter space of our model is an interesting endeavor which appears meaningful in itself. It allows a better understanding of general textbook rules for harmonizing melodies. By implementing and manipulating these principles in our scoring function we can observe the impact of the rigorous application of these principles.

As an example, there are various ways to influence the change rate of chords. Figure 3 shows three sequences of chords at different levels of abstraction. For the middle sequence we used the default parameters as established in the previous sections. The other sequences have been obtained by changing the following parameters with respect to the defaults. The top sequence is generated by tolerating chord changes at every metric level, and by not penalizing root changes. For the bottom sequence we set the context lengths to the length of the entire melody, i.e., all preceding notes are in the preceding context and all following notes are in the following context of a given note. These three sequences show which harmonies are implied by the same melody at different time scales. This could be employed in a hierarchic strategy of harmonization, by e.g. first generating a sequence on a high level to find modulations and extended harmonic sections, and subsequently using that high level sequence as a background for the selection of more fine-grained chord sequences.

5.2 Expert Ratings

Figure 4 shows the distribution of the ratings of the experts. The average over all ratings is 3.52 and the standard deviation is 1.05. It can be observed that only a minority of the harmonizations got a rating lower than 3. Only one harmonization (no. 48) has a highest rating of 2 across the raters, and only six have a highest rating of 3. All 45 others got a 4 or 5 as highest rating. 22 sequences got a 1 or 2 as lowest rating, and 28 sequences 3 or higher. It appears that our algorithm produces an acceptable output, but there are still some issues to address. Some problems we observed are related to tonality, e.g., starting and ending in a different key (mostly the parallel), or including a leading tone at inappropriate places. Also, a low harmonic movement might be unsatisfactory.

6. CONCLUDING REMARKS

We presented a successful approach to generate a sequence of chords to accompany a folk-like melody by leveraging musical expert knowledge and a dynamic programming algorithm to find an optimal trace through the chord space. [2]

There are many directions to further build on the current model. We plan to address the observed shortcomings in a next version. Our framework can be used to explore theory on harmonization or to model implied harmony. It also can serve as tool in educational settings, and of course to generate a accompaniment for a performance.

[2] The full code of our implementation as well as the test set, the expert ratings, and a demo are available at: `https://github.com/pvankranenburg/ismir2023`.

7. ACKNOWLEDGEMENTS

This work has been enabled by the H2020 Project *Polifonia: a digital harmoniser for musical heritage knowledge* funded by the European Commission Grant number 101004746.

8. REFERENCES

[1] J. Bharucha, "Anchoring effects in music: The resolution of dissonance," *Cognitive Psychology*, vol. 16, no. 4, pp. 485–518, 1984.

[2] D. Makris, I. Karydis, and S. Sioutas, "Automatic melodic harmonization: An overview, challenges and future directions," in *Trends in Music Information Seeking, Behavior, and Retrieval for Creativity*. IGI Global, 06 2016, pp. 146–165.

[3] M. Baroni and C. Jacoboni, "Computer generation of melodies: Further proposals," *Computers and the Humanities*, pp. 1–18, 1983.

[4] K. Ebcioğlu, "An expert system for harmonizing four-part chorales," *Computer Music Journal*, vol. 12, no. 3, pp. 43–51, 1988.

[5] H. V. Koops, J. P. Magalhães, and W. B. de Haas, "A functional approach to automatic melody harmonisation," in *Proceedings of the First ACM SIGPLAN Workshop on Functional Art, Music, Modeling amp; Design*. New York, NY, USA: Association for Computing Machinery, 2013, pp. 47–58.

[6] D. Temperley, "An algorithm for harmonic analysis," *Music Perception: An Interdisciplinary Journal*, vol. 15, no. 1, pp. 31–68, 1997.

[7] D. Ponsford, G. Wiggins, and C. Mellish, "Statistical learning of harmonic movement," *Journal of New Music Research*, vol. 28, no. 2, pp. 150–177, 1999.

[8] J.-F. Paiement, D. Eck, and S. Bengio, "Probabilistic melodic harmonization," in *Canadian Conference on AI*, 2006.

[9] I. Simon, D. Morris, and S. Basu, "Mysong: automatic accompaniment generation for vocal melodies," in *CHI '08: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2008, pp. 735–734.

[10] S. A. Raczyński, S. Fukayama, and E. Vincent, "Melody harmonization with interpolated probabilistic models," *Journal of New Music Research*, vol. 42, no. 3, pp. 223–235, 2013.

[11] H. Lim, S. Ryu, and K. Lee, "Chord generation from symbolic melody using blstm networks," in *18th International Society for Music Information Retrieval Conference*, 2017, pp. 621–627.

[12] Y.-C. Yeh, W.-Y. Hsiao, S. Fukayama, T. Kitahara, B. Genchel, H.-M. Liu, H.-W. Dong, Y. Chen, T. Leong, and Y.-H. Yang, "Automatic melody harmonization with triad chords: A comparative study," *Journal of New Music Research*, vol. 50, no. 1, pp. 37–51, 2021.

[13] C.-H. Chuan and E. Chew, "Generating and evaluating musical harmonizations that emulate style," *Computer Music Journal*, vol. 35, no. 4, pp. 64–82, 2011.

[14] P. Van Kranenburg and M. De Bruin, "The meertens tune collections: Mtc-fs-inst 2.0," Meertens Institute, Amsterdam, Meertens Online Reports 2019-1, 2019.

[15] W. B. Hewlett, "A base-40 number-line representation of musical pitch," *Musikometrika*, vol. 4, pp. 1–14, 1992.

[16] M. S. Cuthbert and C. Ariza, "Music21: A toolkit for computer-aided musicology and symbolic music data," in *Proceedings of the 11th International Conference on Music Information Retrieval (ISMIR 2010)*, 2010, pp. 637–642.

[17] C. Ariza and M. S. Cuthbert, "Modeling beats, accents, beams, and time signatures hierarchically with music21 meter objects," in *Proceedings of the International Computer Music Conference*, New York, 2010, pp. 216–223. [Online]. Available: http://mit.edu/music21/papers/2010MeterObjects.pdf

TEXT-TO-LYRICS GENERATION WITH IMAGE-BASED SEMANTICS AND REDUCED RISK OF PLAGIARISM

Kento Watanabe **Masataka Goto**
National Institute of Advanced Industrial Science and Technology (AIST), Japan
{kento.watanabe, m.goto}@aist.go.jp

ABSTRACT

This paper proposes a text-to-lyrics generation method, aiming to provide lyric writing support by suggesting the generated lyrics to users who struggle to find the right words to convey their message. Previous studies on lyrics generation have focused on generating lyrics based on semantic constraints such as specific keywords, lyric style, and topics. However, these methods had limitations because users could not freely input their intentions as text. Even if such intentions can be given as input text, the lyrics generated from the input tend to contain similar wording, making it difficult to inspire the user. Our method is therefore developed to generate lyrics that (1) convey a message similar to the input text and (2) contain wording different from the input text. A straightforward approach of training a text-to-lyrics encoder-decoder is not feasible since there is no text-lyric paired data for this purpose. To overcome this issue, we divide the text-to-lyrics generation process into a two-step pipeline, eliminating the need for text-lyric paired data. (a) First, we use an existing text-to-image generation technique as a text analyzer to obtain an image that captures the meaning of the input text, ignoring the wording. (b) Next, we use our proposed image-to-lyrics encoder-decoder (I2L) to generate lyrics from the obtained image while preserving its meaning. The training of this I2L model only requires pairs of "lyrics" and "images generated from lyrics", which are readily prepared. In addition, we propose for the first time a lyrics generation method that reduces the risk of plagiarism by prohibiting the generation of uncommon phrases in the training data. Experimental results show that the proposed method can generate lyrics with different phrasing while conveying a message similar to the input text.

1. INTRODUCTION

Automatic lyrics generation methods have been proposed as an important research topic in lyrics information processing [1]. With the aim of supporting users who already know what they want to convey in their lyrics but struggle to find the appropriate words, the methods are used in writing support systems providing them with generated lyrics as a source of new inspiration [2–8]. Most previous studies have focused on lyrics generation that is conditioned by semantic constraints, including specific keywords, lyric style, and topics. For example, Watanabe et al.'s system generates lyrics based on pre-defined topics selected by the user, but the limited range of topics results in similar styles of generated lyrics [2]. Oliveira et al.'s system generates poems based on keywords entered by the user, but it cannot generate poems based on sentences or paragraphs representing the user's intention [3, 4].

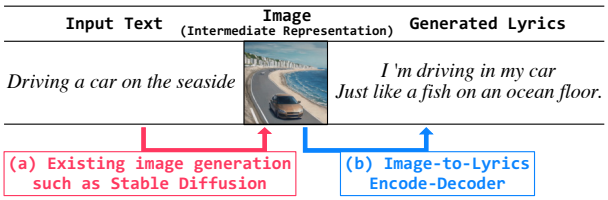


Figure 1. Overview of the proposed text-to-lyrics generation method.

To provide more flexible lyric writing support, we propose generating lyrics based on freely formatted text entered by the user. We believe this approach surpasses the use of semantic constraints such as topics and keywords in terms of flexibility. While existing paraphrase systems [9] can be considered useful for this approach, the paraphrased lyrics may not provide sufficient inspiration because they tend to be similar in wording to the input text. For example, even if a similar phrase "*Driving a car along the coastline*" is generated from the input text "*Driving a car on the seaside*", the user is unlikely to get new inspiration.

Therefore, the aim of this study is to develop a method for generating lyrics that not only have meanings similar to the input text but also use wording different from the input text. For example, if a user freely enters text that represents the content of the lyrics, such as "*Driving a car on the seaside*", our method generates lyrics with different wording, such as "*I'm driving in my car. Just like a fish on an ocean floor.*". As a simple way to achieve this aim, Transformer-based encoder-decoders [10] could be used for generating lyrics from text, but they require large text-lyric paired data for training, which is currently unavailable. To address this issue, we could use text summarization and machine translation to generate text from lyrics and obtain paired data automatically. However, since the generated text

© K. Watanabe and M. Goto. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** K. Watanabe and M. Goto, "Text-to-lyrics generation with image-based semantics and reduced risk of plagiarism", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.*, Milan, Italy, 2023.

and lyric pairs have similar wording, an encoder-decoder trained using those paired data may generate lyrics with wording similar to the input text.

To achieve text-to-lyrics generation without using any paired text data for training, we propose a two-step pipeline framework: (a) using an existing text analyzer to obtain only the semantic representation from the input text, and (b) generating lyrics from the obtained representation. The core idea of this framework is to leverage a text-to-image generation technique such as Stable Diffusion [11] as the text analyzer. An image generated from the input text can serve as a reasonable intermediate representation that captures the meaning of the text while ignoring the details of its wording (Figure 1 (a)). Using the generated image, our image-to-lyrics encoder-decoder generates semantically related lyrics (Figure 1 (b)). It needs many image-lyric pairs as training data, but we can readily prepare those pairs by generating images from lyrics of many songs. This is an advantage of using text-to-image generation. Another advantage is that it can generate images without regard to the input text's format, i.e., whether it is a word, phrase, sentence, or paragraph. We can thus provide flexible lyric writing support that is not constrained by the format of the input text.

Machine learning-based generation methods may inadvertently output portions of the training data directly without modification. This output can be considered plagiarism in some cases [12, 13]. Therefore, this paper also proposes an anti-plagiarism method to reduce this risk. We assume that generating common phrases (word sequences having high commonness [14]) used in many songs is not plagiarism, and reduce the risk of plagiarism by prohibiting the generation of uncommon phrases used in only a few songs. To the best of our knowledge, this is the first study to include such an anti-plagiarism method in lyrics generation.

Experimental results show that our text-to-lyrics generation method can generate lyrics with meaning similar to the input text but expressed differently. Another experiment shows that lyrics generated without using our anti-plagiarism method would result in plagiarizing uncommon phrases in the training data, but those undesirable phrases can successfully be removed by our method.

2. RELATED WORK

While natural language generation methods such as machine translations and chat systems have been actively studied and their performance greatly improved by deep neural networks (DNNs), automatic lyrics generation has also attracted attention as a research topic [1]. Most studies of lyrics generation have focused on lyric-specific musical constraints such as melody [15–20], rhyme [6, 8, 21–25], and audio signal [26–28]. While these lyric-specific musical constraints are an important aspect of lyrics generation, the main focus of this study is on the controllability of the semantic content of the generated lyrics.

Other studies have focused on lyrics generation that is conditioned by semantic constraints such as input keywords, styles, and topics [2–5, 29–32]. However, although these constraints allow some control over the semantic content of the generated lyrics, there may be differences between the user's intentions and the semantic content of the generated lyrics. To improve the usability of the lyrics generation method as a creative tool, we believe that users should be able to enter freely formatted text (words, phrases, sentences, paragraphs, etc.). Our proposed method therefore allows any text format, giving users greater control over the semantic content of the generated lyrics.

Some studies have proposed methods for generating lyrics that are semantically related to the input text [6, 7]. Ram et al. proposed a fine-tuned T5 model [9] that generates single-line lyrics that follow several lines of input lyrics [6]. This method allows the user not only to enter sentences but also to control the rhyme and syllable count of the generated lyrics by adding special tokens at the end of the input sentence. In contrast to that method, in which the generated lyrics are a continuation of the input lyrics, ours generates lyrics that capture the semantic content of the input text. Zhang et al.'s research motivation is similar to ours, as they have also proposed a method for generating lyrics that capture the semantic content of the input text (which they refer to as passage-level text) [7]. To overcome the problem of the lack of text-lyric paired data for training the text-to-lyrics encoder-decoder, they collected lyrics data and passage-level text data (such as short novels and essays) separately and utilized an unsupervised machine translation framework. Specifically, they prepared two encoder-decoders, one for lyric text and one for passage-level text. They then aligned the latent representation space of these two encoder-decoders to build a text-to-lyrics encoder-decoder. In this paper, we propose a novel approach to develop a text-to-lyrics generation method that requires only lyrics data. While Zhang et al.'s method requires the collection of both lyrics and input texts, ours does not require additional text data, thus simplifying the development of the lyrics generation method.

3. TEXT-TO-LYRICS GENERATION WITH IMAGE-BASED SEMANTICS

As described in Section 1, the proposed text-to-lyrics generation method first generates an image from the input text by leveraging an existing text-to-image generation method. It then generates lyrics from the generated image by using our own image-to-lyrics encoder-decoder that we call *I2L*. Since the image serves as an intermediate representation to extract the meaning of the input text, the generated lyrics can have similar meaning but different wording.

The network structure of the I2L is illustrated in Figure 2. By assuming that one paragraph of lyrics can be represented in a single image, we set the unit of the generated lyrics to a paragraph.

We first uses the animation-style image generation method *Anything V3.0.* [1] to obtain an image having a uniform style. The reasons for using Anything V3.0 here are

[1] A fine-tuned Stable Diffusion model. `https://huggingface.co/Linaqruf/anything-v3.0`

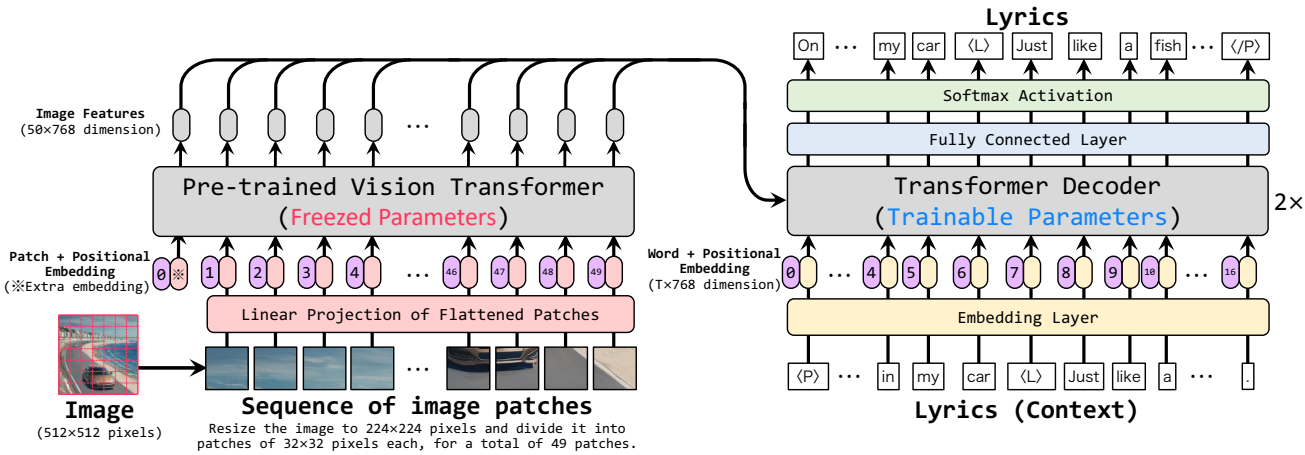


Figure 2. Image-to-lyrics encoder-decoder (I2L) for generating lyrics from an image that is generated from the input text.

(1) it can generate images that represent the input text without prompt engineering, and (2) the use of images with a uniform style facilitates I2L training. The image has a resolution of 512×512 and corresponds to a paragraph of the English lyrics.

As shown in Figure 2, we then segment the generated image into 49 patches and compute the features of the image patches by using a pre-trained Vision Transformer [2] [33] to obtain 50 features (each with 768 dimensions) per image. These 50 image features are fed into the multi-head attention layer of the Transformer decoder [10]. We feed each word in a paragraph into the word embedding and positional embedding layers to compute the word vectors, and feed each word vector into the masked multi-head attention layer of the Transformer decoder. The output of the Transformer decoder is fed into the fully connected layer FC to obtain a vector of vocabulary size dimensions. Finally, we apply the $\mathrm{softmax}$ activation function to this vector to calculate the word probability distribution.

3.1 Parameters

We use 768 as the number of embedding dimensions, 6 as the number of multi-heads, 2 as the number of decoder layers, 1024 as the number of feedforward layer dimensions, and GELU as the activation function. For optimization we use AdamW [34] with a mini-batch size of 8, a learning rate of 0.001, and a warm-up step of one epoch. Training was run for 40 epochs, and the I2L used for testing was the one that achieved the best loss on the development set.

We dare to train our Transformer decoder from scratch using only the lyrics data we have, without reusing available pre-trained large-scale language models (LLMs) such as BERT [35] or GPT-2 [36]. This is because when the training data of LLMs contain copyrighted literary works such as novels, poems, or essays, reusing pre-trained LLMs can result in plagiarizing those works. Since we would like to reduce the risk of plagiarism as described in Section 3.4, we cannot leverage pre-trained LLMs.

[2] https://huggingface.co/google/vit-base-patch32-224-in21k

3.2 Training data

We sample 129,747 English songs from the Music Lyrics Database V.1.2.7 [3] so that each song contains at least three paragraphs. The resulting dataset contains 927,535 paragraphs. This means that we can obtain 927,535 images by using Anything V3.0. We then split these songs into training (90%) and development (10%) sets. We use the top 52,832 words with the highest document-frequency as the vocabulary for training, and convert the other words to a special symbol ⟨unknown⟩. This vocabulary includes ⟨L⟩ tags for line breaks, ⟨P⟩ tags for the beginning of paragraphs, and ⟨/P⟩ tags for the end of paragraphs.

We applied the same procedure not only to the lyrics of English songs but also to the lyrics of 142,772 Japanese songs. This Japanese dataset contains 1,078,500 paragraphs, and the vocabulary size is 50,989 words. To extract word boundaries for Japanese lyrics, we apply the CaboCha parser [37]. Japanese lyrics are pre-translated into English by a Japanese-English translator [4] for use with Anything V3.0. We use these English and Japanese lyrics datasets to train two I2Ls (one for each language).

3.3 Decoding algorithm

We expect that generating and suggesting different variations of lyrics can give users new ideas for writing lyrics. To generate such different variations, we use a sampling method rather than a beam search method. In the sampling method, we sample each word according to the probability distribution calculated by the Transformer decoder. Sampling words according to a probability distribution allows a wide variety of words to be included in the generated lyrics, although some words that make the generated lyrics meaningless may be included. To avoid generating such meaningless lyrics, we use a Top-p sampling method that prohibits sampling words with low generation probabilities [38]. We can generate several lyrics simultaneously by

[3] https://www.odditysoftware.com/page-datasales1.htm

[4] https://huggingface.co/staka/fugumt-en-ja

running Top-p sampling in parallel. The probability distribution for word sampling in Top-p sampling is calculated using the formula $\mathrm{softmax}(\mathbf{z}/\tau)$: where $\mathbf{z}$ is the output of the fully connected layer FC and τ is the temperature parameter. If τ is less than 1, common words with high probability values are more likely to be sampled. In model training we set τ to 1, while in lyrics generation the user can set τ freely.

3.4 Anti-plagiarism method for lyrics generation

One of concerns with lyrics generation based on machine learning is the risk of plagiarism since the generated lyrics may contain phrases that are identical to existing lyrics phrases in training data, potentially leading to copyright infringement issues. To address this issue, we propose a method to reduce the risk of plagiarism in machine learning-based lyrics generation. This method not only allows the generation of new phrases that are not present in the training data, but also permits the use of commonly used phrases such as "*I love you*" in the generated lyrics. In contrast, it prohibits the use of uncommon phrases that we consider to be a form of plagiarism. To achieve this, we create a list of uncommon phrases, $UncommonPhrase$, and prohibit the generation of phrases that are included in this list.

First, we define the uncommon phrases included in $UncommonPhrase$, as well as the new phrases and common phrases that are allowed to be generated. A phrase is defined by a word n-gram, denoted by $\{w_1, ..., w_n\}$, where w is a word. We categorize a phrase as "new", "common", or "uncommon" according to $SN(\{w_1, ..., w_n\})$ defined as the number of songs in which the n-gram occurs in the training data:

- If $SN(\{w_1, ..., w_n\}) = 0$, this n-gram is a new phrase (i.e., it does not appear in the training data).
- If $3 < SN(\{w_1, ..., w_n\})$, this n-gram is a common phrase (i.e., it appears frequently in the training data).
- If $1 \leq SN(\{w_1, ..., w_n\}) \leq 3$, this n-gram is an uncommon phrase (i.e., it appears infrequently in the training data). [5]

Note that there is a possibility of mistaking uncommon phrases for common phrases when duplicate lyrics are contained in the training data, which results in larger SN values than they should be. It could happen when different artists sing the same lyrics, the same lyrics is repeatedly registered, and so on. We therefore identify duplicate lyrics according to the following two criteria: (1) we assume that pairs of lyrics with the same 20-grams are duplicates, and (2) we assume that pairs of lyrics with a normalized edit distance [39] of less than 0.5 are duplicates. To calculate SN accurately, we then concatenate the identified duplicate lyrics and replace those lyrics with the single concatenated lyrics. When lyrics that do not duplicate are mistaken for duplicate lyrics, a common phrase can be mistaken for an uncommon phrase, but it is better than vice versa from the anti-plagiarism viewpoint. This reduced the number of English songs in our lyrics data from 129,747 to 108,497. [6]

Based on this SN criteria, we collect uncommon phrases from our training data. However, it is important to note that even if a word 3-gram is a common phrase, it may become an uncommon phrase when it becomes a word 4-gram. For instance, "*I love you*" is a common 3-gram with a large SN, while "*I love you darling*" is an uncommon 4-gram with a small SN. Therefore we do not use a single value of n but instead consider all values of n within a range from 1 to sufficiently large values. However, it is difficult to store all uncommon phrases in memory because the number of n-grams that have to be listed increases with n. To overcome the memory limitation problem, we propose to use the following procedure to minimize the number of uncommon phrases we need to store in memory: (1) we start by examining 1-grams, then move on to 2-grams, 3-grams, and so on until we have looked at all possible n-grams in the training data. (2) For each target n-gram, we generate all possible sub-n-grams of length 1, 2, ..., $n-1$. If any of these sub-n-grams are already in $UncommonPhrase$, we can skip adding the target n-gram to $UncommonPhrase$ because we know it is uncommon. Otherwise, we add the target n-gram to $UncommonPhrase$. Following this procedure, we collected approximately 22.3M uncommon n-grams with n ranging from 1 to 21 for English lyrics. [7]

After creating $UncommonPhrase$ using the above procedure, we prohibit their generation during Top-p sampling by the following two steps: (1) During word generation, we check whether any sub-n-grams derived from the word sequence $\{w_1, ..., w_t\}$ are included in $UncommonPhrase$. (2) If any of these sub-n-grams are found in $UncommonPhrase$, we prohibit the generation of word w_t by setting its generation probability $P(w_t|\{w_1, ..., w_{t-1}\})$ to zero.

4. QUANTITATIVE EVALUATION

The proposed text-to-lyrics generation method was quantitatively evaluated using two metrics:

Test-set perplexity (PPL): This is a standard evaluation measure for encoder-decoders. The PPL metric measures the degree of predictability of the phrasing in the original text in the test set [40]. A smaller PPL value is better since it indicates that the encoder-decoder has a higher ability to generate lyrics that capture the meaning of the input text.

Normalized edit distance (NED): The normalized edit distance [39] between the generated lyrics and the input text is calculated to evaluate whether the proposed method generates lyrics that differ in wording from the input text. A larger NED is better since it indicates that the generated lyrics have wording more different from the input text.

[5] In this study, we tentatively set the threshold for SN at 3. Since there is no established legal rule, we believe that this threshold will be determined by social consensus in the future. Providing the technical basis for such discussions is also a contribution of this study.

[6] For Japanese lyrics, the number of songs was reduced from 142,772 to 119,595.

[7] For Japanese lyrics, we collected approximately 18.2M uncommon n-grams with n ranging from 1 to 19.

Method	English		Japanese	
	PPL	NED	PPL	NED
I2L (proposed)	**84.86**	**0.78**	**231.49**	**0.92**
S2L	346.73	0.69	306.19	0.86
B2L	544.21	0.71	1051.58	0.66
H2H	163.98	0.68	583.13	0.90

Table 1. Results of quantitative evaluation.

4.1 Experimental dataset

To evaluate the proposed lyrics generation method, we constructed a small test dataset consisting of pairs of lyrics and input text representing the semantic content of the lyrics. Since such a dataset is not available, for English songs, we prepared a test dataset that included plot texts from 20 Disney animated films, taken from Wikipedia, along with their corresponding theme song lyrics. We here assume that the lyrics of each theme song are written based on the content of that film. For Japanese songs, we prepared 51 Japanese animation plot texts and their theme song lyrics.

4.2 Methods Compared

To compare the proposed method with possible different methods, we prepared the following encoder-decoders trained on paired data created in different suitable ways.

Image-to-Lyrics encoder-decoder (I2L) This is the proposed encoder-decoder trained on image-lyric paired data.

Summary-to-Lyrics encoder-decoder (S2L) We converted each lyric paragraph in the training data into a summary using a text summarization method [8] to create summary-lyric paired data. The data is then used to train a Transformer-based summary-to-lyric encoder-decoder.

Back-translated-lyrics-to-Lyrics encoder-decoder (B2L) We translated each lyric paragraph in the training data from English to Japanese to English by using English-Japanese and Japanese-English translation methods [9] to create paired data of the back-translated lyrics and the original lyrics. The data is then used to train a Transformer-based back-translated-lyrics-to-lyrics encoder-decoder.

Half-to-Half encoder-decoder (H2H) Inspired by an existing text-to-lyrics encoder-decoder training method [6], we first split each lyrics paragraph in the training data into first and second halves. We then used this split lyrics data to train a Transformer-based encoder-decoder that generates the second half lyrics from the first half lyrics.

Since the above S2L, B2L, and H2H are also Transformer-based encoder-decoders, their parameter settings are the same as for the proposed I2L. Given one input text, five lyrics were generated by each method. The parameter p for Top-p sampling was set to 0.9 and τ was set to 0.4. The generation process stops when the symbol ⟨/P⟩ (end of paragraph) is generated. For this comparison, we did not use the proposed anti-plagiarism method.

[8] `https://huggingface.co/google/pegasus-xsum` for the English summarization. `https://huggingface.co/tsmatz/mt5\_summarize\_japanese` for the Japanese summarization.

[9] `https://huggingface.co/staka/fugumt-en-ja` for the English to Japanese translation. `https://huggingface.co/staka/fugumt-ja-en` for the Japanese to English translation.

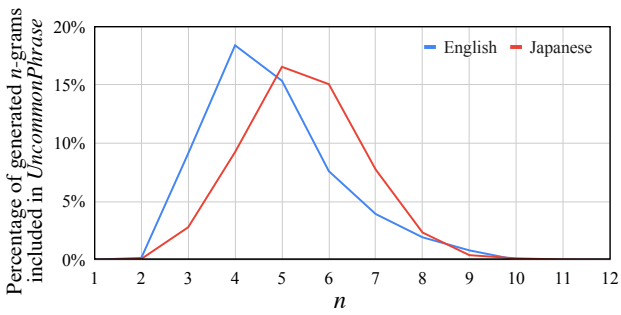


Figure 3. The percentage of generated lyric n-grams that are included in $UncommonPhrase$, a list of phrases that should not be generated (plagiarized). For example, 18.4% at English 4-grams means that among all 4-gram phrases in the generated lyrics, 18.4% are uncommon phrases, though 81.6% are new or common phrases.

4.3 Experimental results

Table 1 indicates that the proposed I2L method had the best PPL in both the English and Japanese experiments and that the NED between the lyrics generated by this method and the input text was the largest ($p_t < 0.05$ based on the paired t-test). As expected, the NEDs were smaller for the S2L and B2L methods, which were trained on paired data where the wording of the input text and lyric pairs was similar. In contrast, although the H2H method can generate lyrics with wording different from the input text, it cannot generate lyrics that are semantically related to the input text like the proposed method can. These findings confirm that image-lyric pairs are more effective than other paired data sets as training data for encoder-decoders generating lyrics that are semantically related to the input text but differ from it in wording.

5. EFFECTIVENESS OF THE PROPOSED ANTI-PLAGIARISM METHOD

We examined whether the absence of the anti-plagiarism method proposed in Section 3.4 results in plagiarizing uncommon phrases found in existing lyrics. In the lyrics generated by the I2L method in Section 4, we calculated the percentage of n-grams included in $UncommonPhrase$.

The results with n ranging from 1 to 12 are shown in Figure 3. The percentage of uncommon 1-grams and 2-grams in the generated lyrics is almost 0%. This indicates that almost all of the generated 1-grams and 2-grams are common phrases used in many existing lyrics, even without the use of the anti-plagiarism method. On the other hand, the percentage of uncommon 3-grams to 8-grams ranged between 3% and 18%. This suggests that many phrases in the generated lyrics may plagiarize if the proposed anti-plagiarism method is not applied. Furthermore, as n increases beyond 9, the n-gram combinations become

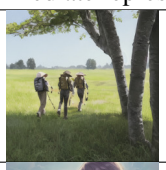
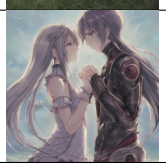
Input text		Image (intermediate representation)		Generated lyrics
A group of explorers are walking through the grass neutral.	⇒		⇒	Out in the country, out of sight We've got to get this together right now I'm going out with you today And some day we'll make a lot better way
We meet again I guess our love is forever.	⇒		⇒	This is the last time We've been together for long years I'm here with you To be forever yours

Table 2. Examples of lyrics generated by our text-to-lyrics generation method with the anti-plagiarism method.

so vast that the generated n-grams are rarely included in $UncommonPhrase$. These results confirm that our machine learning-based lyrics generation method tends to sample common words, but the generated 3- to 8-gram phrases, even though they are composed of common words, may be uncommon enough to raise suspicion of plagiarism. Using the proposed anti-plagiarism method, in contrast, ensures that uncommon phrases contained in $UncommonPhrase$ are never generated, thereby reducing the risk of plagiarism.

While the proposed anti-plagiarism method is effective, it is important to note that it is not intended to be a foolproof solution that ensures legal compliance. Rather, it is designed to provide a helpful guideline for those who wish to generate original lyrics while reducing the risk of plagiarism. We hope that our approach can contribute to further discussions on a reasonable balance between encouraging creativity and respecting intellectual property rights.

6. QUALITATIVE EVALUATION

Table 2 shows two examples of lyrics generated using the proposed method. Given the input text, our method can generate any number of lines of lyrics, but here four lines are generated by stopping the generation process when four ⟨L⟩ (line break) symbols and the ⟨/P⟩ (end of paragraph) symbol are generated. In the first example, the input text is taken from the SICK dataset [41], while in the second example the input text is taken from lyrics in the RWC Music Database [42]. In both examples, our method can generate lyrics that reflect the content of the input text. In the first example, it generates an image that represents the scene described in the input text and generates corresponding lyrics that reflect the image. In contrast, in the second example, our method generates an image of a person with emotional expression corresponding to the input text and generates lyrics that express the emotion depicted in the image. Other examples can be found in the supplementary material A. [10]

In addition to the quantitative evaluation and the generated examples, we evaluated the similarity between the input text and the generated lyrics through a human evaluator. To prepare the input text in an objective way, we collected the titles of the "Hot 100 Songs" in 2022 on the Billboard year-end charts [11], extracted the first verse from their lyrics, and summarized each verse into a short sentence using ChatGPT. [12] Since 9 songs contained explicit content in either the input text or the generated lyrics, they were excluded for ethical reasons. [13] We then showed the evaluator the input text and the lyrics generated from it, and asked to classify whether the impressions of the two were similar or not. As a result, the impressions of the input text and the generated lyrics were judged to be similar for 52 of the 91 songs, confirming that the proposed method can generate lyrics that express the content of the input text to some extent. In cases where the impressions were classified as dissimilar, most of the input texts contain complex situations or abstract content that is difficult to generate as images. Thus, the limitation of this approach is that it cannot generate lyrics for input texts that are difficult to represent as images. Nevertheless, our method is useful as a writing support tool for many situations where users have intentions that can be represented as images, and is also valuable because it pioneered a novel lyric generation approach. Detailed results of the generated lyrics and the judgments are included in the supplementary material B. [14]

7. CONCLUSION

This paper has described a method for generating lyrics that are similar in meaning to the input text but expressed differently. The contributions of this study are as follows: (1) We proposed a novel two-step pipeline framework. First, we apply text-to-image generation as a text analyzer to extract only the semantic content from the input text. Next, we use our proposed image-to-lyrics encoder-decoder to generate lyrics that capture the semantics of the generated image. (2) We proposed a method to reduce the risk of plagiarism by prohibiting the generation of uncommon phrases in the training data and verified its effectiveness. (3) We quantitatively showed that our proposed method outperforms other methods in generating lyrics for our purpose.

Future work will develop the flexible lyric writing support system incorporating the proposed lyrics generation method.

[10] https://github.com/KentoW/ISMIR2023

[11] https://www.billboard.com/charts/year-end/2022/hot-100-songs/

[12] https://chat.openai.com/chat

[13] As future work, we plan to incorporate a filtering function that uses explicit lyrics detection [43–46].

[14] https://github.com/KentoW/ISMIR2023

8. ACKNOWLEDGMENTS

This work was supported in part by JST CREST Grant Number JPMJCR20D4 and JSPS KAKENHI Grant Number JP20K19878, Japan.

9. REFERENCES

[1] K. Watanabe and M. Goto, "Lyrics information processing: Analysis, generation, and applications," in *Proceedings of the 1st Workshop on NLP for Music and Audio (NLP4MusA)*, 2020, pp. 6–12.

[2] 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 (ACM IUI)*, 2017, pp. 559–563.

[3] H. G. Oliveira, T. Mendes, and A. Boavida, "Co-PoeTryMe: A co-creative interface for the composition of poetry," in *Proceedings of the 10th International Conference on Natural Language Generation (INLG)*, 2017, pp. 70–71.

[4] H. G. Oliveira, T. Mendes, A. Boavida, A. Nakamura, and M. Ackerman, "Co-PoeTryMe: Interactive poetry generation," *Cognitive Systems Research*, vol. 54, pp. 199–216, 2019.

[5] R. Zhang, X. Mao, L. Li, L. Jiang, L. Chen, Z. Hu, Y. Xi, C. Fan, and M. Huang, "Youling: An AI-assisted lyrics creation system," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (EMNLP)*, 2020, pp. 85–91.

[6] N. Ram, T. Gummadi, R. Bhethanabotla, R. J. Savery, and G. Weinberg, "Say what? collaborative pop lyric generation using multitask transfer learning," in *Proceedings of the 9th International Conference on Human-Agent Interaction (HAI)*, 2021, pp. 165–173.

[7] L. Zhang, R. Zhang, X. Mao, and Y. Chang, "QiuNiu: A Chinese lyrics generation system with passage-level input," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics - System Demonstrations (ACL)*, 2022, pp. 76–82.

[8] N. Liu, W. Han, G. Liu, D. Peng, R. Zhang, X. Wang, and H. Ruan, "ChipSong: A controllable lyric generation system for Chinese popular song," in *Proceedings of the First Workshop on Intelligent and Interactive Writing Assistants (In2Writing)*, 2022, pp. 85–95.

[9] 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.

[10] 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. 1–11, 2017.

[11] 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, pp. 10 674–10 685.

[12] A. Papadopoulos, P. Roy, and F. Pachet, "Avoiding plagiarism in markov sequence generation," in *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, 2014, pp. 2731–2737.

[13] Q. Feng, C. Guo, F. Benitez-Quiroz, and A. M. Martínez, "When do GANs replicate? on the choice of dataset size," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV 2021)*, 2021, pp. 6681–6690.

[14] T. Nakano, K. Yoshii, and M. Goto, "Musical similarity and commonness estimation based on probabilistic generative models of musical elements," *International Journal of Semantic Computing (IJSC)*, no. 1, pp. 27–52, 2016.

[15] K. Watanabe, Y. Matsubayashi, S. Fukayama, M. Goto, K. Inui, and T. Nakano, "A melody-conditioned lyrics language model," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 2018, pp. 163–172.

[16] X. Lu, J. Wang, B. Zhuang, S. Wang, and J. Xiao, "A syllable-structured, contextually-based conditionally generation of Chinese lyrics," in *Proceedings of the 16th Pacific Rim International Conference on Artificial Intelligence (PRICAI)*, 2019, pp. 257–265.

[17] Y. Chen and A. Lerch, "Melody-conditioned lyrics generation with SeqGANs," in *Proceedings of the 2020 IEEE International Symposium on Multimedia (ISM)*, 2020, pp. 189–196.

[18] Y. Huang and K. You, "Automated generation of Chinese lyrics based on melody emotions," *IEEE Access*, vol. 9, pp. 98 060–98 071, 2021.

[19] X. Ma, Y. Wang, M. Kan, and W. S. Lee, "AI-Lyricist: Generating music and vocabulary constrained lyrics," in *Proceedings of the 29th ACM International Conference on Multimedia (ACM-MM)*, 2021, pp. 1002–1011.

[20] 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 35th AAAI Conference on Artificial Intelligence*, 2021, pp. 13 798–13 805.

[21] G. Barbieri, F. Pachet, P. Roy, and M. D. Esposti, "Markov constraints for generating lyrics with style," in *Proceedings of the 20th European Conference on Artificial Intelligence (ECAI)*, vol. 242, 2012, pp. 115–120.

[22] J. Hopkins and D. Kiela, "Automatically generating rhythmic verse with neural networks," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2017, pp. 168–178.

[23] E. Manjavacas, M. Kestemont, and F. Karsdorp, "Generation of hip-hop lyrics with hierarchical modeling and conditional templates," in *Proceedings of the 12th International Conference on Natural Language Generation (INLG)*, 2019, pp. 301–310.

[24] L. Xue, K. Song, D. Wu, X. Tan, N. L. Zhang, T. Qin, W. Zhang, and T. Liu, "DeepRapper: Neural rap generation with rhyme and rhythm modeling," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, 2021, pp. 69–81.

[25] J. Chang, J. C. Hung, and K. Lin, "Singability-enhanced lyric generator with music style transfer," *Computer Communications*, vol. 168, pp. 33–53, 2021.

[26] O. Vechtomova, G. Sahu, and D. Kumar, "Generation of lyrics lines conditioned on music audio clips," in *Proceedings of the 1st Workshop on NLP for Music and Audio (NLP4MusA)*, 2020, pp. 33–37.

[27] ——, "LyricJam: A system for generating lyrics for live instrumental music," in *Proceedings of the 12th International Conference on Computational Creativity (ICCC)*, 2021, pp. 122–130.

[28] K. Watanabe and M. Goto, "Atypical lyrics completion considering musical audio signals," in *Proceedings of the 27th International Conference on Multimedia Modeling (MMM)*, vol. 12572, 2021, pp. 174–186.

[29] K. Watanabe, Y. Matsubayashi, K. Inui, and M. Goto, "Modeling structural topic transitions for automatic lyrics generation," in *Proceedings of the 28th Pacific Asia Conference on Language, Information and Computation (PACLIC)*, 2014, pp. 422–431.

[30] P. Potash, A. Romanov, and A. Rumshisky, "GhostWriter: Using an LSTM for automatic rap lyric generation," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2015, pp. 1919–1924.

[31] M. Ghazvininejad, X. Shi, Y. Choi, and K. Knight, "Generating topical poetry," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2016, pp. 1183–1191.

[32] H. Fan, J. Wang, B. Zhuang, S. Wang, and J. Xiao, "A hierarchical attention based seq2seq model for Chinese lyrics generation," in *Proceedings of the 16th Pacific Rim International Conference on Artificial Intelligence (PRICAI)*, vol. 11672, 2019, pp. 279–288.

[33] 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 *Proceedings of the 9th International Conference on Learning Representations (ICLR)*, 2021.

[34] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proceedings of the 7th International Conference on Learning Representations (ICLR)*, 2019.

[35] 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, pp. 4171–4186.

[36] 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, p. 9, 2019.

[37] T. Kudo and Y. Matsumoto, "Japanese dependency analysis using cascaded chunking," in *Proceedings of the 6th Conference on Natural Language Learning (CoNLL)*, 2002.

[38] A. Holtzman, J. Buys, L. Du, M. Forbes, and Y. Choi, "The curious case of neural text degeneration," in *Proceedings of the 8th International Conference on Learning Representations (ICLR)*, 2020.

[39] Y. Li and B. Liu, "A normalized Levenshtein distance metric," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1091–1095, 2007.

[40] C. Manning and H. Schutze, *Foundations of statistical natural language processing*. MIT press, 1999.

[41] M. Marelli, S. Menini, M. Baroni, L. Bentivogli, R. Bernardi, and R. Zamparelli, "A SICK cure for the evaluation of compositional distributional semantic models," in *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC)*, 2014, pp. 216–223.

[42] 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, pp. 287–288.

[43] H. Chin, J. Kim, Y. Kim, J. Shin, and M. Y. Yi, "Explicit content detection in music lyrics using machine learning," in *Proceedings of the 2018 IEEE International Conference on Big Data and Smart Computing (BigComp)*, 2018, pp. 517–521.

[44] M. Fell, E. Cabrio, M. Corazza, and F. Gandon, "Comparing automated methods to detect explicit content in song lyrics," in *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP)*, 2019, pp. 338–344.

[45] E. Egivenia, G. R. Setiawan, S. S. Mintara, and D. Suhartono, "Classification of explicit music content based on lyrics, music metadata, and user annotation," in *Proceedings of the 6th International Conference on Sustainable Information Engineering and Technology (SIET)*, 2021, pp. 265–270.

[46] M. Rospocher, "On exploiting transformers for detecting explicit song lyrics," *Entertainment Computing*, vol. 43, p. 100508, 2022.

Papers – Session IV

LP-MusicCaps: LLM-BASED PSEUDO MUSIC CAPTIONING

SeungHeon Doh[♭] **Keunwoo Choi**[♮] **Jongpil Lee**[♯] **Juhan Nam**[♭]
[♭] Graduate School of Culture Technology, KAIST, South Korea
[♮] Gaudio Lab, Inc., South Korea
[♯] Neutune, South Korea
{seungheondoh, juhan.nam}@kaist.ac.kr, keunwoo@gaudiolab.com, jongpillee@neutune.com

ABSTRACT

Automatic music captioning, which generates natural language descriptions for given music tracks, holds significant potential for enhancing the understanding and organization of large volumes of musical data. Despite its importance, researchers face challenges due to the costly and time-consuming collection process of existing music-language datasets, which are limited in size. To address this data scarcity issue, we propose the use of large language models (LLMs) to artificially generate the description sentences from large-scale tag datasets. This results in approximately 2.2M captions paired with 0.5M audio clips. We term it **L**arge Language Model based **P**seudo music caption dataset, shortly, **LP-MusicCaps**. We conduct a systemic evaluation of the large-scale music captioning dataset with various quantitative evaluation metrics used in the field of natural language processing as well as human evaluation. In addition, we trained a transformer-based music captioning model with the dataset and evaluated it under zero-shot and transfer-learning settings. The results demonstrate that our proposed approach outperforms the supervised baseline model. [1]

1. INTRODUCTION

Music captioning is a music information retrieval (MIR) task of generating natural language descriptions of given music tracks. The text descriptions are usually sentences, distinguishing the task from other music semantic understanding tasks such as music tagging. Recently, there have been some progress in music captioning including track-level captioning [1, 2] and playlist-level captioning [3–6]. These approaches usually utilize a deep encoder-decoder framework which is originally developed for neural machine translation [7]. Choi *et al.* [3] used a pre-trained music tagging model as a music encoder and an RNN layer initialized with pre-trained word embeddings for text generation. Manco *et al.* [1] introduced a temporal attention mechanism for alignment between audio and text by pairing a pre-trained harmonic CNN encoder [8] with an LSTM layer. Gabbolini *et al.* [5] generated playlist titles and descriptions using pre-trained GPT-2 [9].

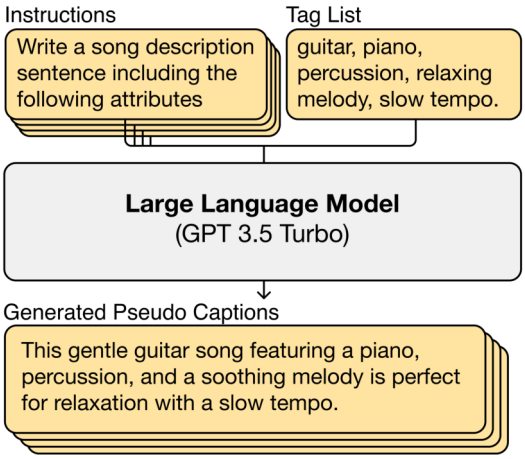


Figure 1. The generation process of pseudo captions by feeding a large language model with instructions and manually-annotated labels.

Currently, the primary challenge of track-level music captioning is the scarcity of large-scale public datasets. Manco *et al.* [1] used private production music datasets. Huang *et al.* [10] also used a private dataset with 44M music-text pairs on YouTube, but this approach is hardly reproducible or affordable for other researchers. To address this data issue, a community-driven data collection initiative has been proposed [11]. As of now, the only publicly available dataset for track-level music captioning is MusicCaps [12], which includes high-quality music descriptions from ten musicians. However, it is limited to 5521 music-caption pairs as it was originally created as an evaluation set for a text-prompt music generator.

With the scale of the aforementioned datasets, it remains difficult to train a music captioning model successfully. A workaround for this situation is to use music tagging datasets and generate sentences with tag concatenation [2, 13] or prompt template [14]. As relying on tagging datasets, however, the tag-to-sentence approaches would have the same limitation tagging datasets have. For example, high false-negative rates of tagging datasets [15]. Tag-

[1] Our dataset and codes are available at https://github.com/seungheondoh/lp-music-caps

© SeungHeon Doh, Keunwoo Choi, Jongpil Lee, Juhan Nam. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** SeungHeon Doh, Keunwoo Choi, Jongpil Lee, Juhan Nam, "LP-MusicCaps: LLM-Based Pseudo Music Captioning", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

ging datasets also has some typical issues text data have, for example, synonyms, punctuation, and singular/plural inconsistencies. Without proper treatment, these can limit the performance of the corresponding music captioning models.

A potential solution is to use strong language models, i.e., large language models (LLMs). LLMs refer to the recent large-scale models with over a billion parameters that exhibit strong few-shot and zero-shot performance [9, 16]. Large language models are usually trained with text data from various domains such as Wikipedia, GitHub, chat logs, medical articles, law articles, books, and crawled web pages [17]. When successfully trained, they demonstrate an understanding of words in various domains [9]. There have been similar and successful use cases of LLMs for general audio understanding [18] and music generation [19].

Motivated by the recent success of LLMs, we propose creating a music captioning dataset by applying LLMs carefully to tagging datasets. Our goal is to obtain captions that are i) semantically consistent with the provided tags, ii) grammatically correct, and iii) with clean and enriched vocabulary. This dataset-level approach is rather pragmatic than sophisticated; it alleviates the difficulty of music captioning tasks not by theory or model, but by data. The aforementioned ambiguous aspects of the music captioning task are addressed by the powerful LLMs that cost reasonably [20], considering the training cost music researchers would spend otherwise. Once the creation is complete, it is straightforward to train some music captioning models by supervised learning.

There are some existing works in the pseudo-labeling using language models. Huang *et al.* [19] introduced the MuLaMCap dataset, which consists of 400k music-caption pairs generated using the large language model and the music-language joint embedding model. They utilized a large language model (LaMDA [21]) to generate 4M sentences using 150k song metadata as input in the format of `{title} by {artist}`. Then the text and music-audio joint embedding model, MuLan, calculates the similarity between music and generated captions, annotating pairs with high similarity [10]. However, it is not possible to reproduce or evaluate this work as the adopted language model as well as the final music-audio embedding model are not publicly available. Moreover, using metadata has some issues – a popularity-biased, limited coverage and a low reliability – as we discuss later in Section 2.1. Wu *et al.* [22] introduce keyword-to-caption augmentation (K2C Aug) to generate captions based on the ground truth tags of audio clips in AudioSet. They used a pre-trained T5 model without any instruction. Finally, Mel *et al.*. [18] introduce WavCaps, a 400k audio captioning dataset using ChatGPT [23]. However, previous approaches only reported task performance and did not directly evaluate the quality of generated captions.

We propose a solution in this paper with three-fold key contribution. First, we propose an LLM-based approach to generate a music captioning dataset, **LP-MusicCaps**. Second, we propose a systemic evaluation scheme for music captions generated by LLMs. Third, we demonstrate that models trained on LP-MusicCaps perform well in both zero-shot and transfer learning scenarios, justifying the use of LLM-based pseudo-music captions.

2. PSEUDO CAPTION GENERATION USING LARGE LANGUAGE MODELS

In this section, we introduce how music-specific pseudo captions are created using a large language model in the proposed method.

2.1 Large Language Model for Data Generation

We first take multi-label tags from existing music tagging datasets. The list of tags are appended with a carefully written task instruction as an input (prompt) to a large language model. The model then generates and returns sentences that (may) describe the music in a way the task instruction conditions. Table 1 shows examples of generated captions according to multi-label tags and task instructions. For the language model, we choose GPT-3.5 Turbo [23] for its strong performance in various tasks. During its training, it was first trained with a large corpus and immense computing power, then fine-tuned by reinforcement learning with human feedback (RLHF) [24] for better interaction with given instruction. As a result, GPT-3.5 Turbo demonstrates state-of-the-art zero-shot abilities in understanding, reasoning, and generating human-like responses to natural language inputs.

Since LLMs contain a wide range of information, music captions may be generated based on some famous musical entities such as the artist name or album name. However, LLMs may generate inaccurate text in a confident tone which is hard to detect without ground truth. This issue, known as hallucination, can be a fun aspect when using LLMs for creative purposes [25]. However, hallucination should be avoided in an application like ours as the resulting captions should be factual. Therefore, we do not use any metadata unlike a previous work [19]. We also added a question to measure hallucination in the proposed evaluation scheme.

2.2 Task Instruction Design

Our proposed caption generation follows the formulation: $\tilde{y}_{\text{cap}} = f_{\text{LLM}}(y_{\text{tag}}, i)$, where y_{tag} and $\tilde{y}_{\text{cap}}$ refer to the multi-label tag and the generated caption, respectively, and i is the task instruction provided. Given that the output can vary based on the task instruction, even with the same model and input, task instructions become a crucial aspect of data generation. Therefore, we define four different tasks and generate captions accordingly.

Writing: *Write a song description sentence including the following attributes.* `{input tags}`

Summary: *Write a single sentence that summarizes a song with the following attributes. Don't write the artist name or album name.* `{input tags}`

Input tags video game theme, no singer, instrumental, analog sounding, small keyboard, beatboxing, playful, cheerful, groovy
Result of "Writing" Instruction This instrumental track has a joyful and playful vibe, perfect for a video game theme. With no singer, the analog-sounding music features a small keyboard and beatboxing, creating a groovy and cheerful atmosphere.
Result of "Summary" Instruction A cheerful and groovy instrumental track with playful beatboxing and an analog sound, featuring a small keyboard and video game-inspired theme.
Result of "Paraphrase" Instruction This upbeat instrumental tune captures the essence of a fun-filled video game theme, incorporating analog-sounding beats, playful beatboxing rhythms, and a groovy small keyboard melody, all without the need for a singer.
Result of "Attribute Prediction" Instruction {"new_attribute": ["8-bit sound", "chiptune style", "retro vibe"], "description": "This instrumental tune is straight out of a video game with its analog sounding melodies and small keyboard tinkles. Beatboxing adds a playful element to the groovy, cheerful vibe. Reminiscent of classic 8-bit sound and chiptune style, this retro vibe is sure to put a smile on your face."}
Ground Truth This is a video game theme cover. The theme belongs to the Super Mario franchise. The main theme is being played on an analog sounding small keyboard. There is an added rhythmic background of beatboxing in this version. The atmosphere is playful. This piece could be used in the background of arcade gaming social media content.

Table 1. An example of generated captions from MusicCaps dataset.

Paraphrase: *Write a song description sentence including the following attributes. Creative paraphrasing is acceptable.* `{input tags}`

Attribute Prediction: *Write the answer as a Python dictionary with new_attribute and description as keys. For new_attribute, write new attributes that show high co-occurrence with the following attributes. For description, write a song description sentence including the following attributes and new attributes.* `{input tags}`

In every instruction, we add 'include / with the following attributes' to prevent hallucination. The "Writing" task instruction is a simple prompt that uses tags to generate a sentence. The "Summary" task instruction aims to compress information into a short length. The "Paraphrase" task instruction expands the vocabulary. Finally, the "Attribute Prediction" task instruction predicts new tags based on tag co-occurrence in large corpora (i.e. the training data of GPT-3.5 Turbo), which is expected to address the issue of high false-negative rates in existing tagging datasets while mitigating the risk of hallucination. In this instruction, 'new attributes' exists to bridge the description and the input, and we only use the 'description' as caption.

3. EVALUATION OF PSEUDO CAPTIONS

It is crucial to ensure the quality of generated captions, especially since they are supposed to be used as ground truth. In this section, we introduce a holistic evaluation scheme that includes objective and subjective assessment – and its result on the captions from the proposed method.

3.1 Objective Evaluation

We conduct evaluation on the generated captions using MusicCaps dataset [12]. It has audio (x), tag list (y_{tag}), and ground truth caption (y_{cap}). The pseudo captions ($\tilde{y}_{\text{cap}}$) are generated with four pre-defined instructions as explained in Section 2.2 for all items in the evaluation split. During the evaluation, the generated captions are compared to the ground truth captions with respect to *n*-gram, neural metrics. We also report diversity metrics.

Following the previous work [5], we measure four *n*-gram metrics [26–28]: BLEU1 to 4 (B1, B2, B3, B4), METEOR (M), and ROUGE-L (R-L). They are all based on *n*-gram precision and recall between the ground truth and generated captions. These metrics capture different aspects of the caption quality. BLEU and METEOR focus on *n*-gram overlap between the generated and ground truth captions, while ROUGE-L measures the longest common subsequence between the two.

In addition, we use BERT-Score (BERT-S) based on pre-trained BERT embeddings to represent and match the tokens in the ground truth with respect to the generated caption [29]. By computing the similarity between the BERT embeddings of each token, BERT-Score can better capture the semantic similarity between the generated and ground truth captions than *n*-gram metrics; as it is more robust to synonyms, paraphrasing, and word order variations.

Finally, we evaluate the diversity of the generated captions by measuring how many different words are used. $novel_v$ indicates the percentage of new vocabulary in generated captions that are not among the training vocabulary. *Vocab* is the number of unique words used in all the generated captions. It is worth noting that diversity metrics are generally considered as subsidiaries and do not capture the overall quality of the generated captions.

3.2 Subjective Evaluation

Following the previous work [12], we set up an A-vs-B human rating task, in which a participant is presented with a 10-second single music clip and two text descriptions. We randomly selected 240 music samples from the MusicCaps evaluation dataset. Since the research goal is to generate

			Supervised Metrics							Diversity Metrics		Length
Methods	LM	Params	B1↑	B2↑	B3↑	B4↑	M↑	R-L↑	BERT-S↑	Vocab↑	Novel_{v}↑	Avg.Token
Baseline												
Tag Concat [2, 13]	-	-	20.25	13.57	8.64	5.42	23.24	19.52	86.24	3506	46.92	20.6±11.2
Template [14]	-	-	25.41	16.15	10.00	6.15	25.57	21.36	87.92	3507	46.93	25.6±11.2
K2C Aug. [22]	T5	220M	6.07	3.01	1.58	0.85	14.23	17.92	86.33	3760	**67.66**	14.7±5.1
Proposed Instruction												
Writing	GPT3.5	175B+	**36.84**	**19.85**	**11.37**	**6.74**	31.44	25.36	89.26	5521	56.17	44.4±17.3
Summary	GPT3.5	175B+	26.12	14.58	8.80	5.52	27.58	**25.83**	**89.88**	4198	49.52	28.6±10.7
Pharapase	GPT3.5	175B+	36.51	18.73	10.33	5.87	30.36	23.40	88.71	6165	59.95	47.9±18.7
Attribute Prediction	GPT3.5	175B+	35.26	18.16	9.69	5.41	**34.09**	23.19	88.56	**6995**	63.16	66.2±21.6

Table 2. Performance of existing pseudo caption generation methods and the proposed method. LM stand for the language model. Avg.Token stand for the average number of token per caption.

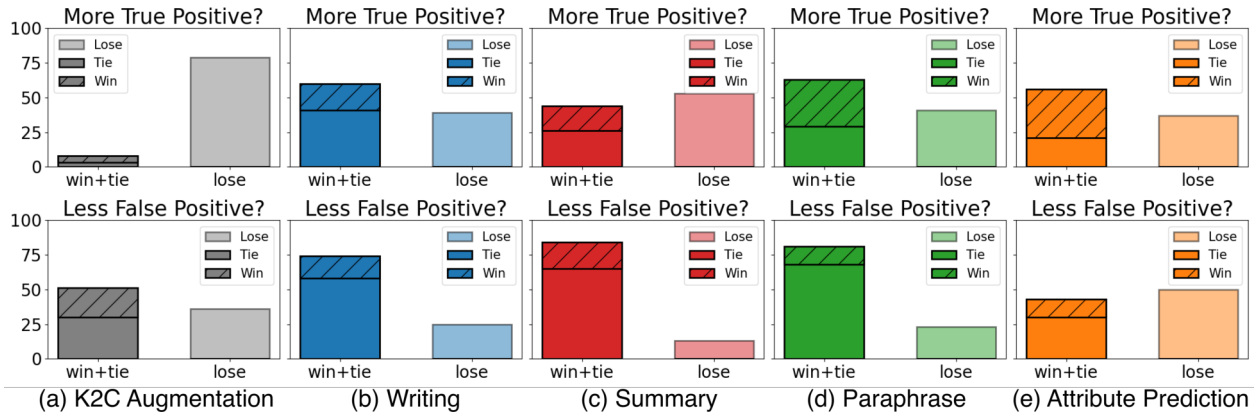


Figure 2. A-vs-B test results. Each method is compared to ground truth in terms of having more true positives and fewer false positives. The proposed methods (b, c, d, e) show comparable **win+tie** performance to ground truth.

music captions that can be used as pseudo-ground truth, one description is always fixed to the ground truth and the other is chosen from 5 types of generated captions including the K2C Augmentation [22] and the four proposed instruction methods. This yields up to 1200 (= 240 x 5) questions. We hired 24 participants who are music researchers or professionals in the music industry. Each of them rated 20 randomly selected questions. As a result, we collected a total of 480 ratings. The rater was asked to evaluate caption quality on two different aspects: (Q1) *More True Positive*: which caption describes the music with more accurate attributes? (Q2) *Less False Positive*: which caption describes the music less wrong? For example, if a method produces long and diverse sentences with many music attributes, it may be advantageous for Q1 but disadvantageous for Q2. Conversely, if a method conservatively produces short sentences with few music attributes, it may be advantageous for Q2 but disadvantageous for Q1. We determine the ranking of conditions by counting the number of wins, ties, and loses in the pairwise tests.

3.3 Results

We compare our LLM-based caption generation with two template-based methods (tag concatenation, prompt template [2]) and K2C augmentation [22]. In Table 2, we present the captioning result for MusicCaps [12] evaluation set. When comparing our proposed method with existing methods, we observe significant differences in *n*-gram metrics. This is because the tag concatenation fails to complete the sentence structure. In the case of K2C Augmentation, due to the absence of instruction, the input tag is excluded from the generated caption, or a sentence unrelated to the song description sentence is created. In contrast, the template-based model shows improved performance as the musical context exists in the template. We next consider diversity metric with BERT-Score. Our proposed method shows higher values in BERT-Score while generating diverse vocabularies. This indicates that the newly created vocabulary does not harm the music semantics.

Comparing within the proposed different task instructions, we can observe that each instruction performs a different role. "Writing" shows a high n-gram performance as it faithfully uses input tags to generate captions. "Summary" has the smallest average number of tokens due to its compression of information, but it shows competitive performance in ROUGE-L which is specialized to summarizing, as well as the highest BERT-Score. "Paraphrase" generates many synonyms, resulting in a large vocabulary size and the use of novel vocabulary. "Attribute Prediction" predicts new tags based on the co-occurrence of tags. This instruction shows lower performance in BLEU but competitive results in METEOR, which utilizes a thesaurus, such as WordNet, to consider the accuracy scores of words with similar meanings, indicating that newly predicted tags have similar semantic with ground truth.

Figure 2 shows the subjective A-vs-B test results. Each

[2] Template example: the music is characterized by `{input tags}`

Dataset	# item	Duration (h)	C/A	Avg. Token
General Audio Domain				
AudioCaps [30]	51k	144.9	1	9.0±N/A
LAION-Audio [22]	630k	4325.4	1-2	N/A
WavCaps [18]	403k	7568.9	1	7.8±N/A
Music Domain				
MusicCaps [12]	6k	15.3	1	48.9±17.3
MuLaMCap* [19]	393k	1091.0	12	N/A
LP-MusicCaps-MC	6k	15.3	4	44.9±21.3
LP-MusicCaps-MTT	22k	180.3	4	24.8±13.6
LP-MusicCaps-MSD	514k	4283.1	4	37.3±26.8

Table 3. Comparison of audio-caption pair datasets. C/A stands for the number of caption per audio. *Although we include MuLaMCap in the table for comparison, it is not publicly accessible.

method is compared to the ground truth in terms of having more true positives (Q1) and fewer false positives (Q2). For the first question, compared to the baseline K2C augmentation, the proposed methods using the instructions show an overwhelmingly higher *win+tie* score. This indicates the importance of music-specific instructions when utilizing LLM. In particular, "Paraphrase" and "Attribute Prediction" achieve high *win* scores by incorporating new information that is different from the existing vocabulary. In the second question, all caption generation methods except "Attribute Prediction" show higher *win+tie* scores than *lose* scores. This advocates the trustworthiness of LLM-based caption generation as it shows a similar or less false-positive rate to the ground truth. With its longest average length, "Attribute Prediction" turns out to be 'too creative' and shows a slightly higher false-positive rate than the ground truth.

4. DATASET: LP-MusicCaps

Based on the proposed pseudo caption generation method, we introduce LP-MusicCaps, an LLM-based Pseudo music caption dataset. We construct the music-to-caption pairs using three existing multi-label tag datasets and four task instructions. The data sources are MusicCaps [12], Magnatagtune [31], and Million Song Dataset [32] ECALS subset [13]. We respectively refer to them as MC, MTT, and MSD. MC contains 5,521 music examples,[3] each of which is labeled with 13,219 unique aspects written by music experts. MTT [31] consists of 26k music clips from 5,223 unique songs including genre, instrument, vocal, mood, perceptual tempo, origin, and sonority features. We used the full 188 tag vocabulary and did not generate captions for tracks that do not have associated tags (decreased to 22k). MSD consists of 0.52 million 30-second clips and 1054 tag vocabulary [13]. The tag vocabulary covers various categories including genre, style, instrument, vocal, mood, theme, and culture. Each dataset uses an average of 10.7 / 3.3 / 10.2 labels per music clip for generating pseudo captions, respectively.

Table 3 provides a comparison of statistics between the LP-MusicCaps family and other audio-caption pair datasets. When comparing the two domains, AudioCaps [30] and MusicCaps have high-quality human annotated captions, but they have fewer captions with shorter audio duration. When comparing large-scale datasets, the music domain lacks available datasets compared to the general audio domain (such as LAION-Audio [22] and WavCaps [18]). Although MuLaMCap has an overwhelming amount of annotated captions, it is not publicly available. In contrast, LM-MusicCaps is publicly accessible and provided with various scales. LP-MusicCaps-MC has a similar caption length to manually written captions while having four times more captions per audio. LP-MusicCaps-MTT is a medium-sized dataset with audio download link, and LP-MusicCaps-MSD has the largest audio duration among various captions in the music domain caption dataset.

[3] We only use 5495 out of the total due to the loss of 26 data samples.

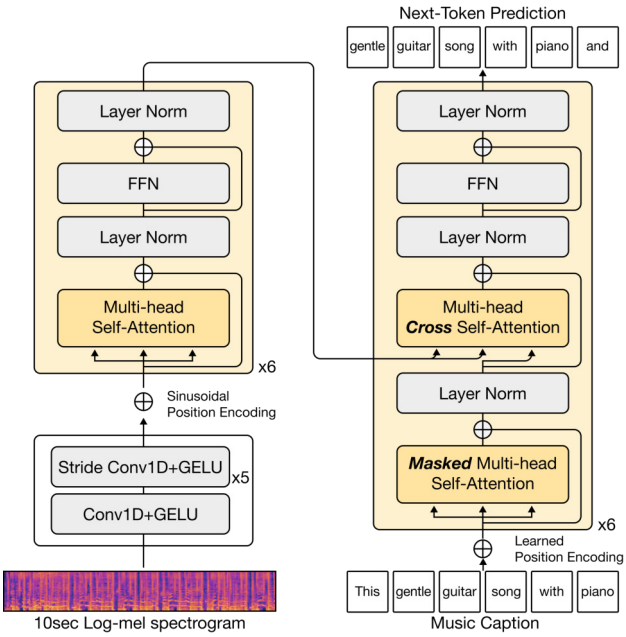


Figure 3. A cross-modal encoder-decoder architecture.

5. AUTOMATIC MUSIC CAPTIONING

We trained a music captioning model and evaluated it under zero-shot and transfer-learning settings. This section reports the experimental results.

5.1 Encoder-Decoder Model

We used a cross-modal encoder-decoder transformer architecture that has achieved outstanding results on various natural language processing tasks [33], lyrics interpretation [34], and speech recognition [35], as shown in Figure 3. Similar to Whisper [35], the encoder takes a log-mel spectrogram with six convolution layers with a filter width of 3 and the GELU [36] activation function. With the exception of the first layer, each convolution layer has a stride of two. The output of the convolution layers is combined with the sinusoidal position encoding and then processed by the encoder transformer blocks. Following the $\text{BART}_{\text{base}}$ architecture, our encoder and decoder both have 768 widths and 6 transformer blocks. The decoder

	Supervised Metrics							Diversity Metrics			Length
Model	B1↑	B2↑	B3↑	B4↑	M↑	R-L↑	BERT-S↑	Vocab↑	Novel_v↑	Novel_c↑	Avg.Token
Baseline											
Supervised Model	28.51	13.76	7.59	4.79	20.62	19.22	87.05	2240	0.54	69.00	46.7±16.5
Zeroshot Captioning											
Tag Concat [2, 13]	4.33	0.84	0.26	0.00	3.10	2.01	79.30	802	46.38	100.00	23.8±12.1
Template [14]	7.22	1.58	0.46	0.00	5.28	6.81	81.69	787	45.24	100.00	25.8±12.4
K2C-Aug [22]	7.67	2.10	0.49	0.10	7.94	11.37	82.99	**2718**	**81.97**	100.00	19.9±7.6
LP-MusicCaps **(Ours)**	**19.77**	**6.70**	**2.17**	**0.79**	**12.88**	**13.03**	**84.51**	1686	47.21	100.00	45.3±28.0
Tansfer Learning											
Tag Concat [2, 13]	28.65	14.68	8.68	5.82	21.88	21.31	87.67	1637	3.30	96.07	41.8±14.3
Template [14]	28.41	14.49	8.59	5.78	21.88	21.25	87.72	1545	**3.62**	**96.77**	41.1±13.2
K2C-Aug [22]	**29.50**	**14.99**	8.70	5.73	21.97	20.92	87.50	**2259**	1.42	84.95	44.1±15.0
LP-MusicCaps **(Ours)**	29.09	14.87	**8.93**	**6.05**	**22.39**	**21.49**	**87.78**	1695	1.47	96.06	42.5±14.3

Table 4. Music captioning results on the MusicCaps eval-set. Avg.Token stands for the average number of token per caption.

processes tokenized text captions using transformer blocks with a multi-head attention module that includes a mask to hide future tokens for causality. The music and caption representations are fed into the cross-modal attention layer, and the head of the language model in the decoder predicts the next token autoregressively using the cross-entropy loss, formulated as: $\mathcal{L} = -\sum_{t=1}^{T} \log p_\theta(y_t \mid y_{1:t-1}, x)$ where x is the paired audio clip and y_t is the ground truth token at time t in a caption with length T.

5.2 Experimental Setup

To evaluate the impact of the proposed dataset on the music captioning task, we compare a supervised model trained on the MusicCaps [12] training split and a pre-trained model trained on an LP-MusicCaps-MSD dataset. For the pre-trained model, we perform both a zero-shot captioning task that does not use any MusicCaps [12] dataset and a fine-tuning task that updates the model using MusicCaps [12] training split. For comparison with other pseudo caption generation methods, we report results on baseline models trained with the same architecture and amount of audio, but different pseudo captions. In addition to all the metrics we used in Section 3.1, we compute $Novel_c$, the percentage of generated captions that were not present in the training set [37]. It measures whether the captioning model is simply copying the training data or not.

For all the experiments, the input of the encoder is a 10-second audio signal at 16 kHz sampling rate. It is converted to a log-scaled mel spectrogram with 128 mel bins, 1024-point FFT with a hann window, and a hop size of 10 ms. All models are optimized using AdamW with a learning rate of 1e-4. We use a cosine learning rate decay to zero after a warmup over the first 1000 updates. For the pre-training dataset, we use 256 batch-size and the models are trained for 32,768 updates. We adopt a balanced sampling [38], which uniformly samples an anchor tag first and then selects an annotated item. For supervised and transfer learning, we use a 64 batch size, 100 epochs. We use beam search with 5 beams for the inference of all models.

5.3 Results

When comparing within zero-shot captioning models, the model trained on the proposed LP-MusicCaps dataset shows a strong performance in general. The model using tag concatenation shows the lowest performance as it fails to generate musical sentences. In case of the model using a prompt template, it demonstrates a slightly higher BERT-Score, while still exhibiting poor performance in terms of n-gram metrics due to its limited vocabulary. The model using K2C augmentation outperforms the other two methods but still falls short due to its lack of a musical context. In general, zero-shot models does not perform as well as the supervised baseline in most of the metrics with few exceptions.

Among the transfer captioning models, the model with LP-MusicCaps pre-training achieves strong performance overall by winning in the BERT-Score and most of the n-gram metrics. It is noteworthy that our proposed model shows a meaningful increase in BERT-Score compared to the supervised model. This improvement is likely a result of successful semantic understanding rather than word-to-word matching. Moreover, by the improvement of Novel_c, the LP-MusicCaps model demonstrates that it can generate new captions instead of repeating the phrases in the training dataset. This advantage is observed in both the zero-shot and supervised tasks in transfer learning models.

6. CONCLUSION

We proposed a tag-to-pseudo caption generation approach with large language models to address the data scarcity issue in automatic music captioning. We conducted a systemic evaluation of the LLM-based augmentation, resulting in the creation of the LP-MusicCaps dataset, a large-scale pseudo-music caption dataset. We also trained a music captioning model with LP-MusicCaps and showed improved generalization. Our proposed approach has the potential to significantly reduce the cost and time required for music-language dataset collection and facilitate further research in the field of connecting music and language, including representation learning, captioning, and generation. However, further collaboration with the community and human evaluation is essential to enhance the quality and accuracy of the generated captions. Additionally, we believe that exploring the use of LLMs for other topics under music information retrieval and music recommendation could lead to novel and exciting applications.

7. REFERENCES

[1] I. Manco, E. Benetos, E. Quinton, and G. Fazekas, "Muscaps: Generating captions for music audio," in *International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2021.

[2] T. Cai, M. I. Mandel, and D. He, "Music autotagging as captioning," in *Proceedings of the 1st Workshop on NLP for Music and Audio (NLP4MusA)*, 2020.

[3] K. Choi, G. Fazekas, B. McFee, K. Cho, and M. Sandler, "Towards music captioning: Generating music playlist descriptions," in *International Society for Music Information Retrieval Conference (ISMIR), Late-Breaking/Demo*, 2016.

[4] S. Doh, J. Lee, and J. Nam, "Music playlist title generation: A machine-translation approach," in *Proceedings of the 2nd Workshop on NLP for Music and Spoken Audio (NLP4MuSA)*, 2021.

[5] G. Gabbolini, R. Hennequin, and E. Epure, "Data-efficient playlist captioning with musical and linguistic knowledge," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2022.

[6] H. Kim, S. Doh, J. Lee, and J. Nam, "Music playlist title generation using artist information," in *Proceedings of the AAAI-23 Workshop on Creative AI Across Modalities*, 2023.

[7] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014.

[8] M. Won, S. Chun, O. Nieto, and X. Serrc, "Data-driven harmonic filters for audio representation learning," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.

[9] 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," in *Proceedings of the Advances in neural information processing systems (NeurIPS)*, 2020.

[10] 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 *International Conference on Music Information Retrieval (ISMIR)*, 2022.

[11] I. Manco, B. Weck, P. Tovstogan, M. Won, and D. Bogdanov, "Song describer: a platform for collecting textual descriptions of music recordings," in *International Conference on Music Information Retrieval (ISMIR), Late-Breaking/Demo session*, 2022.

[12] 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.

[13] S. Doh, M. Won, K. Choi, and J. Nam, "Toward universal text-to-music retrieval," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023.

[14] T. Chen, Y. Xie, S. Zhang, S. Huang, H. Zhou, and J. Li, "Learning music sequence representation from text supervision," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.

[15] 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.

[16] 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, no. 1, jan 2020.

[17] L. Gao, S. Biderman, S. Black, L. Golding, T. Hoppe, C. Foster, J. Phang, H. He, A. Thite, N. Nabeshima, S. Presser, and C. Leahy, "The pile: An 800gb dataset of diverse text for language modeling," 2020.

[18] X. Mei, C. Meng, H. Liu, Q. Kong, T. Ko, C. Zhao, M. D. Plumbley, Y. Zou, and W. Wang, "WavCaps: A ChatGPT-assisted weakly-labelled audio captioning dataset for audio-language multimodal research," *arXiv preprint arXiv:2303.17395*, 2023.

[19] 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.

[20] F. Gilardi, M. Alizadeh, and M. Kubli, "ChatGPT outperforms crowd-workers for text-annotation tasks," *arXiv preprint arXiv:2303.15056*, 2023.

[21] R. Thoppilan, D. De Freitas, J. Hall, N. Shazeer, A. Kulshreshtha, H.-T. Cheng, A. Jin, T. Bos, L. Baker, Y. Du *et al.*, "Lamda: Language models for dialog applications," *arXiv preprint arXiv:2201.08239*, 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 International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023.

[23] 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," in *Proceedings of*

the Advances in neural information processing systems (NeurIPS), 2022.

[24] P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei, "Deep reinforcement learning from human preferences," in *Proceedings of the Advances in neural information processing systems (NeurIPS)*, 2017.

[25] 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 Computing Surveys*, 2023.

[26] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting of the Association for Computational Linguistics (ACL)*, 2002.

[27] S. Banerjee and A. Lavie, "Meteor: An automatic metric for mt evaluation with improved correlation with human judgments," in *Proceedings of the ACL workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, 2005.

[28] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Text summarization branches out*, 2004.

[29] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "Bertscore: Evaluating text generation with bert," in *International Conference on Learning Representations (ICLR)*, 2020.

[30] 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*, 2019.

[31] 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 Conference on Music Information Retrieval (ISMIR)*, 2009.

[32] T. Bertin-Mahieux, D. P. Ellis, B. Whitman, and P. Lamere, "The million song dataset," in *International Conference on Music Information Retrieval (ISMIR)*, 2011.

[33] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020.

[34] Y. Zhang, J. Jiang, G. Xia, and S. Dixon, "Interpreting song lyrics with an audio-informed pre-trained language model," in *International Conference on Music Information Retrieval (ISMIR)*, 2022.

[35] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," *arXiv preprint arXiv:2212.04356*, 2022.

[36] D. Hendrycks and K. Gimpel, "Gaussian error linear units (GELUs)," *arXiv preprint arXiv:1606.08415*, 2016.

[37] M. Stefanini, M. Cornia, L. Baraldi, S. Cascianelli, G. Fiameni, and R. Cucchiara, "From show to tell: a survey on deep learning-based image captioning," *IEEE transactions on pattern analysis and machine intelligence*, 2022.

[38] 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)*, 2021.

A REPETITION-BASED TRIPLET MINING APPROACH FOR MUSIC SEGMENTATION

Morgan Buisson[1] **Brian McFee**[2,3] **Slim Essid**[1] **Hélène C. Crayencour**[4]
[1] LTCI, Télécom Paris, Institut Polytechnique de Paris, France
[2] Music and Audio Research Laboratory, New York University, USA
[3] Center of Data Science, New York University, USA
[4] L2S, CNRS-Univ.Paris-Sud-CentraleSupélec, France

ABSTRACT

Contrastive learning has recently appeared as a well-suited method to find representations of music audio signals that are suitable for structural segmentation. However, most existing unsupervised training strategies omit the notion of repetition and therefore fail at encompassing this essential aspect of music structure. This work introduces a triplet mining method which explicitly considers repeating sequences occurring inside a music track by leveraging common audio descriptors. We study its impact on the learned representations through downstream music segmentation. Because musical repetitions can be of different natures, we give further insight on the role of the audio descriptors employed at the triplet mining stage as well as the trade-off existing between the quality of the triplets mined and the quantity of unlabelled data used for training. We observe that our method requires less non-annotated data while remaining competitive against other unsupervised methods trained on a larger corpus.

1. INTRODUCTION

The task of music structure analysis consists in locating the boundaries between consecutive segments and grouping them into relevant categories, called musical sections. This problem has gained attention in the field of music information retrieval and has numerous applications, such as music generation [1, 2], music recommendation [3] or music similarity estimation [4]. Structure is also strongly linked to other musical elements such as harmony, melody and rhythm [5] and has been leveraged to address other tasks such as beat and downbeat tracking [6] or chord transcription [7].

Most methods that have been proposed for the task of music structure analysis can be categorized according to the structure trait they rely on, namely: *homogeneity*, *novelty* and *repetition* [8]. The homogeneity rule states that musical attributes should be relatively homogeneous inside musical segments or sections. Consequently, transitions from one segment to the next should result in points of important changes in musical features (*i.e.* novelty). The idea of repetition in structure assumes that sections of the same type are rather similar sequences. In other words, musical sections are generally characterized by the degree at which they repeat throughout the entire music piece, which has been the starting point of many algorithms to infer song structures [9–11]. However, both the extent to which two sequences can be considered as repetitions, or how homogeneous a given musical segment is, imply a certain definition of similarity between time instants. Such similarity criteria are usually derived from frame representations based on common audio descriptors such as harmonic and timbral features, or their combinations [8].

A line of work has focused on finding better-suited audio representations so as to make sure that frames from the same musical sections yield similar features and therefore, sharpen transitions between consecutive musical segments. Methods based on contrastive learning have recently been proposed to find such representations [12–15], as they can leverage commonalities from large quantities of music data to learn a distance metric that complies with the aforementioned requirements. Training such models either involves the use of structural annotations [13] or some pre-defined proxies to select frames that should be brought close to one another in the latent space [12, 15]. In the latter case, these heuristics mainly rely on the homogeneity principle and discard the notion of repetition occurring inside a track, preventing them from fully exploiting unlabelled data.

The method introduced in this work aims at bridging the gap between current unsupervised deep metric learning methods for music segmentation and both ideas of homogeneity and repetition that are inherent to musical structure. As in previous work [12, 15], a contrastive learning pipeline using a triplet loss is adopted. However, triplets are mined by seeking repeating sequences inside the input track with respect to various hand-crafted audio features. In a preliminary analysis, a qualitative evaluation of the triplets generated is performed by direct comparison with structural annotations of a manually annotated test dataset. We then measure how these representations impact down-

© M. Buisson, B. McFee, S. Essid and H. C. Crayencour. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** M. Buisson, B. McFee, S. Essid and H. C. Crayencour, "A repetition-based triplet mining approach for music segmentation", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

stream segmentation on two datasets for music structure analysis. Finally, we demonstrate that our approach requires less non-annotated data than previous similar methods. We also give further insight on how the choice of the input features used to mine triplets affects training and its relationship with the music genre that the resulting representations are tested on.

2. RELATED WORK

Numerous methods for music structure analysis rely on measuring similarity between every point of a music recording to retrieve homogeneous segments and transitions between them. Since music is naturally multi-dimensional, many factors such as harmony, timbre or instrumentation can be associated with boundaries between musical sections [16]. Therefore, several strategies have been adopted to capture short-term similar regions, and it has been shown that sharp timbre changes can be a good cue for section transitions [17–19].

However, not all boundaries can be explained solely by such changes in musical features, as the perception of structure is also greatly affected by additional characteristics of a music recording such as parallelism, pauses or musical rules proper to the music genre considered [16]. Therefore, other approaches tend to rely on the repetition principle to characterise the structure of a music piece. For example, early work on music segmentation has attempted to find audio representations to identify repeating elements inside music recordings, such as pitch estimation or polyphonic transcription [10]. Generally, repetition-based methods rely on harmony-related information from the audio, as the instrumentation or other factors are subject to variations between different occurrences of a given musical section [18, 20].

Several algorithms have also been proposed to unify these two types of approaches by recognizing similar regions and repetitions of varying lengths. For example, integrating structural information at different scales into frame representations has led to considerable improvements in the recognition of musical segments [21, 22].

Even though these methods are theoretically well grounded and have proven to be efficient on commonly used datasets, the traditional hand-crafted descriptors they use can fail at accommodating different structure types and music genres. On the other hand, deep learning-based methods are able to extract efficient features from large quantities of data, thus, surpassing traditional audio descriptors [12]. Approaches based on contrastive learning also have the advantage to be easily incorporated into the classical music structure analysis pipeline, by simply replacing the original input features by the deep embeddings they learn from training data. To this end, Wang *et al.* [13] use structural annotations from a labelled training dataset to find positive and negative pairs of frames and a multi-similarity loss function [23]. They additionally employ a mining mechanism to further improve convergence of their model. Using structural annotations allows for explicitly enforcing frames of identical sections to yield similar features regardless of their appearance throughout the track. Despite not relying on annotations, the method in this work is similar to theirs, in the sense that it explicitly considers section repetitions inside a music recording.

A similar method proposed by McCallum [12] proceeds in an unsupervised manner with a triplet loss. This time, positive and negative frames are sampled using time proximity as a proxy: frames occurring within a small time interval are more likely to belong to the same musical sections than those separated by a larger amount of time. While this assumption generally holds true, it completely discards the notion of repetition, which can limit the efficacy of the approach. In the present work, this limitation is addressed by using pairwise frame similarity measures as prior information to guide the triplet sampling mechanism. This temporal-based mining method [12] is used as a baseline in this work and referred to as *temporal sampling*.

3. METHOD

The core of the triplet mining method proposed in this work resides in the estimation of a self-similarity matrix, which should reflect as much as possible section label assignment corresponding to structural annotations. This approximation of ideal pairwise frame similarities should yield high values for frames belonging to the same musical section, and low values otherwise. This self-similarity matrix is used as a probability mass function according to which are sampled, for each given frame, positive and negative examples across the whole input track.

3.1 Triplet loss

The method proposed in this work consists in finding triplets of audio feature patches (x_a, x_p, x_n) where x_a is the anchor, x_p is a positive example from the same musical section and x_n the negative example sampled from a different one without using structural annotations. The models are trained using the triplet loss, which for a given triplet $\mathcal{T} = (x_a, x_p, x_n)$ is expressed as:

$$\mathcal{L}(\mathcal{T}) = [d(f(x_a), f(x_p)) - d(f(x_a), f(x_n)) + \delta]_+ \ , \quad (1)$$

where $d(x, y)$ is a pre-defined distance metric, $[.]_+$ denotes the Hinge loss, $\delta > 0$ is the margin parameter, and $f(x)$ is the projection of x into the embedding space by a deep neural network.

3.2 Finding repetitions

The choice of the input features from which frame-wise similarities are extracted greatly influences the final triplet sampling mechanism. As the goal is to jointly detect homogeneous regions and overall repetitions throughout the input track, we employ a combination of timbral and harmonic features as done in previous work [24, 25]. These features are beat-synchronized beforehand, using the algorithm from Korzeniowski *et al.* [26] implemented in the *madmom* package [27]. One way to emphasize repetition is to encode features into time-delay embeddings, so

that pairwise comparisons are performed over short time-windows: given a sequence $X = \{\mathbf{X}_i\}_{i\in\{1,\dots,N\}}$ of feature vectors, the ith time embedding vector $\tilde{\mathbf{X}}_i$ is obtained by stacking the m feature vectors ranging from $i-(m-1)$ to i:

$$\tilde{\mathbf{X}}_i = \left[\mathbf{X}_i^T \mathbf{X}_{i-1}^T \dots \mathbf{X}_{i-(m-1)}^T\right]^T, \tag{2}$$

where m denotes the embedding dimension, ruling how much of past information is considered. Such transformations have successfully been used for music structure analysis [22], structure-based music similarity [4] and more generally in the field of non-linear time series analysis [28]. The final representation's temporal dimension remains N, as X is first zero-padded before transformation. Then, a self-similarity matrix is built from the obtained sequence of time-lag features such that:

$$M(i,j) = \begin{cases} \exp\left(-\frac{d(\tilde{\mathbf{X}}_i,\tilde{\mathbf{X}}_j)}{b}\right), & \tilde{\mathbf{X}}_j \in \mathrm{NN_k}(\tilde{\mathbf{X}}_i) \\ 0, & \tilde{\mathbf{X}}_j \notin \mathrm{NN_k}(\tilde{\mathbf{X}}_i) \end{cases} \tag{3}$$

where $d(x,y)$ is the euclidean distance, b the bandwidth parameter, $\mathrm{NN_k}(x)$ denotes the k-nearest neighbors of x and $i,j = 1,\dots,N$. The self-similarity matrix M is then filtered with a sigmoid activation, such that:

$$\hat{M}(i,j) = \sigma\left(\frac{M(i,j)}{\max_k M(i,k)}\right), \tag{4}$$

where $i,j = 1,\dots,N$ and the σ function defined as:

$$\sigma(x) = \frac{1}{1+e^{-\alpha(x-\beta)}}, \tag{5}$$

where $\alpha > 0$ is a parameter ruling the steepness of the curve and $\beta \in [0,1]$ a threshold above which the components of S are set to values close to 1. This process is applied both using MFCC and chroma features, from which we obtain their respective filtered self-similarity matrices S_M and S_C using Equation (4) (first row of Figure 1). The matrix S is then obtained by linear combination, such that:

$$S = \gamma S_M + (1-\gamma) S_C, \tag{6}$$

where $\gamma \in [0,1]$ weights the contributions of each feature type. The matrix S (second row, left column of Figure 1) is row-wise min-max normalized and filtered with the sigmoid function defined in Equation (5), diagonal stripes indicating repeating sequences are enhanced by median filtering similar to the one used by McFee *et al.* [18].

3.3 Imposing segment homogeneity

The obtained pairwise similarity S provides information about the repetitions present inside the input track. However, using it as it is to mine positive (large $S(a,p)$) and negative examples (small $S(a,n)$) would result in many trivial triplets, as positives would be located at exact points of repetitions. Therefore, a dilation operation is applied to the matrix S to enlarge these detected regions of repetition. Similar to the method by Serra *et al.* [22], a two-dimensional Gaussian kernel G of size K is convolved with S:

$$S_p = S * G, \tag{7}$$

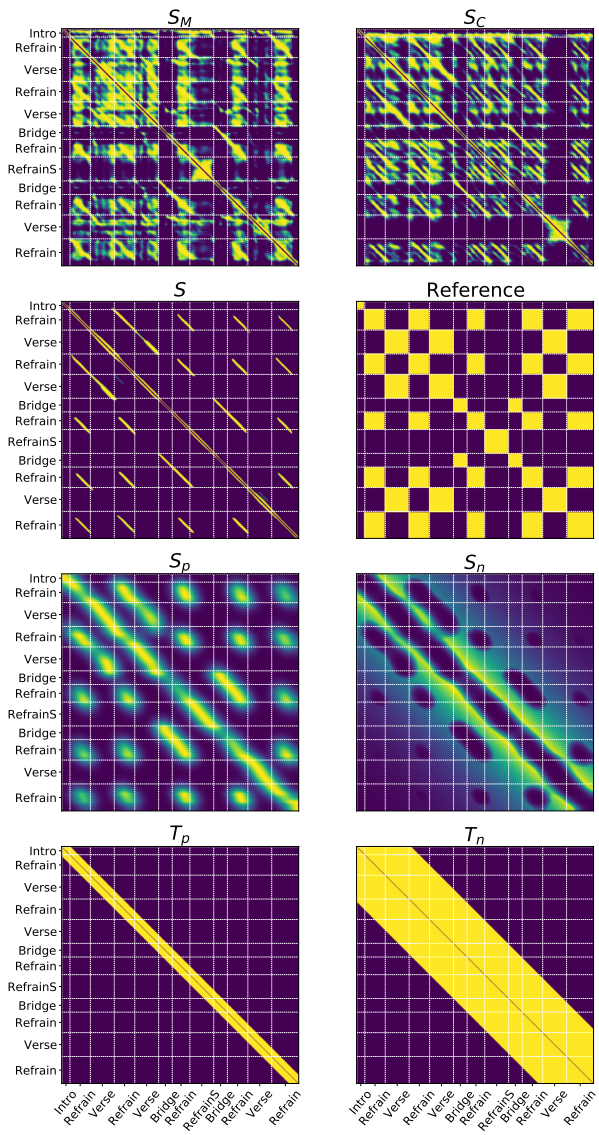


Figure 1. Example of the self-similarity approximation process for *The Beatles — Baby's In Black*. Top to bottom, left to right: self-similarity lag-matrices obtained using MFCC (S_M), chroma features (S_C), median filtered combination (S), reference self-similarity matrix (supervised scenario), positive matrix (S_p), negative matrix (S_n), positive (T_p) and negative (T_n) sampling matrices using *temporal sampling* [12]. White dotted lines denote boundary instants.

This has the effect of blurring the regions of S around its diagonal stripes, which approximates the width of the corresponding musical segments in a more uniform manner than directly using the unfiltered matrix S. The size of the kernel K logically impacts the extent to which this dilation is performed. It was found that setting $K = 8$ (beats) provided a good balance between the amount of dilation and its alignment with segment boundaries (third row, left column of Figure 1), as it blurs repetitions over 2 bars when songs follow a $4/4$ time signature [1].

[1] Such value might induce a bias towards specific western music genres. This parameter should ideally be adapted to each training track.

3.4 Negative mining

While the matrix S_p guides the selection of positive examples for any frame of the input track, the triplet loss requires to find a third point with a different label, called negative example. In our case, such example should belong to a different musical section, which could be easily solved by searching for the least similar frames from the anchor (*i.e.* using the matrix $S_n = 1 - S_p$ for sampling). However, doing so is likely to result in trivial triplets where the relative difference between $d(f(x_a), f(x_p))$ and $d(f(x_a), f(x_n))$ from Equation (1) might already be larger than the margin δ, thus, yielding small gradients that prevent the network from learning features that are discriminative enough [29]. Instead, we enforce negative examples to be chosen close to the anchor's location while still avoiding homogeneous regions indicated by the positive matrix S_p. To this end, the negative sampling matrix S_n is obtained by applying an exponential decay to $1 - S_p$ such that:

$$S_n(i,j) = (1 - S_p(i,j))e^{-\lambda \max\left(\frac{|i-j|}{N}, S_p(i,j)\right)}, \quad (8)$$

where $\lambda > 0$ is a parameter that defines the strength of the smoothing. As a consequence, components near the main diagonal of S_n (third row, right column of Figure 1) receive greater values than those close the opposite edges, thus favoring frames located within consecutive segments of that of the anchor.

The final sampling process works as follows: given an anchor point i_a chosen among the N frames of the input track, the weight attributed to a certain index i_k when sampling the positive example follows the discrete probability distribution defined by the a-th row of S_p, such that $Pr(I = i_k) = S_p(a,k)$. The negative example is chosen in a similar fashion with the matrix S_n, such that $Pr(I = i_k) = S_n(a,k)$.

4. EXPERIMENTAL SETTING

This section details the experiments performed to assess the efficacy of the proposed triplet mining method. First, a preliminary evaluation of the triplets generated is done against structural annotations from a commonly used dataset for music structure analysis. Secondly, we train two separate convolutional neural networks using triplets obtained by *temporal sampling* and those from our method. The obtained embeddings are fed as input to a downstream music segmentation algorithm and performance on both boundary detection and structural grouping is measured. Finally, to gain more insight on the quality of the triplets generated, training is performed on different fractions of the unlabelled training dataset.

4.1 Datasets

Since this work falls under the scope of unsupervised learning, a non annotated external audio collection is used for training. It is composed of $20,000$ tracks, spanning various musical genres such as rock, popular, rap, jazz, electronic or classical. These were retrieved from publicly available playlists and the audio obtained from YOUTUBE. Care has been taken to discard any track from this external collection also present in one of the following testing datasets. Training is separately done on 10%, 50% and 100% of this dataset.

SALAMI: the Structural Annotations for Large Amounts of Music Information (SALAMI) [30] contains $1,359$ tracks ranging from classical, jazz, popular to world and live music. For evaluation, we use the *upper* annotations of a subset of 884 songs labelled by two different annotators.

JSD: the Jazz Structure Dataset [31] gathers 340 jazz recordings provided with two-level annotations: the chorus level (a full cycle of the harmonic schema, which is the annotation level used for evaluation) and a solo level, consisting of one more choruses. These annotations follow the common jazz structure schema that includes the introduction of the main melody (theme), followed by alternating solos from the different musicians and a final return towards the main theme at the end of the track.

4.2 Evaluation metrics

Common evaluation metrics for automatic structure analysis are employed throughout our experiments. For boundary detection, we report the F-measure [2] of the trimmed [3] 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 frame pairwise clustering [21] (PFC), which gives another view on flat segmentation performance in terms of frame-wise section assignment. Additionally, the normalized conditional entropy score (NCE) [33] is also calculated, in order to indicate from a probabilistic perspective the amount of information shared between predicted label distributions and their corresponding reference annotations. In the case where the test dataset has more than one annotator, the best score across annotators is kept, as the goal of the evaluation process is to measure how close to human ground-truth the predicted segmentations are. The average score obtained per metric is reported and the statistical significance is assessed using a paired-sample T-test with $p < 0.05$.

4.3 Implementation details

Input features: All tracks are resampled at 22.05 kHz. We use log-scaled Mel-spectrograms as input to the deep network, with a window and hop size of 2048 and 256 respectively. We compute 60 Mel-band coefficients per frame. Feature patches are composed of 512 frames ($\simeq 5.94$s) and centered at each detected beat location.

Mining parameters: Chroma features are extracted using a minimum frequency of 27.5 Hz over 8 octaves. 20 MFCC coefficients are calculated per frame and the very first one is discarded. Both are calculated with the librosa

[2] All evaluations are done using the mir_eval package [32].

[3] 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.

library [34]. The features are encoded into time-delay representations using context values of $m = 16$ and $m = 8$ beats respectively. The parameters α and β of the sigmoid filtering step are set to 60 and 0.85. We give equal weight to each feature by setting the $\gamma = 0.5$ in Equation (6). Finally, the negative matrix S_n is calculated with a smoothing parameter $\lambda = 5$. These parameters were found using simple grid searches and visual inspections of the obtained self-similarity matrices.

Network architecture: The encoder consists of a convolutional neural network composed of 3 convolutional layers, each followed by a max-pooling layer and Elu activation, and two fully-connected layers comprising 128 units with Elu and linear activations respectively. All convolutional layers use a kernel size of size $(3, 3)$ with 32 filters each. The output embeddings are ℓ_2-normalized before calculating the triplet loss. The models are implemented[4] with Pytorch 1.7.1 [35]. The SGD optimizer with 10^{-4} weight decay and 0.9 momentum is used, the models are trained for a maximum of 200 epochs, where each batch is composed of 256 triplets obtained from one single track. Similar to previous work [12], the margin parameter δ is set to 0.1 and the embedding dimension to $d = 128$.

Downstream segmentation: For all experiments, the embeddings returned by each model are fed as input to spectral clustering [24], as this algorithm jointly performs both boundary detection and structural grouping in an unsupervised manner and has proven to be efficient in previous studies [13, 14]. This also allows one to compare the influence of each of the tested representations into a single unified framework. The original algorithm takes two distinct beat-synchronized audio features as input (MFCC and CQT). We consider this method as a second baseline which we denote as LSD (Laplacian Structural Decomposition). However in our case, it is directly applied to the self-similarity S_p of each track. When this algorithm is combined with deep representations, we simply replace both input features by the embedding matrix. Finally, because spectral clustering outputs multiple levels of segmentation, only the one maximizing the considered metric is reported (HR.5F and HR3F for boundary detection, PFC and NCE for structural grouping).

5. RESULTS

5.1 Preliminary evaluation

We generate 256 triplets per track contained in the SALAMI dataset and report the proportions of true positives, true negatives and correct triplets in Table 1. For comparison purposes, we also provide a random baseline, where each anchor, positive and negative example is uniformly sampled over the whole track. The sampling method proposed significantly improves the selection of negative examples compared to the *temporal sampling* approach. However, random negative sampling performs better than our approach. This was to be expected, since the latter samples negatives over the whole track while our method greatly narrows down the number of probable candidates (see Equation (8)). Conversely, the *temporal sampling* returns a higher proportion of true positives than ours, since these are sampled in a relatively short time window around their respective anchor, thus omitting any section repetition occurring inside the input track. All in all, our approach returns a much higher proportion of correct triplets than either of the comparison strategies while guaranteeing that positive examples are located within the right musical sections and the negative within a relatively short time window around their anchor's.

Sampling	TP	TN	CT
Random	$.401 \pm .22$	$.595 \pm .21$	$.194 \pm .06$
Temporal [12]	$.886 \pm .32$	$.398 \pm .49$	$.325 \pm .47$
Ours	$.800 \pm .40$	$\mathbf{.583 \pm .49}$	$\mathbf{.432 \pm .50}$

Table 1. Triplet mining results on *upper* annotation level of SALAMI dataset. TP, TN, CT: proportions of true positives, true negatives and correct triplets respectively. Results highlighted in bold denote statistically significant improvements over *temporal sampling* according to a paired-sample T-test with $p < 0.05$.

5.2 Segmentation and structural grouping

Table 2 shows the performance of our approach against *temporal sampling* on the *upper* annotations of the SALAMI dataset. Regardless of the amount of training data, our method constantly improves both boundary detection and structural grouping in a significant manner. It is also interesting to see that such improvement is already achieved when the proposed method uses only 10% of the training dataset. This corroborates the results from Section 5.1, showing that improving the triplets quality provides a cleaner training signal and makes learning more efficient.

Method (Split)	HR.5F	HR3F	PFC	NCE
LSD	.195	.486	.707	.682
Temp. (10%) [12]	.280	.665	.770	.677
Ours (10%)	**.291**	**.676**	**.777**	**.691**
Temp. (50%) [12]	.288	.671	.773	.678
Ours (50%)	**.296**	**.682**	**.778**	**.690**
Temp. (100%) [12]	.284	.670	.773	.678
Ours (100%)	**.297**	**.683**	**.781**	**.694**

Table 2. Flat segmentation results on SALAMI (*upper* annotations). Results in bold denote statistically significant improvement over *temporal sampling* on same split (denoted as *Temp.*).

From a more qualitative perspective, Figure 2 shows examples of self-similarity matrices derived from the embeddings trained with *temporal sampling* and our method. In the latter case, consecutive musical sections are better discriminated (clearer block structures on the main diagonal). Section repetitions (visible as diagonal stripes and off-diagonal blocks) are more straightforward to recog-

[4] Code: `github.com/morgan76/Triplet_Mining`

nize, especially those with relatively small durations (sections A, B or D).

Method (Split)	HR.5F	HR3F	PFC	NCE
LSD	.195	.486	.707	.682
Temp. (10%) [12]	.221	.568	.739	.745
Ours (10%)	.219	**.586**	.744	.749
Temp. (50%) [12]	.243	.586	.763	.766
Ours (50%)	.222	.583	.755	.758
Temp. (100%) [12]	.229	.590	.766	.767
Ours (100%)	.225	.592	.754	.760

Table 3. Flat segmentation results on JSD (*chorus* annotation level). Results in bold denote statistically significant improvement over *temporal sampling* (denoted as *Temp.*) on same split.

Results on the JSD dataset are given in Table 3. Here, the improvements made are not as consistent. However, when using only 10% of the training dataset, the performance of our approach remains within the same range than that of the baseline when trained on larger splits. Compared to the results obtained on SALAMI, the small improvements made here can be associated with the way structure is defined in terms of feature similarity in jazz.

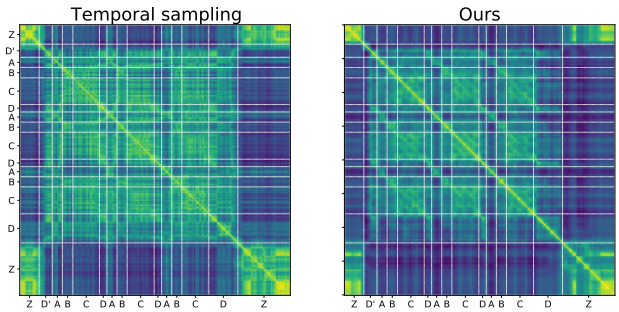


Figure 2. Example of self-similarity matrices for the track SALAMI 1380. Left: encoder trained with *temporal sampling*. Right: encoder trained using the proposed triplet mining method. White dotted lines denote boundary instants.

5.3 Discussion on mining parameters

Impact on triplet selection: The sampling parameters could further be tuned to improve performance. More specifically, the audio descriptors employed at the first stage and their combination could be adapted to the training data in order to better emphasize more specific aspects of the audio. For example, some music genres such as pop music or rock generally rely on the repetition of certain chord progressions [1]. However, introducing a degree of timbral homogeneity allows for differentiating two sections that are semantically similar, such as in the example from Figure 1, 'refrain' and 'refrain-Solo'. Putting more emphasis on timbral features might be better adapted to music genres such as jazz, where structure is highly influenced by changes in soloists. As an example, Figure 3 displays the positive sampling matrices obtained when varying the γ parameter from Equation (6). It is clear to see that favoring timbral similarity helps better approximating segment transitions and mutual dissimilarities between the successive solos of saxophone, piano and guitar.

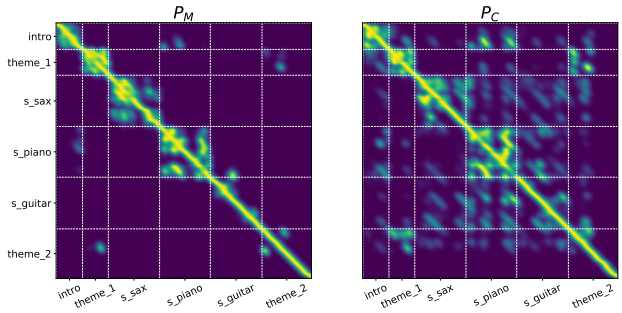


Figure 3. Example of positive sampling matrices for *Michael Brecker — Song for Bilbao*. Left: emphasis on timbral content ($\gamma = 0.9$). Right: emphasis on harmonic content ($\gamma = 0.1$). White dotted lines denote boundary instants.

Impact on segmentation: To illustrate how the balance between harmonic and timbral features impacts the final segmentation, the encoder is trained on the 10% and 50% splits of the dataset with $\gamma = 0.9$, thus putting a stronger emphasis on the MFCC-based similarity at the triplet mining stage. All other parameters are kept to their initial values described in Section 4.3. The segmentation results summarized in Table 4 show that the choice of the parameter γ does impact the training process. In this case, putting more weight on timbral information seems to make the representations more sensitive to timbral changes and improves boundary detection (HR3F) in a significant manner compared to *temporal sampling*.

Method (Split)	HR.5F	HR3F	PFC	NCE
Temp. (10%) [12]	.221	.568	.739	.745
Ours (10%, $\gamma = 0.9$)	.223	**.585**	.743	.750
Temp. (50%) [12]	.243	.586	.763	.766
Ours (50%, $\gamma = 0.9$)	.234	**.607**	.769	.772

Table 4. Flat segmentation results on JSD (*chorus* annotation level) with emphasis on timbral features ($\gamma = 0.9$). Results in bold denote statistically significant improvement over *temporal sampling* (denoted as *Temp.*) on same split.

6. CONCLUSION

This work introduced a repetition-based triplet mining mechanism to learn efficient audio representations prior to music segmentation, which can significantly improve both boundary detection and structural grouping, while needing less data than previous similar methods. Complementary experiments demonstrate that this sampling process can be further adapted to the final type of segmentation desired by either emphasizing harmonic or timbral information from the input track.

7. ACKNOWLEDGEMENTS

This work was performed using HPC resources from GENCI-IDRIS (Grant 2022-AD011013255R1).

8. 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*, vol. 3, no. 1, 2020.

[2] S. Dai, Z. Jin, C. Gomes, and R. B. Dannenberg, "Controllable deep melody generation via hierarchical music structure representation," *ISMIR*, 2021.

[3] A. Bozzon, G. Prandi, G. Valenzise, M. Tagliasacchi *et al.*, "A music recommendation system based on semantic audio segments similarity," *Proceeding of Internet and Multimedia Systems and Applications-2008*, pp. 182–187, 2008.

[4] J. P. Bello, "Measuring structural similarity in music," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 7, pp. 2013–2025, 2011.

[5] S. Dai, H. Zhang, and R. B. Dannenberg, "Automatic analysis and influence of hierarchical structure on melody, rhythm and harmony in popular music," 2020.

[6] M. Fuentes, B. McFee, H. C. Crayencour, S. Essid, and J. P. Bello, "A music structure informed downbeat tracking system using skip-chain conditional random fields and deep learning," in *ICASSP*, 2019.

[7] M. Mauch, K. C. Noland, and S. Dixon, "Using musical structure to enhance automatic chord transcription." in *ISMIR*, 2009.

[8] J. Paulus, M. Müller, and A. Klapuri, "State of the art report: Audio-based music structure analysis." in *ISMIR*, 2010.

[9] M. Goto, "A chorus section detection method for musical audio signals and its application to a music listening station," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 5, pp. 1783–1794, 2006.

[10] R. B. Dannenberg and N. Hu, "Pattern discovery techniques for music audio," *Journal of New Music Research*, vol. 32, no. 2, pp. 153–163, 2003.

[11] M. Müller and F. Kurth, "Towards structural analysis of audio recordings in the presence of musical variations," *EURASIP Journal on Advances in Signal Processing*, vol. 2007, pp. 1–18, 2006.

[12] M. C. McCallum, "Unsupervised learning of deep features for music segmentation," in *ICASSP*, 2019.

[13] J.-C. Wang, J. B. L. Smith, W. T. Lu, and X. Song, "Supervised metric learning for music structure features," in *ISMIR*, 2021.

[14] J. Salamon, O. Nieto, and N. J. Bryan, "Deep embeddings and section fusion improve music segmentation," in *ISMIR*, 2021.

[15] M. Buisson, B. McFee, S. Essid, and H. C. Crayencour, "Learning multi-level representations for hierarchical music structure analysis," in *ISMIR*, 2022.

[16] J. B. Smith, C.-H. Chuan, and E. Chew, "Audio properties of perceived boundaries in music," *IEEE transactions on multimedia*, vol. 16, no. 5, pp. 1219–1228, 2014.

[17] F. Kaiser and G. Peeters, "A simple fusion method of state and sequence segmentation for music structure discovery." in *ISMIR*, 2013.

[18] B. McFee and D. P. Ellis, "Learning to segment songs with ordinal linear discriminant analysis," in *ICASSP*, 2014.

[19] K. Ullrich, J. Schlüter, and T. Grill, "Boundary detection in music structure analysis using convolutional neural networks." in *ISMIR*, 2014.

[20] J. Paulus and A. Klapuri, "Music structure analysis using a probabilistic fitness measure and a greedy search algorithm," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 6, pp. 1159–1170, 2009.

[21] M. Levy and M. Sandler, "Structural segmentation of musical audio by constrained clustering," *IEEE transactions on audio, speech, and language processing*, vol. 16, no. 2, pp. 318–326, 2008.

[22] J. Serra, 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.

[23] X. Wang, X. Han, W. Huang, D. Dong, and M. R. Scott, "Multi-similarity loss with general pair weighting for deep metric learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 5022–5030.

[24] B. McFee and D. Ellis, "Analyzing song structure with spectral clustering." in *ISMIR*, 2014.

[25] G. Shibata, R. Nishikimi, and K. Yoshii, "Music structure analysis based on an lstm-hsmm hybrid model." in *ISMIR*, 2020.

[26] F. Korzeniowski, S. Böck, and G. Widmer, "Probabilistic extraction of beat positions from a beat activation function." in *ISMIR*, 2014.

[27] 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.

[28] H. Kantz and T. Schreiber, *Nonlinear time series analysis*. Cambridge university press, 2004, vol. 7.

[29] A. Hermans, L. Beyer, and B. Leibe, "In defense of the triplet loss for person re-identification," *arXiv preprint arXiv:1703.07737*, 2017.

[30] 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.

[31] S. Balke, J. Reck, C. Weiß, J. Abeßer, and M. Müller, "Jsd: A dataset for structure analysis in jazz music," *Transactions of the International Society for Music Information Retrieval*, vol. 5, no. 1, 2022.

[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 *ISMIR*, 2014.

[33] H. M. Lukashevich, "Towards quantitative measures of evaluating song segmentation." in *ISMIR*, 2008.

[34] 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.

[35] 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," *Advances in neural information processing systems*, 2019.

PREDICTING MUSIC HIERARCHIES WITH A GRAPH-BASED NEURAL DECODER

Francesco Foscarin[1] **Daniel Harasim**[2] **Gerhard Widmer**[1]
[1] Institute of Computational Perception, Johannes Kepler University Linz, Austria
[2] Independent Researcher
francesco.foscarin@jku.at

ABSTRACT

This paper describes a data-driven framework to parse musical sequences into dependency trees, which are hierarchical structures used in music cognition research and music analysis. The parsing involves two steps. First, the input sequence is passed through a transformer encoder to enrich it with contextual information. Then, a classifier filters the graph of all possible dependency arcs to produce the dependency tree. One major benefit of this system is that it can be easily integrated into modern deep-learning pipelines. Moreover, since it does not rely on any particular symbolic grammar, it can consider multiple musical features simultaneously, make use of sequential context information, and produce partial results for noisy inputs. We test our approach on two datasets of musical trees – time-span trees of monophonic note sequences and harmonic trees of jazz chord sequences – and show that our approach outperforms previous methods. [1]

1. INTRODUCTION

Tree-like representations are a powerful tool in many approaches to music analysis, such as Schenkerian Theory and the Generative Theory of Tonal Music (GTTM). In the Music Information Retrieval (MIR) literature, we find tree models of melodies [1–4], chord progressions [5–8], and rhythm [9–13]. Parallels between aspects of music and language are often drawn, as these have similar hierarchical properties and their underlying cognitive mechanisms could be closely related [14]. However, with a few exceptions, such as instrument grouping and metrical information in scores, music is generally encoded sequentially without explicit information about its hierarchical organisation. The task of creating such hierarchies from a sequential representation is called *parsing* and it is an active object of study in the MIR community [3, 7, 11, 15].

Current parsing approaches are based on *generative grammars*, typically context-free-grammars (CFG) or similar related mechanisms, which can be fundamentally seen as a set of expansion rules generating a tree from the top (the root) to the elements that compose the sequence (the leaves). Grammar rules can be enriched with a probability model that permits the ranking of different parses by plausibility. When a grammar is available, parsing can be achieved with grammar-based parsing algorithms, typically variants of the Cocke–Younger–Kasami (CYK) algorithm [16]. While the grammar rules are most often built by hand, by relying on musicologists' knowledge, the probabilities can be learnt from data if sufficient amounts of musical sequences with ground-truth tree annotations are available. The grammar approach has the strong advantage of leveraging an interpretable and cognitively plausible mechanism. Still, it has the following limitations: it is hard to achieve robustness against noisy data, which can cause a complete failure with no output in case the sequence cannot be produced by the grammar rules; it requires a high degree of domain knowledge; it is challenging to account for multiple musical dimensions in a single grammar rule; and parsing is usually so slow for long sequences that heavy pruning is necessary (CYK-parsing complexity is cubic in the length of the sequence, parallelisation does not help much, and there is no active research in developing dedicated hardware).

Inspired by recent research in the field of natural language processing (NLP), we propose a novel, *grammar-less* approach that requires little domain knowledge (only for the feature extraction phase), can easily consider multiple musical features and sequential information, produces partial results for noisy input, and is potentially scalable to longer sequences and larger datasets (since its components are proven to succeed in such scenarios). Our system works by predicting *dependency trees* which consist of dependency arcs between the input sequence elements. Such a structure can be used as-is or later be converted into *constituent trees* which are typically used to model music hierarchies (see Figure 1). The probability of each dependency arc is predicted in parallel (i.e., without considering other dependencies during prediction) by leveraging the rich contextual information produced by a transformer encoding of the input sequence. This set of probabilities is then run through a post-processing algorithm to ensure a valid tree structure (i.e., no cycles of dependency arcs).

We pair our Music Dependency Parser MuDeP with a

[1] All our code and data are publicly available at https://github.com/fosfrancesco/musicparser

© F. Foscarin, D. Harasim, and G. Widmer. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).
Attribution: F. Foscarin, D. Harasim, and G. Widmer, "Predicting Music Hierarchies With a Graph-Based Neural Decoder", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

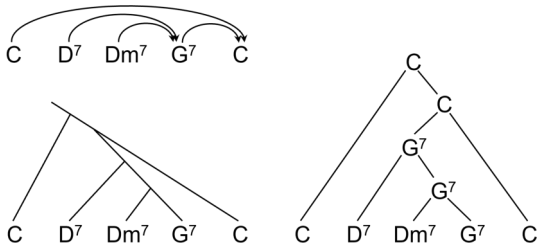


Figure 1. The tree harmonic analysis of the A Section of "Take the A Train" in three different representations. Top: dependency tree, Left: GTTM-style constituent tree. Right: CFG-style constituent tree.

procedure that enables its usage from constituent trees, and test it on two tree datasets: the time-span treebank from the GTTM database [17], which expresses subordinate relations between notes in monophonic melodies; and the Jazz Harmony Treebank (JHT), a set of harmonic analyses for chord sequences [18]. We compare the results of our system with the best-performing available approaches and obtain new state-of-the-art results.

2. RELATED WORK

Music Trees and Music Parsing. Trees of musical sequences have traditionally been notated as constituent trees [1–3, 5–13, 19], with few exceptions, such as the usage of a dependency-based evaluation metric [20], and the computation of pairwise voice dependencies [4, 21].

A system for parsing jazz chord sequences into harmonic analyses has been proposed by Harasim et al. [7] and later evaluated on a larger dataset [20]. We compare our results to this approach below. Automatic grammar-based parsing of time-span GTTM trees has been attempted by Hamanaka et al. [22, 23] and Nakamura et al. [2]. The latter obtained comparable results with an approach that doesn't require manual parameter tuning, and we compare our system with it. More recently, deep-learning-based approaches were also proposed [3, 24, 25] but the first two focus only on GTTM metrical and grouping information, and the latter focus mainly on evaluating the usage of time-span trees for melodic morphing and we could not reproduce their results.

Natural Language Parsing. Our model architecture is inspired by the graph-based dependency parser of Dozat and Manning [26, 27]. This model, extended with second-order dependency predictions [28] and pretrained language models [29], is still the state of the art for NLP sentence parsing [30]. Still, we make some substantial changes: the embedding layer is adapted to work from musical input, the encoder is a transformer instead of an LSTM, and, instead of the bilinear layer for arc prediction, we use a linear layer. All these choices are motivated by ablation studies.

3. TREE FORMATS FOR MUSIC ANALYSES

In this section, we detail the types of tree used in this paper, highlight their differences, and propose algorithms to translate between them.

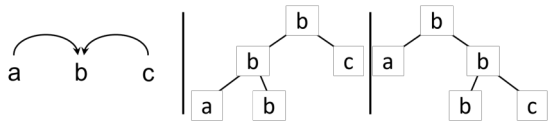


Figure 2. A dependency tree with double-sided dependencies (left). It corresponds to two possible constituent trees (middle and right).

3.1 Constituent vs Dependency Trees

A tree can be defined recursively as a node with an arbitrary number (including 0) of children that are also trees. The node that is not a child of another node in the tree is called *root*, the nodes that do not have children are called *leaves*, and the remaining nodes are called *internal nodes*. When a tree is used to model some relations of the elements of a sequence there are two possible configurations: *dependency trees*, where each node (leaf, internal, and root) represents one and only one element of the sequence; and *constituent trees* where all elements of the sequence are represented in the leaves, and root and internal nodes represent nested groupings of such elements.

Among the constituent trees there exist different representations. The bottom part of Figure 1 shows the two kinds we consider in this paper: the one introduced by Lerdahl and Jackendoff [31] in their Generative Theory of Tonal Music (GTTM), and the one built from the Context-Free-Grammar (CFG) of jazz harmony by Harasim et al. [7]. The two representations convey almost the same information: they are both binary trees (i.e., every node has either 0 or 2 children), the internal nodes are denoted by line intersections on the first, and by explicit labels on the second; they both specify an order of importance among the children (i.e., the choice of a *primary* and *secondary* child) by the straight line continuation, or by labelling the node with the label of the primary child. However, this latter mechanism cannot differentiate between primary and secondary when both children have the same label; therefore, the GTTM representation is slightly more informative.

Our approach does not directly treat constituent trees but considers dependency trees. Each child in such a tree is called *dependent*, and the node of which it is a child is called the *head*. Dependency trees can represent the same information as the binary constituent trees described above. Indeed, a dependent-head arc is equivalent to a head-labelled constituent node with two children: the primary is again the head, and the secondary is the dependent. There is only one ambiguity: the dependency tree does not encode a splitting order in the case of *double-sided dependencies*, a configuration in which one head has dependents on both sides. This makes the dependency-to-constituent transformation not unique (see Figure 2). This configuration is never present in our datasets (i.e., the root is always the left-most or right-most element in the sequence) thus we don't handle it. For more general datasets, one could add a binary classifier that predicts the splitting order.

The dependency trees built from the constituent trees

are *projective*, i.e., for all their arcs $x_{\text{dep}} \rightarrow x_{\text{head}}$, there is a path from the head to every element x that lies between the head and the dependent in the sentence [32]. This means that there are no "crossing arcs", e.g., $x_1 \rightarrow x_3, x_2 \rightarrow x_4$.

Before proceeding with the paper, we introduce some notation we will use in the next sections. We denote the sequence that constitutes the input of our system as $x = [x_1, \ldots, x_\lambda]$, where λ is the sequence's length. We represent the dependency tree over x as the set of dependent-head [2] indices that corresponds to each arc $x_{\text{dep}} \rightarrow x_{\text{head}}$:

$$y = \{(\text{dep}, \text{head}) \mid \text{dep}, \text{head} \in [1, \ldots, \lambda]\} \quad (1)$$

3.2 Tree Conversion Algorithms

Since the ground-truth annotations in our datasets are constituent trees, we translate them into dependency trees for training. We also translate tree predictions back to constituent trees to run constituent-based evaluation metrics, and when we are interested in using such a representation as input for further applications. We assume our constituent trees to be binary trees and not contain double-sided dependencies. For simplicity, we consider CFG-style constituent trees with labels in their internal nodes.

3.2.1 Dependency to Constituent Tree

Existing NLP implementations of this transformation are unnecessarily complicated for our scenario because they consider compound node labels and double-sided dependencies [33]. Instead, we present a recursive top-down algorithm which yields a unique constituent solution for every single-sided dependency tree.

The algorithm takes a fully formed dependency tree and starts with the root of the (to-be-built) constituent tree. At each step, it removes one dependency and adds two new constituent nodes. The recursive function takes as input a dependency tree node and a constituent tree node, both labelled with the same sequence element. The constituent node gets assigned two children: the primary is labelled with the element of the input nodes, and the secondary is labelled with the dependent that is further away in the sequence. The choice of which is the left and the right child respects their label position in the sequence. The considered dependency is removed from the tree and the recursive function is called two times, once for each constituent child (with the corresponding dependency node). The process stops when the dependency tree node has no dependents.

3.2.2 Constituent to Dependency Tree

This algorithm was used in the literature (e.g., [18]). It starts from a fully formed constituent tree and a dependency tree without any dependency arcs, consisting only of the nodes labelled with sequence elements. The algorithm groups all internal tree nodes with their primary child (which all have the same label) and uses all secondary child relations originating from each group to create dependency arcs between the group label and the secondary child label.

[2] We indicate dependency arcs as arrows pointing in the direction of the head. Note that in other (NLP) papers, the opposite convention is used.

4. PARSING TECHNIQUE

Our goal is to predict a dependency tree y for a given musical sequence x. Our pipeline consists of three steps: feature extraction from x; prediction of dependency relations; and postprocessing to ensure that the output is a valid tree structure. In the training phase, the output (before postprocessing) is compared with the ground truth dependency tree and a loss is computed to update the model parameters via backpropagation.

4.1 Feature Extraction

For each input element, $x_i \in x$, we produce three groups of features. The first is a "static" description of the element (i.e., without any temporal information), the second encodes the element's duration, and the third encodes the element's metrical position, i.e., its position in the measure relative to the hierarchy induced by the time signature. The static description is built differently for chords and notes, while the other two are independent of the input type. Note that, due to our model architecture (see next section), we need categorical features and it is not primarily important to keep their number small or to have them ordered.

For note sequences, the *static description* of each element is a single integer corresponding to either the MIDI pitch of the element in $[0, \ldots, 127]$ if it is a note or with the value 128 if the element is a rest. For chord sequences, we use three integers. The first in $[0, \ldots, 11]$ encodes the pitch-class of the chord root. The second in $[0, \ldots, 5]$ specifies the basic form of the chord among major, minor, augmented, half-diminished, diminished, and suspended (sus). The last in $[0, 1, 2]$ denotes a chord extension among 6, minor 7, or major 7. The chord labels were simplified by the author of the dataset to only include these extensions, but in a more general scenario, a larger set of integers could be used. The chord sequences do not contain rests.

We represent the *durations of the elements* with discrete values normalised by the duration of the measure. We pre-collect the list of all durations occurring in the dataset and encode each element's duration as an index on that list. For the GTTM dataset, this would be an integer in $[0, \ldots, 44]$, while for the JHB dataset, it is an integer in $[0, \ldots, 5]$. The number of possibilities is very different, since the temporal position of chords follows much simpler rules, mostly occurring only at the beginning or in the middle of a bar for simple time signatures and at three bar positions for compound time signatures. For tied notes, we consider a single note with the total duration, and notes can last more than one measure. This is different from the annotations in the JHT in which each measure opens a new chord symbol, even if the same chord is repeated in consecutive measures.

To represent the *metrical position*, we use an inverse measure of metrical strength, encoded with a single integer in $[0, \ldots, 5]$. This integer is computed as a function of the normalised temporal position in the measure $t \in [0, 1[$, and the time signature numerator. Each time-signature numerator is associated with a template of metrical divisions m, as proposed by Foscarin [10] and here extended to more time signatures. For example, a time signature with a nu-

Level \ Time Sig	2/*	3/*	4/*	6/*	8/*	9/*	12/*
0	1	1	1	1	1	1	1
1	2	3	2	2	2	3	2
2	2	2	2	3	2	3	2
3	2	2	2	2	2	2	3
4	2	2	2	2	2	2	2

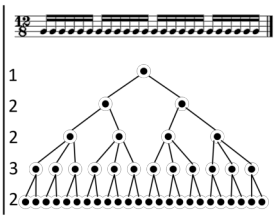


Figure 3. Left: metrical divisions m for different time signature numerators. Right: visualisation of metrical divisions for a measure with time signature 12/8.

merator 12 (e.g., $12/8$ or $12/4$) will have metrical divisions $m = [1, 2, 2, 3, 2]$, i.e., the whole measure at level 0 is divided into two parts at level 1, each resulting part is divided in 2 at level 2, then 3 at level 3, and 2 at level 4. Table 3 reports metrical divisions for all numerators we consider.

Each level l in the metrical division defines a temporal grid with step $\delta_l = 1/\prod_{l=0}^{l} m_l$, and the *inverse metrical strength* is defined as the lowest level for which the note position falls on the temporal grid, $\min_l(l \mid t/\delta_l \in \mathbb{N})$. For example, a time signature $6/8$ defines grids with steps $[1, \frac{1}{2}, \frac{1}{6}, \frac{1}{12}, \frac{1}{24}]$, and the notes of the measure |♩♩♪♩.| will have normalised temporal position $[0, \frac{2}{6}, \frac{1}{2}]$ and inverse metrical strength $[0, 2, 1]$. If the note doesn't align with any temporal grid, then its inverse metrical strength is the maximum, 5 in our settings. Using metrical strength as input to our system may seem overly complicated. However, given the small size of our datasets and the high variety of time signatures, we need a mechanism to encode metrical information generalisable across different time signatures. It is to be expected that with a larger dataset size, this feature could be discarded, as the model could learn similar information from the list of notes with duration.

The feature extraction process lets us build the input matrix $\mathcal{X} \in \mathbb{N}^{\lambda \times \phi}$ where λ is the sequence length, and ϕ is the number of features for each element: 3 for the GTTM dataset, and 5 for the JHT dataset. Before moving on, it has to be noted that there exist other more general ways of transforming symbolic music into convenient inputs for deep learning models, notably the tokenisation techniques, e.g., [34, 35] inspired by NLP research. However, given the small dimension of our dataset and the fact that our melodies are strictly monophonic, we prefer to use a more compact, ad-hoc input representation. Our parsing framework remains general and usable with other techniques.

4.2 Model

Our model consists of two parts: an encoder and an arc predictor (see Figure 4). The goal of the encoder is to enrich the input features $\mathcal{X}$ with contextual information. The arc predictor uses the enriched sequence features to predict whether each possible pair of elements in the input sequence should be connected by a dependency arc.

The first part of the encoder is an embedding layer, a learnable look-up table which maps our collection of categorical features (each integer) to points in a continuous multidimensional space. Specifically, we use ϕ embedding

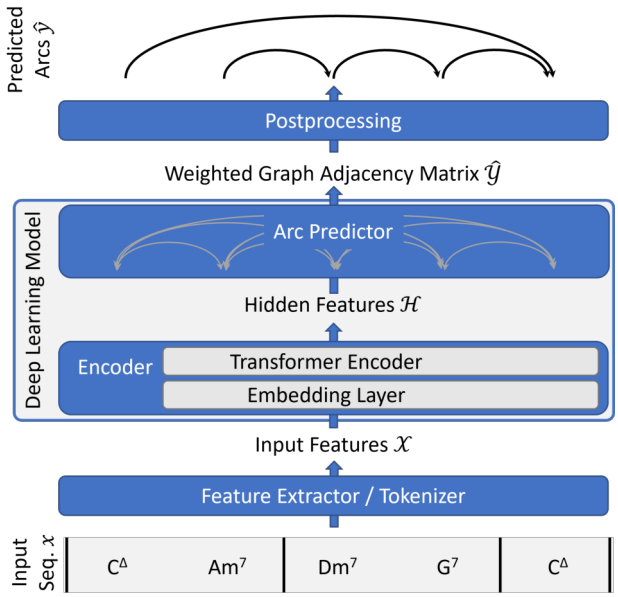


Figure 4. Architecture of our model. The input displayed is an example of a chord sequence, but the same architecture is used for note sequences.

layers (one for each input feature), which work independently, and map all values that the feature can have to a vector of a fixed embedding dimension. All vectors are then summed together to obtain a unique representation while keeping the input size small (see [36] for an explanation of why summing is better than concatenating). After the embedding layer, we have the encoder part of a transformer [37] with relative position representations [38]. It outputs a matrix with the same number of rows as the input matrix $\mathcal{X}$ (one for each sequence element) but with a (possibly) different number of hidden-feature columns h. Onto this, we concatenate a new learnable single row that acts as the head of the root node. The result is a new matrix $\mathcal{H} \in \mathbb{Q}^{(\lambda+1)\times h}$.

The arc predictor part of our model is a multilayer perceptron (MLP) that performs the binary classification task of deciding whether each pair $(x_{\text{head}}, x_{\text{dep}})$ in the Cartesian product of the input elements, i.e., $\{(\text{head}, \text{dep}) \mid \forall\ \text{head}, \text{dep} \in [1, \ldots, \lambda]\}$, should be connected by a dependency arc. Depending on the input representation and the specific task we are targeting, there may be some pairs that are not connectable by a dependency arc, for example, pairs where head $=$ dep. For the GTTM input, pairs for which at least one element is a rest are also not connectable. Therefore, the binary classification is performed only on a subset of all pairs Λ that we call *potential arcs*. For every potential arc $(x_{\text{dep}}, x_{\text{head}})$, we predict the probability $\hat{y}_{\text{dep,head}}$ of a dependency arc by concatenating the two rows of $\mathcal{H}$ that correspond to the head's and the dependent's index into a single vector of length $2h$ and giving it as input to the MLP. We concatenate the two inputs instead of summing or multiplying them because our arcs are directed, so we need to preserve the order when aggregating the two embeddings. Moreover, despite the bilinear layer being a major selling

point of Dozat's paper [26], we find that the concatenation approach yields better results. We can collect the output for all potential arcs into a *weighted graph-adjacency matrix* $\hat{Y}$, which is a $\lambda \times \lambda$ matrix with entries $\hat{y}_{\text{head,dep}}$ at the corresponding indices. We assign a probability 0 to the matrix entries that correspond to arcs $\notin \Lambda$.

4.3 Training Loss

In the training phase, we use the sum of the binary-cross-entropy (BCE) loss and the (multiclass) cross-entropy (CE) loss. The BCE loss is computed independently for each potential arc and measures the difference between the ground-truth label (0 or 1) and the predicted probability. We also use the CE loss because our problem can be framed as a multiclass classification problem where for each element we predict his head among $\lambda + 1$ possibilities (each sequence element plus a dummy element for the root and rests elements). The CE loss is therefore applied column-wise to the adjacency matrix $\hat{Y}$ predicted by our model.

In NLP, the BCE loss was used by [27] while the CE loss is used more generally, for example, by [26, 39]. We experimentally found that the sum of the two losses yields the best results.

4.4 Postprocessing

Since the prediction of our model is made independently for each potential arc, simply taking the row-wise maximum of the weighted adjacency matrix to select which head to assign to each element of the sequence could produce dependency cycles and, therefore, not yield a tree structure. We use a maximum-spanning-tree algorithm to find the tree over $\hat{Y}$ with the highest weight. Since our dependency trees are projective, we use the Eisner algorithm [40] which solves this problem using bottom–up dynamic programming with a time complexity of $O(\lambda^3)$. For applications involving non-projective trees other post-processing approaches such as Chu-Liu/Edmonds [41, 42] ($O(\lambda^2)$) are implemented in our framework.

5. EXPERIMENTS

Below, we describe the datasets, evaluation metrics, and experimental settings for the two kinds of trees we consider.

5.1 Datasets and preprocessing

We obtain the melodic time-span trees from the GTTM database [17], which contains MusicXML encodings of monophonic scores and a dedicated XML-based encoding of the constituent time-span trees (among other trees that we don't consider in this paper). We extract the note features with the Python library Partitura [43]. Some pieces have two different trees, and we keep only the first. We also discard 4 pieces that we could not import due to inconsistencies in the XML file encoding. In total, we have 296 melodies of lengths between 10 and 127 (notes + rests). For training, we augment the dataset by considering all transpositions between one octave higher and one octave lower.

We obtain the chord analyses from the Jazz Harmony Treebank (JHT) [9], which encodes both chord labels and harmonic analyses as constituent trees, in JSON format. As discussed in Section 3 this format does not distinguish between the primary and the secondary child when both have the same chord label. In this case, we assume by default that the right is the primary. The dataset contains two kinds of trees: open and complete constituents. We use the former for comparison reasons since the results are reported only for those [20]. In total, we have 150 sequences of lengths between 11 and 38 chords. For training, we augment the dataset by considering all 12 possible transpositions of the chord roots.

5.2 Evaluation metrics

The papers we compare use different metrics, and we implement all of them. The work of Harasim [20] uses two metrics, one more relevant for dependency trees and the other for constituent trees. The first is the *arc accuracy*, i.e., the normalised cardinality of the intersection between the set of predicted arcs and the ground truth arcs. The second is the *span accuracy*, computed as the normalised cardinality of the intersection between all the spans of the predicted constituent tree (i.e., the pair of the leftmost and rightmost leaf that is part of the subtree rooted at any non-leaf node) and the spans of the ground truth tree (see [20] for a more detailed explanation). Nakamura et al. [2] use the *node accuracy* metric, i.e., the normalised cardinality of the intersection between nodes in the predicted and ground truth trees, where two nodes are considered equal if the labels of the parent and children (or a dummy label if the node have no parent or children) are equal.

We also report another metric, the *head accuracy*, computed as the multiclass classification accuracy on the indices of the predicted heads, ordered by their dependent. For example for the dependency tree of Figure 1, this would correspond to the accuracy computed on the sequence $[4, 2, 3, 4, -1]$, where -1 indicates the root (which has no head). This is similar to the arc accuracy but enforces the presence of a dependency head for each sequence element (which may not be the case for a generic system), and gives more weight to the correct root prediction. It is also faster to compute and commonly used in NLP, so we include it to set a metric for future research. Note that all metrics presented above don't consider the nodes corresponding to rests, since they are only part of the input sequence, but not part of the tree.

5.3 Results

For our experiments, we set the hyperparameters of our encoder to an embedding size of 96, and 2 transformer layers of hidden size 64. The arc predictor (MLP) has 2 linear layers with the same hidden size. We use the GeLU activation [44] and the AdamW optimiser [45], with a learning rate of 0.0004 and weight decay of 0.05. We train with learning rate warm-up [46] of 50 steps and cosine annealing to limit the problem of high variance in the initial and final stages of training. The latter was particularly important

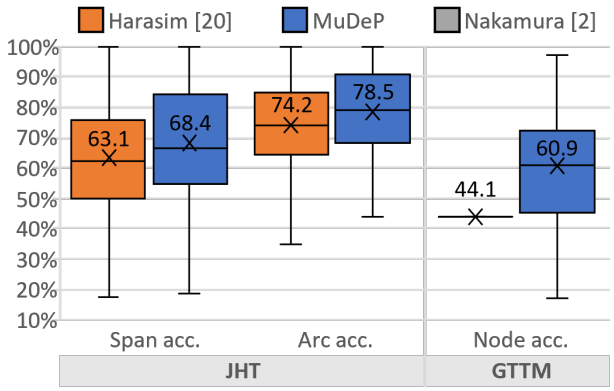


Figure 5. Boxplots of three accuracy metrics (higher is better) computed with leave-one-out cross-validation and their average. For Nakamura et al. [2], we report the average from their paper, so there is no deviation information.

since we did not use a validation set to perform early stopping due to the small size of our datasets. We train for 60 and 20 epochs for the JHT and GTTM datasets, respectively, since the latter is bigger and the data augmentation yields twice as many pieces in total). The training time is roughly the same, around 1 hour on a GPU RTX 1080.

We compare the results of our MuDeP on the JHT with Harasim [20] and on the GTTM with Nakamura et al. [2]. We use leave-one-out cross-validation, i.e., for a dataset with N pieces, we run our system N times, by training on $N-1$ pieces and evaluating with the remaining one. As shown in Figure 5, MuDeP outperforms previous methods.

By comparing the head accuracy between JHT and GTTM (79.2% vs 57.9%), it is clear that time-span prediction is a much harder problem than the chord analysis problem, despite the dataset being bigger. Another interesting result is that the span accuracy is lower than head accuracy for the JHT dataset (63.1%), but higher for the GTTM(64.8%). Apparently, the main problem for JHT is to select which two chords to connect, but the arc direction (i.e., which is the head and which is the dependent) is almost always correctly inferred; conversely, for the GTTM dataset, the system often connects the correct notes, but in the wrong direction. And this type of misprediction is punished in the head accuracy, but not in the span accuracy.

The full piece-wise statistics on all metrics, a graphical rendering of all our predicted trees, and the qualitative evaluation of some examples are available in our repository.

5.4 Ablation study

We report the difference in head accuracy averaged over 10 runs with 90/10 random train/test split for the JHT dataset. Regarding the loss, sole usage of the (multiclass) CE loss reduced the accuracy by 0.3%, and only using the binary CE loss reduced the accuracy by 4.1%. The use of a bilinear layer in the decoder reduced the accuracy by 1.2%. The absence of post-processing did not reduce the accuracy (when the network is fully trained, otherwise the reduction is very evident). This is promising but it does not automatically implies that the network is producing correctly formed trees since dependency loops could still be present in the output. There are also cases when the postprocessing is reducing the accuracy, by incorrectly deciding which arc to remove in a dependency loop.

6. CONCLUSION AND FUTURE WORK

We presented MuDeP, a system for the dependency parsing of music sequences, and a procedure to make it applicable to constituent trees. MuDeP improves upon previous methods, by incorporating the ability to consider multiple musical features simultaneously, taking advantage of sequential context, and handling noisy inputs robustly. Moreover, since it is based on widely researched deep learning components, it has the potential to scale to large datasets and longer sequences. The bottleneck for such scalability is the postprocessing algorithm with cubic complexity. Two solutions exist to this problem: if one is interested in non-projective trees, algorithms with a square complexity are available. Apart from that, our system is already having good accuracy without the postprocessing phase, as highlighted in the ablation study. Therefore, a faster heuristic may suffice to correct the few problematic dependencies without decreasing the performance.

Since our deep learning model is a black box, it is notably complicated to find a human-understandable explanation of its functioning. Although work in this direction exists [47, 48], it is still very limited [49]. Therefore, our model is mainly intended for scenarios in which one is only interested in obtaining the parsing trees, for example, to use them as input for another MIR task. Conversely, this paper might have limited utility if one's goal is to model music understanding and interpretation by humans. Grammar-based models are much more suitable for this goal, although there is a (somewhat speculative) possibility that the dependency-arc probabilities in our approach relate to first-guess heuristics.

As research on the deep learning components we use is rapidly evolving, any new discovery is likely to benefit our system. Self-supervised pretraining on larger datasets of monophonic music or chord sequences, for example by predicting next or masked tokens, could also improve the performance, as already proved for language parsing. While the goal of this paper was to present a general framework, we can also think about several domain-specific improvements, for example, training the GTTM time-span parser with a multi-target approach to predict at the same time the metrical, time-span, and prolongation structure. We hope that this work will motivate the development of more datasets of hierarchical music analyses, including datasets of dependency trees, which may be a valid alternative to constituent structures, and even open up more possibilities due to the missing projectivity constraints. Finally, we intend to explore in future research how the knowledge encoded in our model could be reused to guide other tasks, for example, automatic chord recognition from audio files.

7. 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).

8. REFERENCES

[1] S. Abdallah, N. Gold, and A. Marsden, "Analysing symbolic music with probabilistic grammars," *Computational music analysis*, pp. 157–189, 2015.

[2] E. Nakamura, M. Hamanaka, K. Hirata, and K. Yoshii, "Tree-structured probabilistic model of monophonic written music based on the generative theory of tonal music," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 276–280.

[3] M. Hamanaka, K. Hirata, and S. Tojo, "Time-span tree leveled by duration of time-span," in *Proceedings of the International Symposium on Computer Music Multidisciplinary Research (CMMR)*, 2021, pp. 155–164.

[4] C. Finkensiep and M. A. Rohrmeier, "Modeling and inferring proto-voice structure in free polyphony," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2021, pp. 189–196.

[5] M. Rohrmeier, "Towards a generative syntax of tonal harmony," *Journal of Mathematics and Music*, vol. 5, no. 1, pp. 35–53, 2011.

[6] M. Granroth-Wilding and M. Steedman, "A robust parser-interpreter for jazz chord sequences," *Journal of New Music Research*, vol. 43, no. 4, pp. 355–374, 2014.

[7] D. Harasim, M. Rohrmeier, and T. J. O'Donnell, "A generalized parsing framework for generative models of harmonic syntax." in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2018, pp. 152–159.

[8] O. Melkonian, "Music as language: putting probabilistic temporal graph grammars to good use," in *Proceedings of the ACM SIGPLAN International Workshop on Functional Art, Music, Modeling, and Design*, 2019, pp. 1–10.

[9] D. Harasim, T. J. O'Donnell, and M. A. Rohrmeier, "Harmonic syntax in time: rhythm improves grammatical models of harmony," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2019, pp. 335–342.

[10] F. Foscarin, F. Jacquemard, and P. Rigaux, "Modeling and learning rhythm structure," in *Proceedings of the Sound and Music Computing Conference (SMC)*, 2019.

[11] F. Foscarin, F. Jacquemard, P. Rigaux, and M. Sakai, "A parse-based framework for coupled rhythm quantization and score structuring," in *Proceedings of the Mathematics and Computation in Music International Conference (MCM)*. Springer, 2019, pp. 248–260.

[12] F. Foscarin, R. Fournier-S'Niehotta, and F. Jacquemard, "A diff procedure for xml music score files," in *Proceedings of the International Conference on Digital Libraries for Musicology (DLfM)*, 2019.

[13] M. Rohrmeier, "Towards a formalization of musical rhythm." in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2020, pp. 621–629.

[14] W. T. Fitch and M. D. Martins, "Hierarchical processing in music, language, and action: Lashley revisited," *Annals of the New York Academy of Sciences*, vol. 1316, no. 1, pp. 87–104, 2014.

[15] M. Tsuchiya, K. Ochiai, H. Kameoka, and S. Sagayama, "Probabilistic model of two-dimensional rhythm tree structure representation for automatic transcription of polyphonic midi signals," in *Proceedings of the Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*. IEEE, 2013, pp. 1–6.

[16] I. Sakai, "Syntax in universal translation," in *Proceedings of the International Conference on Machine Translation and Applied Language Analysis*, 1961.

[17] M. Hamanaka, K. Hirata, and S. Tojo, "Musical structural analysis database based on gttm," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2014, pp. 325–330.

[18] D. Harasim, C. Finkensiep, P. Ericson, T. J. O'Donnell, and M. Rohrmeier, "The jazz harmony treebank," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2020, pp. 207–215.

[19] 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.

[20] D. Harasim, "The learnability of the grammar of jazz: Bayesian inference of hierarchical structures in harmony," Ph.D. dissertation, EPFL, 2020.

[21] C. Finkensiep, "The structure of free polyphony," Ph.D. dissertation, EPFL, 2023.

[22] M. Hamanaka, K. Hirata, and S. Tojo, "Implementing "a generative theory of tonal music"," *Journal of New Music Research*, vol. 35, no. 4, pp. 249–277, 2006.

[23] ——, "FATTA: Full automatic time-span tree analyzer," in *International Computer Music Conference (ICMC)*, vol. 1, 2007, pp. 153–156.

[24] ——, "deepGTTM-III: Multi-task Learning with Grouping and Metrical Structures," in *Proceedings of the International Symposium on Computer Music Multidisciplinary Research (CMMR)*, 2018, pp. 238–251.

[25] Y.-R. Lai and A. W.-Y. Su, "Deep learning based detection of GPR6 GTTM global feature rule of music scores," in *Proceedings of the International Conference on New Music Concepts*, vol. 56, 2021.

[26] T. Dozat and C. D. Manning, "Deep biaffine attention for neural dependency parsing," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

[27] ——, "Simpler but more accurate semantic dependency parsing," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2018.

[28] X. Wang, J. Huang, and K. Tu, "Second-order semantic dependency parsing with end-to-end neural networks," 2019, pp. 4609—-4618.

[29] H. He and J. D. Choi, "Establishing strong baselines for the new decade: Sequence tagging, syntactic and semantic parsing with bert," in *Proceedings of the International Florida Artificial Intelligence Research Society Conference*, 2019, pp. 228–233.

[30] M. Zhang, "A survey of syntactic-semantic parsing based on constituent and dependency structures," *Science China Technological Sciences*, vol. 63, no. 10, pp. 1898–1920, 2020.

[31] F. Lerdahl and R. S. Jackendoff, *A generative theory of tonal music*. MIT press, 1985.

[32] J. Daniel, M. James H *et al.*, *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition*. Prentice Hall series in artificial intelligence, 2007.

[33] L. Kong, A. M. Rush, and N. A. Smith, "Transforming dependencies into phrase structures," in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2015, pp. 788–798.

[34] N. Fradet, J.-P. Briot, F. Chhel, A. El Fallah Seghrouchni, and N. Gutowski, "MidiTok: A python package for MIDI file tokenization," in *Late-Breaking Demo Session of the International Society for Music Information Retrieval Conference (ISMIR)*, 2021.

[35] N. Fradet, J.-P. Briot, F. Chhel, A. E. F. Seghrouchni, and N. Gutowski, "Byte pair encoding for symbolic music," *arXiv preprint arXiv:2301.11975*, 2023.

[36] EuroCC National Competence Center Sweden (ENCCS), "Graph neural networks and transformer workshop," https://enccs.github.io/gnn_transformers/notebooks/session_1/1b_vector_sums_vs_concatenation/, 2022.

[37] 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.

[38] P. Shaw, J. Uszkoreit, and A. Vaswani, "Self-attention with relative position representations," in *Proceedings of the North American Chapter of the Association for Computational Linguistics*, 2018.

[39] D. Fernández-González and C. Gómez-Rodríguez, "Transition-based semantic dependency parsing with pointer networks," *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2020.

[40] J. M. Eisner, "Three new probabilistic models for dependency parsing: An exploration," in *Proceedings of the International Conference on Computational Linguistics (COLING)*, 1996.

[41] J. Edmonds, "Optimum branchings," *Journal of Research of the national Bureau of Standards*, vol. 71, no. 4, pp. 233–240, 1967.

[42] Y.-J. Chu, "On the shortest arborescence of a directed graph," *Scientia Sinica*, vol. 14, pp. 1396–1400, 1965.

[43] C. E. Cancino-Chacón, S. D. Peter, E. Karystinaios, F. Foscarin, M. Grachten, and G. Widmer, "Partitura: A Python Package for Symbolic Music Processing," in *Proceedings of the Music Encoding Conference (MEC)*, Halifax, Canada, 2022.

[44] D. Hendrycks and K. Gimpel, "Gaussian error linear units (GELUs)," *arXiv preprint arXiv:1606.08415*, 2016.

[45] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations (ICLR)*, 2019.

[46] 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, pp. 4475–4483.

[47] S. Mishra, B. L. Sturm, and S. Dixon, "Local Interpretable Model-agnostic Explanations for Music Content Analysis," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2017, pp. 537–543.

[48] F. Foscarin, K. Hoedt, V. Praher, A. Flexer, and G. Widmer, "Concept-based techniques for "musicologist-friendly" explanations in a deep music classifier," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2022.

[49] V. Praher, K. Prinz, A. Flexer, and G. Widmer, "On the veracity of local, model-agnostic explanations in audio classification: targeted investigations with adversarial examples," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2021.

STABILIZING TRAINING WITH SOFT DYNAMIC TIME WARPING: A CASE STUDY FOR PITCH CLASS ESTIMATION WITH WEAKLY ALIGNED TARGETS

Johannes Zeitler **Simon Deniffel** **Michael Krause** **Meinard Müller**
International Audio Laboratories Erlangen, Germany
{johannes.zeitler, michael.krause, meinard.mueller}@audiolabs-erlangen.de

ABSTRACT

Soft dynamic time warping (SDTW) is a differentiable loss function that allows for training neural networks from weakly aligned data. Typically, SDTW is used to iteratively compute and refine soft alignments that compensate for temporal deviations between the training data and its weakly annotated targets. One major problem is that a mismatch between the estimated soft alignments and the reference alignments in the early training stage leads to incorrect parameter updates, making the overall training procedure unstable. In this paper, we investigate such stability issues by considering the task of pitch class estimation from music recordings as an illustrative case study. In particular, we introduce and discuss three conceptually different strategies (a hyperparameter scheduling, a diagonal prior, and a sequence unfolding strategy) with the objective of stabilizing intermediate soft alignment results. Finally, we report on experiments that demonstrate the effectiveness of the strategies and discuss efficiency and implementation issues.

1. INTRODUCTION AND RELATED WORK

Deep neural networks (DNNs) have been commonly used in many music information retrieval (MIR) tasks, such as music transcription [1], or pitch class estimation (PCE) [2, 3]. The latter provides a widely-used feature representation for various subsequent processing pipelines, e.g., audio thumbnailing [4], or chord recognition [3]. Deep learning-based feature extractors yield the highest prediction accuracy when trained on data from the same distribution, which is, however, often not readily available. Thus, one major challenge is the acquisition of a sufficient amount of correctly labeled training data. In classical music, it is often difficult to automatically annotate strongly aligned targets (short: *strong* targets), i.e., with frame-wise target labels, due to changes of tempo. On the other hand, weakly aligned targets (short: *weak* targets) only globally correspond to the input without containing frame-wise local alignments [5,6]. These weak targets are relatively easy to obtain, e.g., by only annotating start and end of an audio segment and deriving targets from the musical score. In our definition of weak targets, the order of the target vectors is correct, but their duration is unknown. Using weak targets in DNN training requires a loss function that aligns network predictions with the corresponding weak targets.

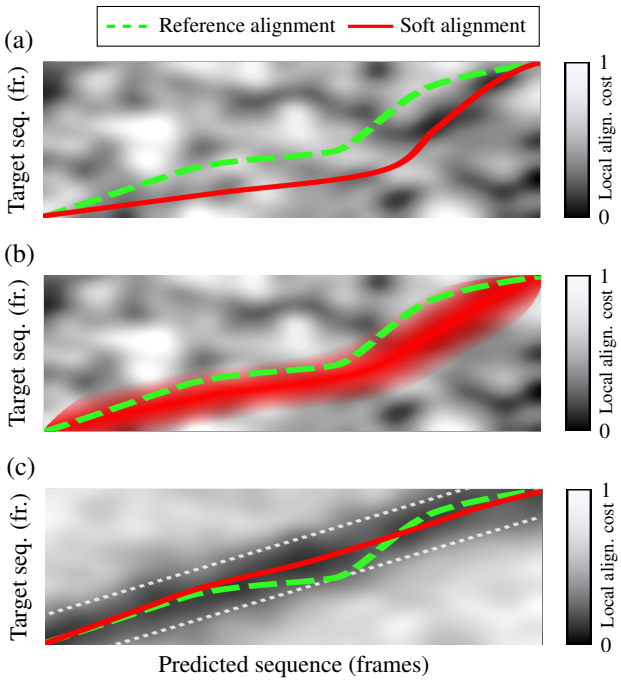


Figure 1: Deviation of strong reference alignments (dashed green) and soft alignments (red) and stabilizing strategies. **(a)** Alignment mismatch of standard SDTW. Stabilizing alignments with **(b)** hyperparameter scheduling and **(c)** diagonal prior.

In classification tasks, one widely used technique for training DNNs with weakly aligned targets is the connectionist temporal classification (CTC) loss [7], which aligns network predictions with a sequence of discrete labels. Despite being extendable to multi-label problems such as multi-pitch estimation (MPE) [8], CTC remains limited to discrete targets and is algorithmically complex.

In contrast to CTC, dynamic time warping (DTW) can be used to measure similarity between two real-valued sequences and has been successfully applied in, e.g., music

© J. Zeitler, S. Deniffel, M. Krause, and M. Müller. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** J. Zeitler, S. Deniffel, M. Krause, and M. Müller, "Stabilizing Training with Soft Dynamic Time Warping: A Case Study for Pitch Class Estimation with Weakly Aligned Targets", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.*, Milan, Italy, 2023.

synchronization and structure analysis [9]. Recently, differentiable approximations of the minimum function [10–12] have been included in DTW, enabling the usage of the DTW principle in gradient-based optimization algorithms. The algorithm proposed in [10], soft dynamic time warping (SDTW), uses so-called *soft alignments* to compute a differentiable cost measure between sequences of different length. In [13], SDTW is used in the context of performance-score synchronization and [6] employed SDTW as a loss function to train DNNs for MPE with weakly aligned pitch annotations. Experiments in [6] indicated training instabilities with SDTW when the sequence lengths of inputs and targets are significantly different. This poses a severe problem in many MIR tasks, where sequences of input audio are typically very long, while weakly labeled targets, i.e., without note durations, are significantly shorter.

In this paper, we investigate the cause of training instabilities under the SDTW loss and show that it is due to a mismatch between the estimated soft alignment and the reference alignment (see Figure 1a) in the early stages of training. This mismatch causes incorrect parameter updates and the training may diverge. Therefore, we introduce and investigate strategies to decrease this alignment error to stabilize training. In particular, we analyze a hyperparameter scheduling strategy to yield smooth alignments in the early training phase (see Figure 1b) as well as the strategy of adding a diagonal prior to the SDTW cost matrix to initially favour diagonal alignments (see Figure 1c). Furthermore, we investigate a sequence unfolding approach, where we uniformly stretch the weak target sequence to the length of the input sequence as proposed in [6]. We choose DNN-based PCE as an exemplary task to study the training process of standard SDTW and the impact of our stabilizing strategies. We demonstrate that the hyperparameter scheduling and the diagonal prior strategies reliably reduce label mismatch in the early training stage and therefore lead to successful trainings. In addition, these two strategies are computationally efficient and require only small modifications to the standard SDTW algorithm.

The remainder of this article is structured as follows. First, in Section 2, we discuss the SDTW loss function and define the concept of soft alignments. Next, in Section 3, we introduce three conceptually different strategies for stabilizing DNN training under SDTW loss. After describing the experimental setup in Section 4, we evaluate cause and effect of training problems with SDTW in Section 5, along with the impact of our stabilizing strategies. Finally, we conclude with Section 6 and give an outlook to potential areas of future research regarding SDTW-based training in MIR.

2. INTRODUCTION TO SDTW

In this section, we introduce SDTW as a loss function in a DNN training framework and define the concept of soft alignments, closely following [10, 14].

2.1 Definition

Let $X = \{\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{N-1}\}$ denote a sequence of DNN predictions, $Y = \{\mathbf{y}_0, \mathbf{y}_1, \ldots, \mathbf{y}_{M-1}\}$ denote a sequence of weak targets and $Y^{\mathrm{S}} = \left\{\mathbf{y}_0^{\mathrm{S}}, \mathbf{y}_1^{\mathrm{S}}, \ldots, \mathbf{y}_{N-1}^{\mathrm{S}}\right\}$ denote a sequence of strong targets, where $\mathbf{x}_n, \mathbf{y}_m, \mathbf{y}_n^{\mathrm{S}} \in \mathbb{R}^D$ for $n \in \{0, 1, \ldots, N-1\}$ and $m \in \{0, 1, \ldots, M-1\}$. Without loss of generality, we assume $N \geq M$.

Using the mean squared error (MSE) as a local cost function, the elements of the cost matrix $\mathbf{C} \coloneqq \mathbf{C}_{X,Y} \in \mathbb{R}^{N \times M}$ are computed as

$$\mathbf{C}_{X,Y}(n, m) = \|\mathbf{x}_n - \mathbf{y}_m\|_2^2 \,. \tag{1}$$

We next define binary alignment matrices $\mathbf{A} \in \{0, 1\}^{N \times M}$ which align two sequences of length N and M. Each matrix $\mathbf{A}$ encodes an alignment via a path of ones from cell $(0, 0)$ to $(N-1, M-1)$ using only vertical, horizontal, and diagonal unit steps [10]. All cells not corresponding to the alignment are set to zero. The set of all binary alignment matrices for sequences of length N and M is denoted $\mathcal{A}_{N,M}$. Using a differentiable approximation of the minimum function

$$\mathrm{softmin}^{\gamma}(\mathcal{S}) = -\gamma \log \sum_{s \in \mathcal{S}} \exp(-s/\gamma) \tag{2}$$

for a given finite set $\mathcal{S} \subset \mathbb{R}$ and a hyperparameter $\gamma \in \mathbb{R}$, the SDTW cost is given by

$$\mathrm{SDTW}_{\mathbf{C}}^{\gamma} = \mathrm{softmin}^{\gamma}\left(\{\langle \mathbf{A}, \mathbf{C} \rangle,\ \mathbf{A} \in \mathcal{A}_{N,M}\}\right) \tag{3}$$

and can be computed efficiently via dynamic programming [10]. The inner product $\langle \mathbf{A}, \mathbf{C} \rangle$ is the sum of all elements of $\mathbf{C}$ along the alignment given by $\mathbf{A}$.

2.2 Soft Alignments

The expectation over all alignments $\mathbf{A}$ for a cost matrix $\mathbf{C}$ is captured by the soft alignment matrix [14]

$$\mathbf{E}_{\mathbf{C}}^{\gamma} = \sum_{\mathbf{A} \in \mathcal{A}_{N,M}} p_{\mathbf{A},\mathbf{C}}^{\gamma} \mathbf{A} \quad \in \mathbb{R}^{N \times M} \,, \tag{4}$$

where the probability of an alignment is defined as

$$p_{\mathbf{A},\mathbf{C}}^{\gamma} = \frac{\exp(-\langle \mathbf{A}, \mathbf{C} \rangle / \gamma)}{\sum_{\mathbf{A}' \in \mathcal{A}_{N,M}} \exp(-\langle \mathbf{A}', \mathbf{C} \rangle / \gamma)} \,. \tag{5}$$

The soft alignment matrix is of particular interest as it is the the gradient of the SDTW cost w.r.t. the local cost matrix

$$\nabla_{\mathbf{C}} \mathrm{SDTW}_{\mathbf{C}}^{\gamma} = \mathbf{E}_{\mathbf{C}}^{\gamma} \tag{6}$$

and is computed during the backward pass of an SDTW training step with a dynamic programming algorithm [10, 14]. In contrast to the *binary* alignments $\mathbf{A}$, the entries of the soft alignment matrix $\mathbf{E}_{\mathbf{C}}^{\gamma}(n, m)$ can be interpreted as the *probability* of an alignment path going through cell (n, m). Only if this soft alignment assigns probability mass to the correct alignments (n, m), the local cost terms (1) between the correct pairs of predictions $\mathbf{x}_n$ and

targets $\mathbf{y}_m$ constitute the overall SDTW cost and the DNN parameters can be successfully trained.

The hyperparameter γ, also termed temperature, controls the smoothness of the softmin function (2). Larger values of γ lead to smooth minima in (3), i.e., with contributions of multiple alignments $\mathbf{A}$, and therefore a "blurry" soft alignment matrix $\mathbf{E}_{\mathbf{C}}^{\gamma}$ (see Figure 1b). On the other hand, small values of γ promote "sharp" soft alignments $\mathbf{E}_{\mathbf{C}}^{\gamma}$ with fewer non-zero entries (see Figure 1a), as (2) converges to the hard minimum function in the limit $\gamma \to 0$ and a single binary alignment $\mathbf{A}$ becomes dominant in (3) and (4).

3. STABILIZING TRAINING WITH SDTW

In this section, we introduce three strategies for stabilizing SDTW-based training: hyperparameter scheduling, diagonal prior, and sequence unfolding.

3.1 Hyperparameter Scheduling

As described in Section 2, the softmin temperature parameter γ controls the smoothness of the SDTW soft alignments. While a low value of γ is desirable to ensure exact correspondences between predictions and targets due to sharp alignments, the latter are problematic in the initial training phase as inaccurate predictions from randomly initialized network parameters lead to erroneous alignments, thus hampering convergence. Therefore, as a first strategy to stabilize SDTW training, we discuss an epoch-dependent scheduling of γ. Starting a training with a large softmin temperature $\gamma^{\text{start}} = 10$ makes the soft alignment fuzzier, which leads to coarse, yet mostly meaningful target assignments (see Figure 1b). After ten epochs with $\gamma = 10$, when the trained network predicts meaningful features, we linearly reduce γ during the following ten epochs to a final value of $\gamma^{\text{final}} = 0.1$, which stays constant for the remaining training.

3.2 Diagonal Prior

On average, the correct alignment of two sequences with arbitrary symbol durations has a higher probability to be close to the diagonal than to deviate from it. Therefore, as a second approach to stabilize the initial training phase, we investigate an additive prior $\mathbf{P} \in \mathbb{R}^{N \times M}$ which penalizes elements of the cost matrix $\mathbf{C}$ that are far from the diagonal (see Figure 2 for an illustration of a prior matrix). A similar strategy was employed in [15] for restricting speech-text alignments to the diagonal. Assuming equal symbol durations, the diagonal alignment of a target $\mathbf{y}_m$ starts at input frame $q_m = \lfloor \frac{Nm}{M} \rfloor$ and ends at $q_{m+1} - 1$. To yield no penalty along the diagonal and a smoothly increasing penalty for distant alignments, we define the elements of the prior matrix as

$$\mathbf{P}(n,m) = 1 - \begin{cases} 1, & q_m \leq n < q_{m+1} \\ \exp\left(\frac{(n-q_m)^2}{-2\nu}\right), & n < q_m \\ \exp\left(\frac{(n-q_{m+1})^2}{-2\nu}\right), & n \geq q_{m+1}\,, \end{cases} \tag{7}$$

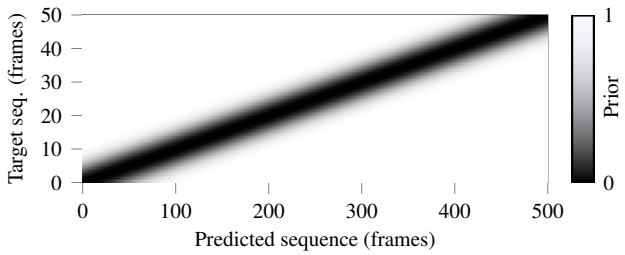


Figure 2: Diagonal prior matrix $\mathbf{P}$ for $N = 500$, $M = 50$ and $\nu = 1000$.

where the parameter ν controls the sharpness of the prior. In our experiments, we use $\nu = 1000$. Finally, the prior matrix is added to the cost matrix with a weight ω to obtain the penalized cost matrix

$$\mathbf{C}_{\mathbf{P}} \coloneqq \mathbf{C} + \omega \mathbf{P}\,, \tag{8}$$

which replaces $\mathbf{C}$ in (3) to (6). Similarly to the hyperparameter scheduling strategy, we choose a constant prior weight $\omega = 3$ during the first five epochs and then linearly reduce it to $\omega = 0$ during the following five epochs.

Note that the numerical parameters for the strategies presented in Sections 3.1 and 3.2 were determined empirically by the authors and small changes did not affect the training performance. However, when training on sequences of different length, with a different learning rate, or other DNN types, parameters should be adjusted on a validation set. As presented in Section 5, analysis of the soft alignment matrix $\mathbf{E}_{\mathbf{C}}^{\gamma}$ provides a good indication of the current alignment stability.

3.3 Sequence Unfolding

Based on the observation that equal sequence lengths stabilize SDTW training, a third strategy is to uniformly unfold the target sequence (see also [6]). The unfolded target sequence $Y^{\text{U}} = \{\mathbf{y}_0^{\text{U}}, \mathbf{y}_1^{\text{U}}, \ldots, \mathbf{y}_{N-1}^{\text{U}}\}$ is constructed by uniformly repeating elements from the weakly aligned target sequence, i.e., setting

$$\mathbf{y}_n^{\text{U}} \leftarrow \mathbf{y}_{\lfloor \frac{Mn}{N} \rfloor} \tag{9}$$

to yield equal sequence lengths of the predictions X and the targets Y^{U}. Note that the repetition of target vectors introduces ambiguities, leading to multiple optimum alignments.

4. EXPERIMENTAL SETUP

In this section, we describe the task for our case study, the employed dataset, as well as the used DNN architecture and the training procedure.

4.1 PCE Task

We choose PCE from music recordings as an illustrative case study to investigate the problems of the SDTW loss function and the effect of the stabilizing strategies. In our experimental setting, a DNN takes N frames of input audio (including context) and, for all frames, predicts twelve-dimensional pitch class activation vectors X (see Figure 3

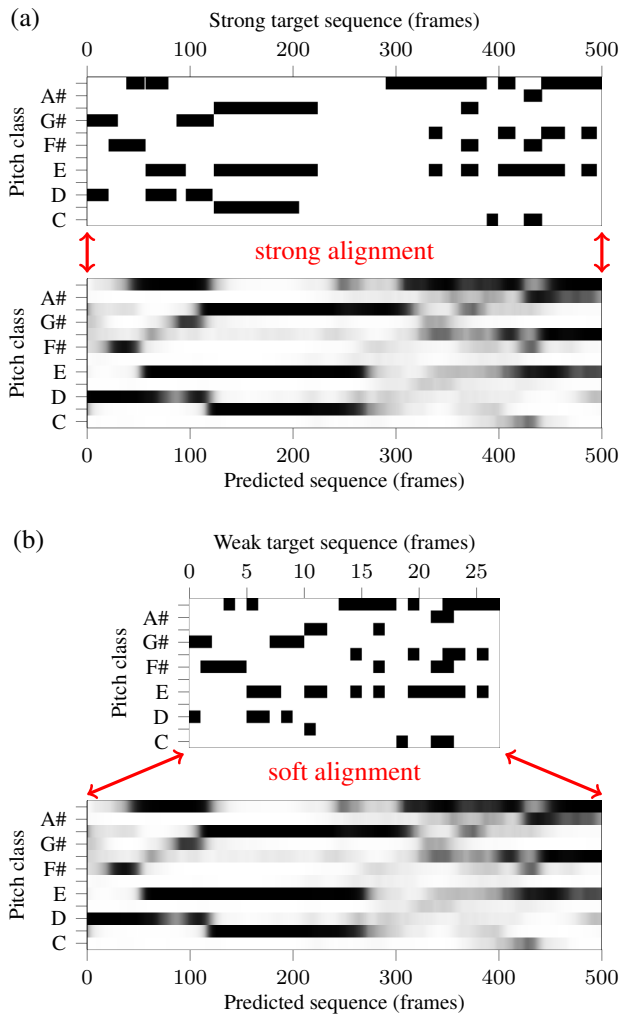


Figure 3: Alignment between training targets and predicted pitch class features X for the running example from *Frühlingstraum*. **(a)** Strong reference alignment for MSE loss with strong targets Y^{S}. **(b)** Soft alignment for SDTW loss with weak targets Y.

for an illustration of predicted pitch class features). We want to train the DNN such that the predictions X match the training targets as close as possible. In the case of strong targets Y^{S}, each predicted frame $\mathbf{x}_n$ is assigned to exactly one target frame $\mathbf{y}_n^{\mathrm{S}}$ using a strong alignment (see Figure 3a). When using weak targets Y, SDTW internally computes a soft alignment based on the cost matrix $\mathbf{C}_{X,Y}$ to assign predictions and targets (see Figure 3b).

4.2 Dataset

Throughout all experiments, we use the Schubert Winterreise dataset (SWD) [16] which contains audio recordings and strongly aligned pitch class annotations. Winterreise is a song cycle for piano and singer, consisting of 24 songs. For each song, SWD comprises nine different performances, resulting in $9 \cdot 24$ recorded songs with a total duration of $10\,\mathrm{h}\ 50\,\mathrm{min}$. We split the dataset for training, validation, and testing using a performance split [16]. The publicly available performances by Huesch (HU33, recorded in 1933) and Scarlata (SC06, recorded in 2006) were annotated manually [16] and constitute the test set. For training and evaluation we choose sequences of length $N = 500$, corresponding to approximately $8.7\,\mathrm{s}$ of audio at a sampling rate of $22\,050\,\mathrm{Hz}$ and a hop length of 384 samples. In order to generate weak training targets Y from SWD (which provides strongly aligned pitch class annotations Y^{S}, see Figure 3a), we remove all adjacent repetitions of a pitch class vector (see Figure 3b) [5]. We choose an excerpt from the song *Frühlingstraum*, performed by Randall Scarlata (SC06), as a running example (see Figure 3) to visualize the soft alignment matrices (see Figure 4).

Layer	Kernel Size	Stride	Output Shape
Prefiltering			
LayerNorm			$(N+74, 216, 5)$
Conv2D	15×15	(1,1)	$(N+74, 216, 20)$
MaxPool	3×1	(1,1)	$(N+74, 216, 20)$
Dropout			
Binning to MIDI pitches			
Conv2D	3×3	(1,3)	$(N+74, 72, 20)$
MaxPool	13×1	(1,1)	$(N+74, 72, 20)$
Dropout			
Time reduction			
Conv2D	75×1	(1,1)	$(N, 72, 10)$
Dropout			
Chroma reduction			
Conv2D	1×1	(1,1)	$(N, 72, 1)$
Dropout			
Conv2D	1×61	(1,12)	$(N, 12, 1)$

Table 1: Musically motivated CNN architecture [3, 5].

4.3 DNN Architecture and Training

We adapt a conceptually simple and musically motivated five-layer convolutional neural network (CNN) from [3, 5] with 43383 trainable parameters to predict twelve-dimensional pitch class activation vectors from an input sequence. Table 1 provides an overview of the architecture. We choose the harmonic constant-Q transform (HCQT) [17] with five harmonics as an audio feature representation, spanning six octaves at a resolution of three bins per semitone (resulting in 216 frequency bins starting from C1), a hop length of 384 samples and a frame rate of $57.4\,\mathrm{Hz}$. From an input sequence of length $N + 74$, the CNN sequentially predicts N vectors of pitch class activations. For the prediction of one frame, the CNN's receptive field covers 37 adjacent context frames on each side. Leaky ReLU with a negative slope of 0.3 is used as a nonlinearity after all hidden convolutional layers and sigmoid activation is used after the final layer. The dropout rate is set to 0.2. All models are trained using the Adam optimizer [18] with a batch size of 32 and an initial learning rate of 0.001. We reduce the learning rate by a factor of two if the validation loss did not decrease during the last four epochs, and terminate the training if the validation loss did not decrease during the last twelve epochs. At the end of training, the model from the epoch with the lowest validation loss is restored. The source code for reproducing our experiments, as well as the trained models are available on `github.com/groupmm/stabilizing_sdtw`.

Loss	Targets	γ	Strategy	F-measure mean	F-measure std
MSE	strong	-	-	**0.82**	0.07
SDTW	weak	0.1	-	0.56	0.37
SDTW	weak	0.3	-	0.63	0.32
SDTW	weak	1.0	-	0.24	0.36
SDTW	weak	3.0	-	0.31	0.38
SDTW	weak	10.0	-	0.57	0.37
SDTW	weak	$10 \to 0.1$	hyp. sched.	0.80	0.04
SDTW	weak	0.1	diag. prior	**0.81**	**0.02**
SDTW	weak	0.1	seq. unfold.	0.53	0.04

Table 2: Averaged test results for DNNs trained on strongly aligned reference targets as well as DNNs trained with SDTW on weakly aligned targets using either the standard configuration or the discussed stabilization strategies. We report the mean (higher is better) and standard deviation (lower is better) of the F-measure.

5. EVALUATION

In this section, we investigate the training process as well as the prediction accuracy under the standard SDTW loss, and compare it to the discussed stabilizing strategies. For quantitative evaluation, we repeat all DNN trainings ten times from random initializations. For the test set predictions of each trained model, we compute the F-measure w.r.t. time-pitch class bins using a threshold of 0.5. The mean and standard deviation of the F-measures from all trained models are displayed in Table 2.

5.1 Baseline: Strongly Aligned Targets

As a first baseline and an upper bound for all following experiments, we consider DNN training with strongly aligned targets Y^{S}. For the sequence lengths $M = N = 500$ and an MSE loss function, the networks achieve the overall highest mean F-measure of 0.82 with a standard deviation of 0.07 on the test set.

5.2 Standard SDTW

We next analyze DNN training with weak targets Y and the unmodified SDTW formulation from [10, 19] as a loss function. We investigate five different values of $\gamma \in \{0.1, \ldots, 10\}$ which we keep constant during training. Analyzing the mean F-measure on the test set in Table 2, the five variants with standard SDTW yield comparably low results between 0.24 and 0.57, and high standard deviations between 0.32 and 0.38. Between 20 % ($\gamma = 0.3$) and 70 % ($\gamma = 1.0$) of all training runs converged to the all-zero output, indicating a highly unstable training process of standard SDTW. In order to determine the cause of these instabilities, we analyze the quality of automatically generated soft alignments in the SDTW algorithm by visualizing the soft alignment matrix for the running example after training epochs one and 25, respectively. To highlight the effects of small and large values of γ, we focus on the edge cases $\gamma \in \{0.1, 10.0\}$. For $\gamma = 0.1$, the estimated soft alignment exhibits a sharp structure (see Figure 4a), which, after a collapse to a single target frame

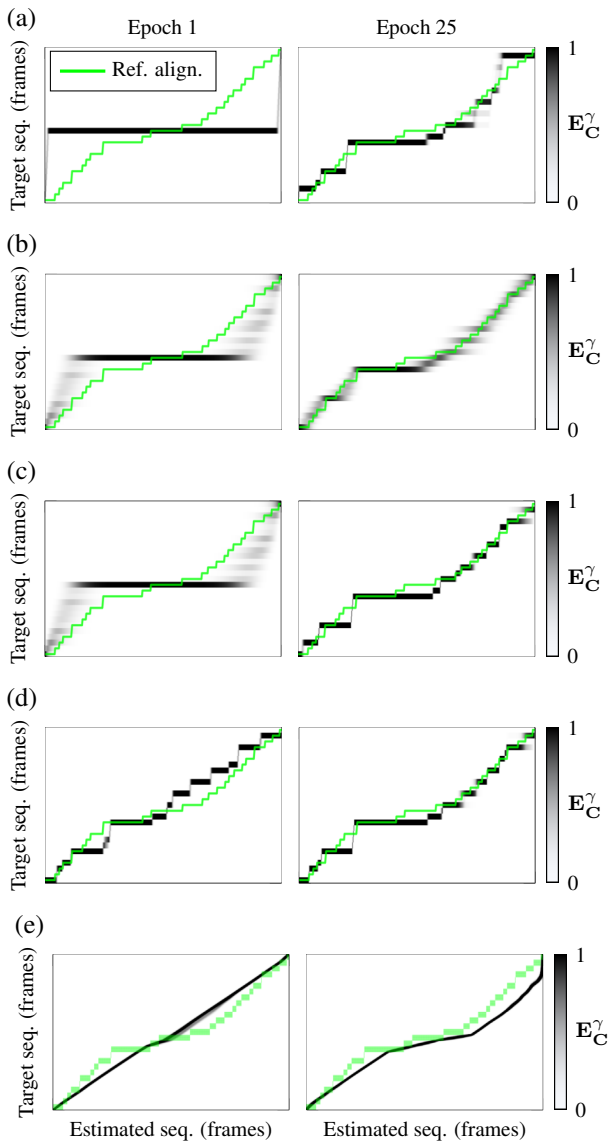


Figure 4: Reference alignment (green) and soft alignment matrix $\mathbf{E}_{\mathbf{C}}^{\gamma}$ (gray/black) for the running example after training epoch 1 (left) and epoch 25 (right) for different training strategies. **(a)** $\gamma = 0.1$, **(b)** $\gamma = 10$, **(c)** hyperparameter scheduling, **(d)** diagonal prior, **(e)** sequence unfolding.

at epoch one, still only marginally overlaps with the reference alignment after 25 epochs. This sharp and erroneous soft alignment causes unstable gradient updates and leads to the collapse of many training runs. When choosing a large softmin temperature $\gamma = 10$, SDTW yields "blurry" soft alignments (see Figure 4b) which at least partially capture the actual target frames in early epochs and coincide well with the reference alignments as training progresses. However, a blurry soft alignment also leads to blurry network predictions as multiple target frames are aligned to each predicted frame, thus resulting in a low F-measure when compared to strongly aligned targets..

5.3 Stabilizing Strategies

After evaluating the unsatisfactory training behavior of standard SDTW, we investigate the effect of the previously

introduced training strategies in the following section. We empirically choose $\gamma = 0.1$ as the final softmin temperature in all following experiments, as sharp alignments are necessary for training an estimator with frame-wise precision.

5.3.1 Hyperparameter Scheduling

First, we combine the advantages of high and low values of γ in a hyperparameter scheduling strategy. Starting a training with $\gamma = 10$, the soft alignment matrix for our running example after one epoch is blurry and at least partially overlapping with the reference alignment (see Figure 4c). The successive reduction to $\gamma = 0.1$ until epoch 20 permits sharp alignments at a later training stage. Indeed, Figure 4c shows a soft alignment after epoch 25 which is sharp and coincides well with the reference. The mean F-measure (0.80) in Table 2, as well as the standard deviation (0.04), are the second best of all SDTW-based trainings. However, as the softmin function in (2) is a lower bound for the minimum function [12] which becomes tight for $\gamma \to 0$, the SDTW loss is increasing when decreasing γ, despite unchanged network parameters. Therefore, this strategy does not allow for loss-based learning rate scheduling and early stopping before γ is set to its final value.

5.3.2 Diagonal Prior

The second strategy stabilizes SDTW trainings with low values of γ by adding a penalty cost to off-diagonal elements of the cost matrix. For our running example in Figure 4d, the soft alignment is indeed close to the diagonal after the first training epoch. As, on average, the alignments are diagonal, this often leads to correct assignments of predictions and targets even for randomly initialized DNNs. When the prior weight ω is reduced to zero after the initial training phase, the network is still able to adapt to off-diagonal alignments, as seen in our running example in Figure 4d. Analyzing the performance metrics in Table 2, using a diagonal prior yields the highest mean F-measure (0.81) and the lowest standard deviation (0.02) of all SDTW variants, almost reaching the mean F-measure of the baseline experiments with strong targets and element-wise MSE loss. Moreover, when the prior weight ω is reduced during training, the loss also decreases and therefore learning rate scheduling and early stopping are possible from the beginning.

5.3.3 Sequence Unfolding

Last, we investigate the strategy of unfolding the weak target sequence to the length of the input, which was employed in [6]. For this strategy, we observe fully diagonal soft alignments in the initial training phase, as visualized for our running example in Figure 4e. This is caused by the equal length of the predicted and the target sequence, which can be aligned using only diagonal steps. In the SDTW formulation from [10], the cost of a diagonal step is equal to the cost of a vertical or horizontal step. Thus, for a uniform cost matrix (which is probable at the initial training phase due to random network initialization), taking a diagonal step only accumulates half the cost compared to going "around the corner", i.e., one step in the vertical and one in the horizontal direction, or vice versa. This diagonalizing behavior leads, on average, to decent soft alignments in the early training phase (as discussed in Section 5.3.2). However, in contrast to the additive diagonal prior strategy, the implicit diagonalization of alignments is not reduced during the training, as can be seen in Figure 4e, which still exhibits strong diagonal components after 25 training epochs. Thus, the softly aligned SDTW targets seldom match the reference targets and performance remains low, resulting in a mean F-measure of 0.53 in Table 2.

Note that the sequence unfolding strategy adds a significant computational overhead compared to the previous two strategies, as unfolding always corresponds to using a target sequence length of $M = N$. The forward and backward pass of the SDTW loss function both have linear complexity w.r.t. the sequence lengths $\mathcal{O}(MN)$ [10]. Thus, in our setting with $N = 500$ and a mean length of the weak target sequences in the test set of $M = 24$, the unfolding strategy leads to an increase in the computational cost of the SDTW loss by a factor of more than 20.

6. CONCLUSION AND OUTLOOK

In this paper, we analyzed DNN training instabilities with SDTW as a loss function by the example of PCE. By analysis of the soft alignment matrix, we argued that alignment mismatch in the early training phase often causes a collapse of the training procedure. Motivated by these findings, we investigated three strategies for stabilizing the early training phase. We found that the previously applied strategy of unfolding the weakly aligned target sequence leads to almost exclusively diagonal alignments due to a naïve weighting of horizontal, vertical, and diagonal alignment steps. Furthermore, this strategy is computationally inefficient, as it increases the target sequence length. In contrast, the two introduced strategies of hyperparameter scheduling and diagonal prior can be implemented with negligible additional computational cost and stabilize SDTW-based training by two different mechanisms. The hyperparameter scheduling strategy promotes smooth alignments in the early training phase, which increases the probability of the predicted frame being at least partially aligned to the correct target. Penalizing off-diagonal alignments in the SDTW cost matrix by an additive diagonal prior is a strategy that initially restricts the soft alignment to a region of high probability. Experimental evaluation showed that these strategies reliably stabilize the SDTW training process. Implementing them as a default in the SDTW loss highly increases convergence rates.

Future research on SDTW-based loss functions in MIR applications might incorporate musically informed prior information, e.g., based on note durations or tempo annotations extracted from the musical score. Furthermore, the preference of diagonal alignment steps could be addressed by choosing different step weights.

7. ACKNOWLEDGEMENTS

This work was supported by the German Research Foundation (DFG MU 2686/7-2). 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.

8. 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, 2019.

[2] 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.

[3] 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.

[4] M. A. Bartsch and G. H. Wakefield, "Audio thumbnailing of popular music using chroma-based representations," *IEEE Transactions on Multimedia*, vol. 7, no. 1, pp. 96–104, 2005.

[5] C. Weiß and G. Peeters, "Training deep pitch-class representations with a multi-label CTC loss," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Online, 2021, pp. 754–761.

[6] M. Krause, C. Weiß, and M. Müller, "Soft dynamic time warping for multi-pitch estimation and beyond," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Rhodes Island, Greece, 2023.

[7] 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 International Conference on Machine Learning (ICML)*, Pittsburgh, Pennsylvania, USA, 2006, pp. 369–376.

[8] C. Weiß and G. Peeters, "Learning multi-pitch estimation from weakly aligned score-audio pairs using a multi-label CTC loss," in *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, USA, 2021, pp. 121–125.

[9] M. Müller, *Fundamentals of Music Processing – Using Python and Jupyter Notebooks*, 2nd ed. Springer Verlag, 2021.

[10] M. Cuturi and M. Blondel, "Soft-DTW: a differentiable loss function for time-series," in *Proceedings of the International Conference on Machine Learning (ICML)*, Sydney, NSW, Australia, 2017, pp. 894–903.

[11] A. Mensch and M. Blondel, "Differentiable dynamic programming for structured prediction and attention," in *Proceedings of the International Conference on Machine Learning (ICML)*, Stockholmsmässan, Stockholm, Sweden, 2018, pp. 3459–3468.

[12] I. Hadji, K. G. Derpanis, and A. D. Jepson, "Representation learning via global temporal alignment and cycle-consistency," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Virtual, 2021, pp. 11 068–11 077.

[13] R. Agrawal, D. Wolff, and S. Dixon, "A convolutional-attentional neural framework for structure-aware performance-score synchronization," *IEEE Signal Processing Letters*, vol. 29, pp. 344–348, 2021.

[14] M. Blondel, A. Mensch, and J. Vert, "Differentiable divergences between time series," in *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, Virtual, 2021, pp. 3853–3861.

[15] K. Shih, R. Valle, R. Badlani, A. Łańcucki, W. Piang, and B. Catanzaro, "RAD-TTS: Parallel flow-based TTS with robust alignment learning and diverse synthesis," in *International Conference on Machine Learning (ICML), Third Workshop on Invertible Neural Networks, Normalizing Flows, and Explicit Likelihood Models*, Virtual, 2021.

[16] C. Weiß, F. Zalkow, V. Arifi-Müller, M. Müller, H. V. Koops, A. Volk, and H. Grohganz, "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.

[17] R. M. Bittner, B. McFee, J. Salamon, P. Li, and J. P. Bello, "Deep salience representations for F0 tracking in polyphonic music," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Suzhou, China, 2017, pp. 63–70.

[18] 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.

[19] M. Maghoumi, E. M. Taranta, and J. LaViola, "DeepNAG: Deep non-adversarial gesture generation," in *Proceedings of the International Conference on Intelligent User Interfaces (IUI)*, College Station, Texas, USA, 2021, pp. 213–223.

FINDING TORI: SELF-SUPERVISED LEARNING FOR ANALYZING KOREAN FOLK SONG

Danbinaerin Han[1] **Rafael Caro Repetto**[2] **Dasaem Jeong**[1]
[1]Department of Art & Technology, Sogang University [2]Kunstuniversität Graz
naerin71@sogang.ac.kr, rafael.caro-repetto@kug.ac.at, dasaemj@sogang.ac.kr

ABSTRACT

In this paper, we introduce a computational analysis of the field recording dataset of approximately 700 hours of Korean folk songs, which were recorded around 1980-90s. Because most of the songs were sung by non-expert musicians without accompaniment, the dataset provides several challenges. To address this challenge, we utilized self-supervised learning with convolutional neural network based on pitch contour, then analyzed how the musical concept of *tori*, a classification system defined by a specific scale, ornamental notes, and an idiomatic melodic contour, is captured by the model. The experimental result shows that our approach can better capture the characteristics of tori compared to traditional pitch histograms. Using our approaches, we have examined how musical discussions proposed in existing academia manifest in the actual field recordings of Korean folk songs.

1. INTRODUCTION

Folk songs are considered as a musical language. Not only that, but they are also regarded as the root of all traditional music. Folk songs are potent embodiments of a region's cultural and linguistic characteristics, serving as a foundation for all artistic and musical developments since their inception. It is believed that research for folk music can provide new inspiration for existing art music research, and facilitate an interdisciplinary approach that encompasses both music and other fields.

In Korean musicology, there have been ongoing discussions aimed at identifying regional characteristics in folk music. Through numerous debates, *tori* is used as the most representative theory in many studies. Even though the use of tori as a term to describe the musical characteristics of Korean folk music is widespread and its utility is well acknowledged, the existing tori classification methods were unable to fully explain the features of the actual music. Various scholars have defined the different tori according to musical characteristics such as scales (intervals between notes), primary notes, ornamentation, ending note and idiomatic melodic patterns in folk songs, leading to a refinement process for the classification of tori. So far, there are still ongoing opinions, controversies, and discussions concerning the existing tori [1, 2].

In order to conduct valuable discussions on analyzing folk songs, the relationships within the audio must be examined and systematically shared. However, checking numerous audio files individually is extremely time-consuming and difficult, so previous research has focused on analyzing small amounts of audio. The most common method used by musicologists has been transcription, which involves listening to the music and notating it in a specific music notation system. However, the task of transcribing orally transmitted music into a music notation system has inherent limitations [3], such as quantizing the pitch and rhythmic features of folk songs.

For these reasons, there have been several research on analyzing folk music empowered with methods derived from music information retrieval (MIR) on audio recording. For example, Indian *raga* music has been analyzed based on first-order pitch distrubutions [4], and for traditional three-part Georgian singing, F0-based tonal analysis [5] or development of tuning systems [6] has been introduced. [7] also presented analysis using audio-signal processing on Flamenco and Arab-Andalusian vocal music.

In this paper, we take a computational approach to analyze a vast corpus of Korean folk songs, utilizing deep-learning-based methodologies. Our primary aim is to investigate the connection between conventional musicological classifications of Korean folk songs, *tori*, and the actual field recordings. Through a comparison of our analytical results with established musicological frameworks, our objective is to offer insights on identifying meaningful clustering or distinguishable features among Korean folk songs. We also share our code and metadata that contains links to original audio with our manual tori labels for 218 songs [1].

2. KOREAN FOLK SONGS AND TORI

2.1 Dataset

The "Anthology of Korean Traditional Folksongs" is an audio collection consisting solely of traditional Korean folk songs [8]. As part of a cultural project conducted by Munhwa Broadcasting Corporation (MBC) from 1989

[1] https://github.com/danbinaerinHan/finding-tori

© D. Han, R. Caro Repetto, and D. Jeong. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).
Attribution: D. Han, R. Caro Repetto, and D. Jeong, "Finding Tori: Self-supervised Learning for Analyzing Korean Folk Song", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

to 1995, the folk song collection was compiled under the direction of Sang-il Choi. The available audio contains 15,861 songs, with approximately 700 hours in total length.

The audio was field-recorded across 153 city/county and 1,010 villages in South Korea. The metadata accompanying the recordings includes title, machine-readable lyrics, regions, recording dates, and the singer's name and age. The region represents the administrative districts of South Korea at the time of recording, which consists of nine categories in total. Due to its extensive audio corpus and detailed accompanying metadata, this dataset has become a crucial resource for research on Korean folk music within the domestic academic community [9–11]. We obtained the audio material through crawling the original website, [2] which has been hosted by MBC since April 2022, where every audio and metadata is openly published. We received official approval for using these data for research purposes from MBC.

2.2 Tori

Tori was proposed by a Korean musicologist named Bo-hyeong Lee in the early 1980s to explain the musical characteristics of regional folk songs [12, 13]. Prior to the development of tori theory, Korean music academia used the terms *jo* and *cheong* to explain the musical features of folk songs, based on the tonal center and intervals between notes [14]. However, Bo-hyeong Lee argued that *jo* and *cheong* are insufficient in capturing the unique musical characteristics of Korean folk music. Therefore, he proposed the tori classification system as an alternative approach, which covers primary notes, ornamentation, ending note and idiomatic melodic patterns.

The most widely-used tori classification method divides songs into four categories. We encourage readers to refer Figure 3, which presents scale of three tori with staff notation. The pitch name in this section is used as an conventional representation, not absolute pitches. **Gyung-tori** utilizes five notes and is mainly sung in the capital region. It often uses a gentle vibrato overall and finessed ornamental melody. **Menari-tori** is widely distributed throughout the eastern regions and the entire Korean Peninsula. When the melody ascends, it leaps through notes, while during the descending melody, it comes down through passing notes. **Yukjabaegi-tori** is commonly found in the southern regions of the Korean Peninsula. It is characterized by a thick and vibrant vibrato in G note and passing briefly through E♭ in a descending melody from E♭ to D. **Sushimga-tori** commonly appears in the northwestern region of the Korean Peninsula, characterized by its unique vibrato inflecting upward.

2.3 Tori annotation

As the dataset did not include any tori labels, one of the authors, who has over 10 years of experience in Korean traditional music, selected a subset of 218 high-quality recordings and manually annotated them in terms of tori classification.

[2] http://urisori.co.kr/urisori-en/doku.php/

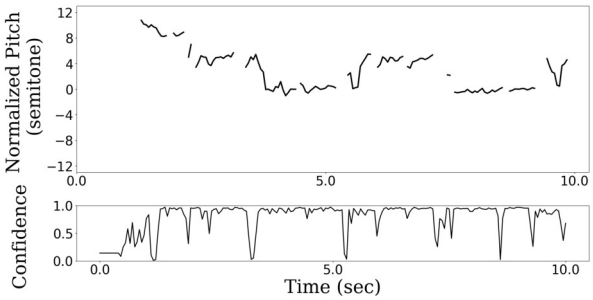


Figure 1. Example of pitch contour from the dataset extracted by CREPE. F0 value under confidence of 0.8 was masked out

To create this tori-subset, we focused on identifying clear musical characteristics present in the tori classification method, such as pitch scale, ornamentation, and idiomatic melodies, rather than considering the audio's recorded region. We found that there are not many instances of sushimga-tori in the audio dataset, as the dataset predominantly consists of folk songs from the central and southern regions of the Korean Peninsula. Also, we observed that songs belonging to menari-tori appeared not only in the eastern region but also nationwide. In conclusion, we mainly focused on the remaining three types of tori, *gyung-tori*, *yukjabaegi-tori*, and *menari-tori*. Finally, audio recordings that were closer to speech or chanting (non-musical) and did not exclusively belong to any other subdivided tori types were labeled as 'others'. We have labeled totally 218 songs: 65 gyung, 73 menari, 49 yukja, and 31 others.

3. METHODOLOGY

In this study, we take an approach of representing the musical characteristics of a given folk song using a high-dimensional single vector, which can also be called an embedding of the song. If this embedding accurately represents the musical characteristics of tori, it can be utilized for various purposes including tori classification, song similarity searches and clustering songs in the corpus based on tori similarity.

Even though the dataset is provided with audio recordings, we focus on the contour of fundamental frequency (F0) rather than using audio directly. By doing so, our embedding could better capture the melodic features of the songs rather than timbral characterstics such as the dialect of the singer. To extract the F0 contour from each audio recording, we utilized CREPE [15], a widely-used CNN-based model. It extracts F0 value for every 10 ms as well as confidence score ranging from 0 to 1. Figure 1 illustrates an example of an extracted pitch contour.

3.1 Pitch Histogram

Since the characteristics of a musical system can be largely studied from the pitch distribution of its melodies, there

has been numerous research using pitch histograms to analyze traditional musics from different culture, such as for Indian Carnatic music [16], Turkish Makam [17], Arab-Andalusian music [18], or Iranian *dastgāhi* music [19]. Therefore, we applied a similar approach to Korean folk music.

One of the important features one has to know to exploit pitch histogram is the tonic of the song so that one can normalize the pitch histograms of each song into relative pitch, rather than directly using absolute pitch value. During our preliminary experiment, we found that the tonic of each recording in the dataset is usually the most frequently appearing pitch in the song. Therefore, we determined the tonic by counting the number of appearances of each pitch in terms of time frames, utilizing a 100 cents range to count pitch (i.e. MIDI pitch 60.49 and 59.51 are counted as the same pitch). As the center pitch may differ from the typical 440 Hz tuning, we identified the best matching pitch by adjusting the pitch center by increments of 10 cents and selecting the pitch that showed the maximum pitch count.

3.2 CNN contour encoder

While the pitch histogram can offer a comprehensible representation of each song, it cannot fully capture the concept of tori as it does not consider the relationship between each note in the scale. Additionally, because most of the recordings were sung by non-professional singers without instrumental accompaniment, the intervals of singing and tonic frequency has signifcant amount of noise. Therefore, we employ a convolutional neural network (CNN) to learn the representation of a given pitch contour. This approach allows us to better capture the underlying musical characteristics of each song in relation to tori classification.

The problem is setting the training objective of the CNN model. One option is to train the model in a supervised manner, as typically done in other MIR tasks, such as classification tasks with an annotated dataset. In our dataset, region labels, which represents the administrative region where the recording took place, can be used for training labels. It can be regarded as a reasonable approach considering that tori has strong correlation with the regional characteristics of each folk song. However, we discovered several different types of tori in a single recorded region, which would make it difficult to learn distinguishable musical characteristics only using the region label.

To address this issue, we propose to adopt a self-supervised representation learning, so that we can obtain a representation of given pitch contour that is consistent within a song without extra annotated labels as shown in Fig. 2. This approach of exploiting intra-song similarity has been widely used for music audio representation learning [20,21]. We use triplet loss with hinge margin as Equation 1,

$$L = \max(0, m - \text{Sim}(v_a, v_p) + \text{Sim}(v_a, v_n)) \quad (1)$$

where m denotes hinge margin, $\text{Sim}(v_a, v_{p,n})$ represents cosine similarity between anchor vector v_a and positive vector v_p or negative vector v_n.

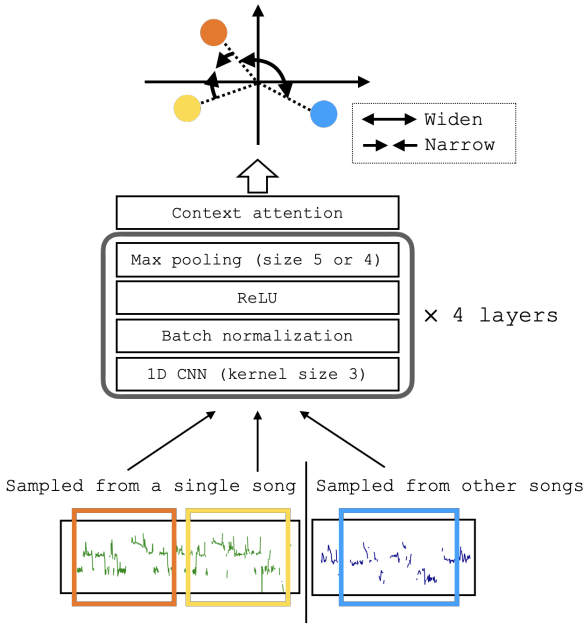


Figure 2. Self-supervised learning with triplet loss using cosine similarity, with four layers of convolutional neural network

As presented in Fig. 2, the model consists of 4 layers of 1D convolution layer, each with a kernel size of three and channel size of 64, 128, 256, and 256, respectively, each followed by batch normalization layer. We used a max pooling layer between each convolutional layer with size 5 for the first layer and 4 for the second and third layers. On top of the convolutional stack, we used context attention to summarize the arbitrary length of vectors, which was initially proposed in hierarchical document classification [22], with a slight modification of multi-head attention weight [23] instead of single-head. Context attention is a type of attention mechanism that uses an independent learnable parameter named context vector as a query while using the input vectors as key and value. The final embedding size of the model was 256.

The input is a pitch contour in the shape of $T \times 2$, where T denotes the number of time step of pitch contour. Throughout the experiment, we used the frame rate of 20 Hz, which means that 30 seconds of pitch contour is converted to 600 time steps. During the training, we randomly slice the contour to have 30 seconds length, while using the entire contour in the test or visualization. Two channels are tonic-normalized MIDI pitch and confidence of the F0 estimation at that time frame. Since the automatic F0 estimation includes noise, especially in the unsung silent part, we masked the estimated pitch value to zero if the confidence is lower than 0.8.

4. EXPERIMENTS

4.1 Evaluation Design

In the experiments, we explored the quantitative relation between tori and different embedding schemes to see which type of embedding can explain the concept of tori

better. To evaluate the correlation between tori and the given embedding scheme, we exploit our tori-labeled subset with two metrics.

The first metric is the ranking of cosine similarity. If the embedding shares important characteristics with the concept of tori, we can expect that songs with high cosine similarity in the given embedding space share the same tori with the query song. Therefore, we calculated the cosine similarity between each song in the tori-subset, and obtained normalized discounted cumulative gain (nDCG), which is a frequently used metric to evaluate the quality of search results. If all the other n songs with the same tori are ranked within n-th order in the similarity, nDCG becomes one.

The second metric is the tori classification accuracy using a random forest classifier. If the embedding includes essential characteristics that define tori, one can expect that it can be exploited to classify the tori for a given song. Among many options, we used a simple random forest classifier. The random forest classifier with 100 trees was trained with 75% of the tori-set and tested with the remaining 25%. For each embedding scheme, we repeated the procedure 30 times with different train/test split and reported mean and deviation of accuracy to ensure that the results were not dependent on the specific dataset split.

4.2 Training

While a pitch histogram can be obtained from a given song without additional training data, CNN models require training procedures. We employ F0 contour extracted from the "Anthology of Korean Traditional Folksongs" to train our model.

Some of the songs in the dataset were recorded with multiple singers or percussion accompaniment. Because these can degrade the performance of F0 estimation, we excluded them while training the CNN model. Instead of manually filtering the dataset, we used a CNN-based sound event detection model [24] to calculate the activation of 'choir' and 'percussion,' which are included among the model's event vocabulary. We filtered the song by the maximum activation value exceeding a certain threshold, which was manually decided by observing the activation distribution across the dataset.

The CNN model was trained for 25,000 updates. For the self-supervised learning using triplet loss, we used eight negative samples with one positive sample, and hinge loss using a margin of 0.4. For the region-supervised training, we use cross-entropy loss with class weight to address the class imbalance issue. The architecture of the CNN remained the same, but we added a fully connected layer to project the embedding to logits for nine different region categories. We used the Adam optimizer with an initial learning rate of 0.001 and batch size of 128. The entire training process was conducted on a single RTX A5000, which took approximately 10 minutes to complete 25,000 updates. We validated the model's training procedure and hyperparameters using a specific split of the tori-subset and chose the simplest setting to avoid overfitting the hyperparameters to the small subset.

	Similarity Ranking	Random Forest Classifier
Embedding	nDCG	Accuracy
Hist. (25 bin)	0.783	0.744 $\pm$0.058
Hist. (124 bin)	0.777	0.722 $\pm$0.054
CNN (region)	0.792	0.634 $\pm$0.055
CNN (SSL)	**0.853**	**0.848** $\pm$0.039

Table 1. Experiment results on the tori subset. Hist. denotes normalized pitch histogram, specified with the number of bins to cover two-octave range. region and SSL denotes the region-supervised and self-supervised learning.

4.3 Results

The evaluation result of each embedding scheme is presented in Table 1. As tori is closely related to the use of pitch, the pitch histogram showed about 74 % accuracy when combined with a random forest classifier, and 0.783 or 0.777 nDCG in the similar song search. The result shows that increasing the resolution of the histogram does not achieve better performance. The performance of CNN embeddings trained with region classification was worse than the pitch histogram in classification accuracy. However, the CNN embedding trained with self-supervised learning showed the best nDCG and classification accuracy, even though both CNN shares the same architecture except the final projection layer.

The performance gain compared to the pitch histogram can be caused by the fact the concept of tori includes not only the pitch scale the song uses but also ornamentation and idiomatic phrases it, which can be easily captured by contour CNN.

The result also shows that the region label did not help to learn tori-related characteristics. Even though the concept of tori is strongly related to each region, folk songs were widely spread nationwide at the time of the recording, which made it difficult to learn coherent musical characteristics from the region label.

It is also worth noting that self-supervised learning did not use any other musical knowledge that is related to tori, except that it was trained to extract constant embedding throughout a given song regardless of the segment position. From a machine learning point of view, the model has to extract an embedding that can explain the entire song or distinguish it from other songs from a given fragment. The high correlation between the trained embedding and tori labels shown in the evaluation implies that the concept of tori is clearly related to the distinguishable characteristics of each Korean folk song.

5. ANALYSIS AND DISCUSSION

In this section, we introduce interesting musicological findings that we have found from the pitch histogram and embedding learned from self-supervised training.

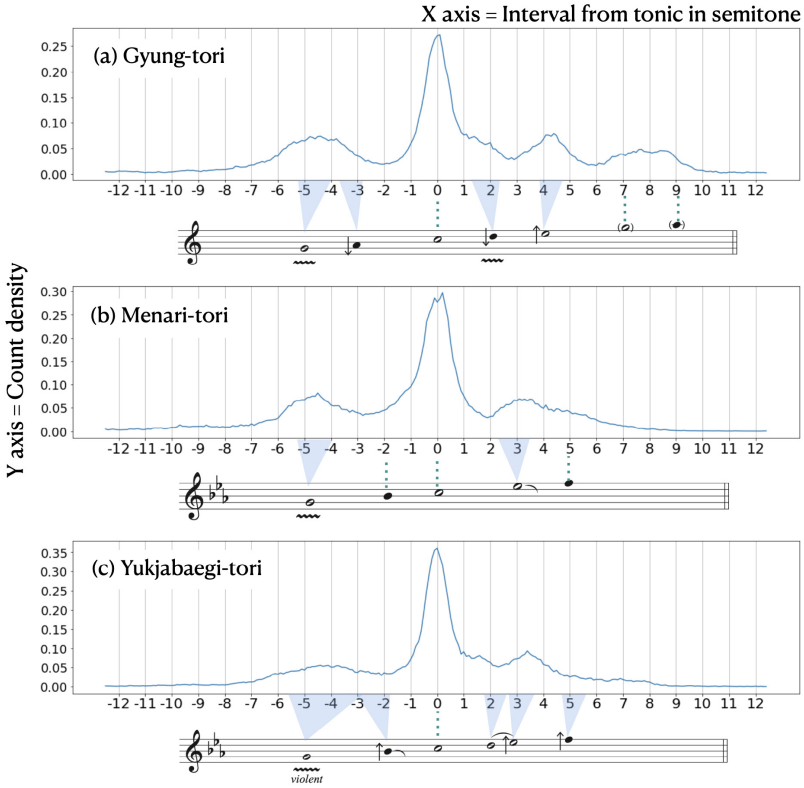


Figure 3. Average pitch histogram of each tori obtained from the tori-subset and corresponding scale description.

5.1 Pitch Histogram Analysis

To explain the scale and characteristics alongside with pitch histogram of each tori, we present the staff notation of each tori in Figure 3 based on various studies describing tori [2, 25, 26]. In the score, the *cheong* (a main note that generally appears most frequently) was fixed as the C5. The unfilled notes are primary tones, and the filled notes are passing tones. In cases where there is a vibrato mark below the note, it is called *yoseong*, meaning the pitch generally oscillates up and down. In addition, the arrow on the left side of the note indicates that the pitch is slightly raised or lowered compared to the equal-tempered pitch. The slur mark flowing down to the right of the note represents *twoeseong*, which means the pitch slides down.

While menari-tori and yukjabaegi-tori remain within a perfect fifth range from the center, in gyung-tori, there is a larger proportion of high notes (G5-A5) or more from the center. This indicates that the distribution of notes from low to high in gyung-tori is evenly spread. This could be due to the fact that folk songs of gyung-tori are more commonly found in popular folk songs than in those that appear nationwide. Perfect 4th below the center tone is common in all Korean music. However for example in gyung-tori, there is vibrato in the G4 note, and the A4 note has a low pitch, resulting in a distinct distribution in that part. In menari-tori, it can be observed that the B♭4 note is used as a passing tone between C5 and G4. Yukjabaegi-tori is characterized by a vibrant vibrato in the G4, resulting in a gentle distribution in the nearby pitch than other tori.

Whether a major third or a minor third is used in the scale is an important factor in distinguishing the regional musical characteristics of the eastern and western parts of the Korean Peninsula in the traditional method. Subtle pitch differences can be observed in all three tori. In gyung-tori, the D5 note is clearly distributed lower and the E5 note is definitely higher than the line appearing the equal-tempered tune. In yukjabaegi-tori, it is also confirmed that the E♭ note is slightly higher than the minor third note's

Through the above analysis, we found that pitch histograms can be utilized to roughly identify musical characteristics, such as the range, interval from tones and subtle pitches, etc. However, we also acknowledged that this approach solely exhibits the constituent pitches and their frequencies, without accounting for other aspects like uses of melody and ornamentation and more.

5.2 UMAP visualization

We visualized the embeddings learned through self-supervised learning in 2D using UMAP, a dimension reduction technique that is frequently used for visualization of high-dimensional vectors in Figure 4. In the first figure, we demonstrated the distribution of four different tori labels. In addition, by examining the metadata information of the embeddings, we figured out several interesting topics from this visualization. We also implemented a web demo using this visualization where one can directly access the corresponding audio recording [3] .

Menari-tori has a much wider distribution area and en-

[3] https://danbinaerinhan.github.io/korean-folksong-visualization/

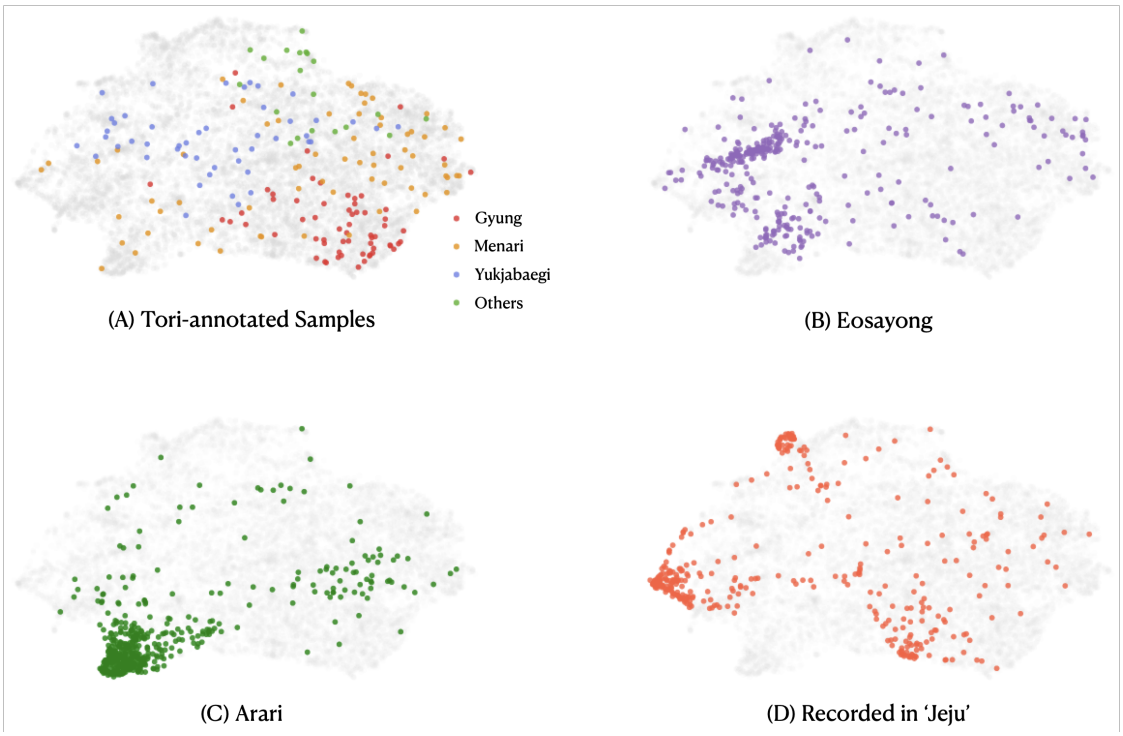


Figure 4. UMAP visualization of folk song embeddings obtained from our self-supervised-trained CNN model. Note that no tori label was used to train the model.

compasses a broader musical range. In Figure 4 (A), the menari-tori distribution can be seen to be wider compared to the other tori. We mentioned that several subdivided tori of it have been identified by scholars. Among them, eosayong-tori is a representative example of refining the musical characteristics of eastern folk songs [11]. Eosayong-tori, a lamenting song sung by lumberjacks, was suggested to have distinct musical characteristics compared to menari-tori. For instance, in eosayong-tori, the lowest pitch is a semitone higher, and it concludes on the lowest pitch of the scale instead of the final pitch and the middle pitch found in menari-tori. We examined embeddings corresponding to eosayong-tori in the metadata's title information, and they were clearly gathered in a different space than the area annotated as menari-tori, as presented in Figure 4 (B).

Similar results can be observed in the following cases with *arari*. *Arirang*, having repetitive refrain lyrics such as "arirang" or "arari", is the most representative Korean folk song, with countless versions appearing in various regions. Among them, *arari*, another name for *Jeongseon arirang*, was sung primarily in Gangwon province before spreading nationwide in the 1930s [27,28]. Arari is also regarded as a separated tori from menari-tori for some researchers [29], with its own slower tempo, monotonous skeleton tones, and the use of decorative tones. In Figure 4 (C), we can see the clustering of entities with the title arari extracted from the metadata.

Another interesting example is songs from Jeju. Folk songs sung from Jeju island, which is the largest island in Korea, have fewer research results compared to other regions. In Jeju folk songs, the interval between the pitches is narrower compared to other provinces'. Due to these reasons, the traditional method of transcribing into staff notation was unsuitable for capturing the characteristics of Jeju music [30–32]. However, using our embeddings, we can identify three clusters as in Figure 4 (D). It implies the potential usefulness of our approach to uncover the unique characteristics of folk songs in Jeju Island.

We also checked the UMAP of the pitch histograms but could not find clear clustering as in the presented examples.

6. CONCLUSIONS

In this paper, we have presented our computational approach to obtaining a high-dimensional embedding vector for a given pitch contour, which allows us to analyze a vast amount of Korean folk songs. Using these embeddings and a manually labeled subset, we have examined how musical discussions proposed in existing academia are manifested in our dataset. As our results cover various music characteristics, the learned embeddings can be utilized as a monitoring aid when dealing with numerous undefined data to review and find the tori. Also, we have discussed the methods and possibilities for utilizing and interpreting experimental results in the future research. Through this material, we can review and refine our understanding of the concept of the tori, and provide easily accessible resources and utilize them as appropriate evidence. Furthermore, there is potential to clarify the musical characteristics of regions that are distinct from other regions, like Jeju, about which existing research has not been as active. Ultimately, we expect this research to be valuable in illuminating the relationships and transformations of folk songs over time.

7. ACKNOWLEDGEMENT

We wish to express our deepest gratitude to Sang-il Choi and his team for their invaluable work on the "Anthology of Korean Traditional Folksongs". Their effort to collect more than ten thousand unique recordings from across Korea has not only significantly contributed to the preservation of the rich tapestry of Korean folk music but has also created a monumental resource for academic research. Without their extraordinary efforts, our work would not have been possible. We sincerely thank them for laying such a robust foundation and for enabling future researchers and us to build upon it. We also appreciate MBC for kindly allowing us to utilize the dataset for this research.

The web demo was implemented with the kind help of Dongmin Kim.

8. REFERENCES

[1] E. J. Shin, "Reconsideration on the modes(tori) of korean folk song(한국 민요 선법 (토리) 의 연구 성과 검토 및 논점)," *The Society Of Korean Folk Song(한국민요학*, vol. 46, pp. 153–195, 2016.

[2] G. R. Sung, *Controversial Topics in the Research of Korean Music Theory, 한국음악이론 연구의 쟁점*. Publishing department of the academy of Korean studies(한국학중앙연구원출판부), 2020.

[3] H. J. Kim, *Transcription and Interpretation of Korean Folk Songs(민요의 채보와 해석)*. Publish Company Minsokwon(민속원), 2013.

[4] G. K. Koduri, S. Gulati, P. Rao, and X. Serra, "Rāga recognition based on pitch distribution methods," *Journal of New Music Research*, vol. 41, no. 4, pp. 337–350, 2012.

[5] S. Rosenzweig, F. Scherbaum, and M. Müller, "Detecting stable regions in frequency trajectories for tonal analysis of traditional georgian vocal music." in *Proc. of Int. Society of Music Information Retrieval Conf. (ISMIR)*, 2019, pp. 352–359.

[6] 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.

[7] N. Kroher, E. Gómez, A. Chaachoo, M. Sordo, J.-M. Díaz-Báñez, F. Gómez, and e.-R. Mora, Joaquin", *Computational Ethnomusicology: A Study of Flamenco and Arab-Andalusian Vocal Music", bookTitle="Springer Handbook of Systematic Musicology*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2018, pp. 885–897.

[8] S. I. Choi, "Articles on recordings : 'anthology of korean traditional folksongs' about the project and the records published(창간 10 주년 기념호: 음반;'한국민요대전'사업과 음반 발간)," *Korean Recording Studies(한국음반학)*, vol. 10, pp. 459–480, 2000.

[9] I. G. Bae, "Musicological study of the tori of chungcheong province : focusing on the rice field weeding folksongs contained in『mbc hankuk minyo daejeon (the compilation of korean folksongs)』(충청도 토리를 찾기 위한 시론:[mbc 한국민요대전] 수록 논매기소리를 중심으로)," *The Society Of Korean Folk Song(한국민요학*, vol. 12, pp. 159–184, 2003.

[10] N. kyong Choi, "Distribution of folk songs and its musical characteristics(1) (민요의 종류와 음악적 특성의 분포 연구 (1)-모심는 소리, 노젓는 소리, 만선 풍장소리를 중심으로)," *Research Institute of Korean Studies(민족문화연구)*, vol. 41, pp. 85–127, 2004.

[11] Y. W. Kim, ""a study of the yoengnam folk song, eosayong(嶺南民謠 어사용의 音組織 硏究)," *The Society Of Korean Folk Song(The Society Of Korean Folk Song(한국민요학*, vol. 6, pp. 45–131, 1999.

[12] B. H. Lee, "Music and folk song of central and western region and tori(경서토리권의 무가, 민요)," *Collection of Korean Music Studies(나운영박사 회갑기념 한국음악논총)*, 1982.

[13] ——, "Music culture of the menari-tori folk song(메나리토리 무가 민요권의 음악문화)," *Koanthro(한국문화인류학)*, vol. 15, pp. 233–249, 1983.

[14] D. W. Baek, "The definition and interpretation of terminology of korean musicology, 한국음악학에서의 용어 개념 정의와 해석," *National Folk Museum of Korea(민속학연구)*, no. 14, pp. 193–230, 2004.

[15] 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.

[16] G. K. Koduri, J. Serrà Julià, and X. Serra, "Characterization of intonation in carnatic music by parametrizing pitch histograms," in *Proc. of the 13th Int. Society for Music Information Retrieval Conf.* International Society for Music Information Retrieval (ISMIR), 2012.

[17] A. C. Gedik and B. Bozkurt, "Pitch-frequency histogram-based music information retrieval for turkish music," *Signal Processing*, vol. 90, no. 4, pp. 1049–1063, 2010.

[18] N. Pretto, B. Bozkurt, R. Caro Repetto, and X. Serra, "Nawba recognition for arab-andalusian music using templates from music scores," in *Proc. of the 15th Sound and Music Computing Conference (SMC2018)*. Cyprus University of Technology, 2018, pp. 394–99.

[19] B. Nikzat and R. Caro Repetto, "KDC: An open corpus for computational research of *dastgāhi* music," in *Proc. of the 23rd Int. Society for Music Information Retrieval Conf.* International Society for Music Information Retrieval (ISMIR), 2022, pp. 321–328.

[20] J. Lee, J. Park, and J. Nam, "Representation learning of music using artist, album, and track information," in *Machine Learning for Music Discovery Workshop, International Conference on Machine Learning*, 2019.

[21] J. Spijkervet and J. A. Burgoyne, "Contrastive learning of musical representations," in *Proc. of the 22nd Int. Society for Music Information Retrieval (ISMIR)*, 2021.

[22] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proc. of the 2016 conf. of the North American chapter of the association for computational linguistics: human language technologies*, 2016, pp. 1480–1489.

[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] 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, 2020.

[25] B. H. Lee, "Joga jisihaneun seonbeobgwa torieui gaenyeom(조(調)가 지시하는 선법과 토리의 개념)," *한국음악연구*, vol. 51, pp. 245–271, 2012.

[26] Y. W. Kim, *Introduction to Korean Traditional Music(국악개론)*. Eumaksekye(음악세계), 2020.

[27] D. H. Kang, "A study on the folksong's ecology and cultural meaning of chongson arari in korean folksong arirang(정선아라리의 민요생태와 문화적 의미)," *The Society Of Korean Folk Song(The Society Of Korean Folk Song(한국민요학*, vol. 23, pp. 257–288, 2008.

[28] Y. W. Kim, "A study of the formation of arirang(<아리랑> 형성과정에 대한 음악적 연구)," *한국문학과 예술*, vol. 7, pp. 5–55, 2011.

[29] ——, "Reexamination of eastern folk song tori -necessity to set the lower category of menari-tori and proposal(동부민요 토리의 재검토-동부민요 하층위 설정의 필요성을 중심으로)," *Studies in Korean music(한국음악연구)*, vol. 61, pp. 35–64, 2017.

[30] Y. B. Cho, "A study of the musical style of jeju folk songs, 濟州道 民謠의 音樂樣式 硏究," Ph.D. dissertation, the academy of Korean studies(한국정신문화연구원 한국학대학원), 1997.

[31] B. H. Lee, "Research methodology of jejudo tori based on traditional tori features(전통적 토리 특성에 의한 제주도 토리 연구 방법론)," *Korean Recording Studies(한국음반학)*, vol. 28, pp. 105–130, 2018.

[32] E. J. Shin, "A review and proposal of studies of the mode of the folk songs in cheju island(제주 민요 음조직에 대한 연구 검토 및 제언)," *The Society for Korean History Musicology(한국음악사학회)*, vol. 62, pp. 169–206, 2019.

SINGER IDENTITY REPRESENTATION LEARNING USING SELF-SUPERVISED TECHNIQUES

Bernardo Torres[*1] **Stefan Lattner**[2] **Gaël Richard**[1]
[1] LTCI, Telecom Paris, Institut Polytechnique de Paris
[2] Sony Computer Science Laboratories Paris
bernardo.torres@telecom-paris.fr

ABSTRACT

Significant strides have been made in creating voice identity representations using speech data. However, the same level of progress has not been achieved for singing voices. To bridge this gap, we suggest a framework for training singer identity encoders to extract representations suitable for various singing-related tasks, such as singing voice similarity and synthesis. We explore different self-supervised learning techniques on a large collection of isolated vocal tracks and apply data augmentations during training to ensure that the representations are invariant to pitch and content variations. We evaluate the quality of the resulting representations on singer similarity and identification tasks across multiple datasets, with a particular emphasis on out-of-domain generalization. Our proposed framework produces high-quality embeddings that outperform both speaker verification and wav2vec 2.0 pre-trained baselines on singing voice while operating at 44.1 kHz. We release our code and trained models to facilitate further research on singing voice and related areas.

1. INTRODUCTION

Singer representation learning is a complex task in Music Information Retrieval (MIR) that involves extracting a representation of a singer's voice, capturing their unique identity or vocal timbre. This task is closely related to singer recognition, which comprises two major tasks: singer identification and singer verification. The first aims to determine the singer of a given song from a fixed set of singers in the dataset, while the latter aims to determine if two audio excerpts come from the same singer or not. Singer representation learning has many potential applications, including retrieval tasks (such as retrieving songs with a similar singing voice), and providing singer embeddings for conditioning Singing Voice Synthesis (SVS) [1] and Singing Voice Conversion (SVC) systems [2].

*Work mostly conducted during an internship at Sony CSL Paris

© B, Torres, S. Lattner, and G. Richard. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).
Attribution: B, Torres, S. Lattner, and G. Richard, "Singer Identity Representation Learning using Self-Supervised Techniques", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

Singer recognition is related to speaker recognition, a well-established domain with vast literature. Historically, it has received a much greater interest in particular due to the need for authentication by voice in many telecommunications applications. Singing voice, however, is different from speech in several ways, typically containing a wider variance of phoneme duration, utterances, and a wider pitch range, which makes singer recognition more challenging. Moreover, the lack of large labeled datasets further restricts the development of data-driven approaches.

In this study, we investigate if speaker recognition models trained on labeled speech data can be applied to singing voice, and whether self-supervised learning (SSL) models trained on singing voice data can achieve comparable performance. We compare different self-supervised techniques, including SimCLR [3], Uniformity-Alignment [4], VICReg [5] and BYOL [6], trained on a large collection of isolated vocal tracks. We also explore high-frequency regions that are traditionally ignored in speech [7, 8] but might be present in singing voice by working in 44.1 kHz sampling rate. Finally, we evaluate the generalization capabilities of our models on out-of-domain data.

Our main contributions are as follows: 1. We perform singer representation learning experiments using self-supervised techniques, an area that few works have explored. 2. We train encoders that operate at 44.1 kHz on a large dataset of singing voice recordings. 3. We conduct an extensive evaluation of the obtained embeddings for singer identification and singer similarity tasks, comparing them with publicly available pre-trained speech baselines. 4. We measure the out-of-domain generalization capabilities of our models on four public datasets.

2. RELATED WORK

Singer recognition has traditionally relied on acoustic features such as Mel-frequency cepstral coefficients (MFCCs) or Line Spectral Frequencies (LSFs) to capture timbre [9–11]. Some approaches focus on singer identification on polyphonic music [12, 13], while others separate vocals from background [14, 15]. In speaker verification literature, time-invariant embeddings such as i-vector [16] or x-vector [17] have been extensively used, and the domain has shifted towards data-driven approaches using deep neural networks to encode acoustic features into a lower-dimensional representation that captures speaker charac-

teristics. Temporal aggregation is used to remove the time dimension, and these systems are usually optimized using speaker label infomation for classification or metric learning losses. Recent works have also explored SSL for speaker verification [18–21].

SSL has been successful in many domains, particularly with approaches such as SimCLR [3], MoCo [22], CPC [23], and BYOL [6]. In the audio domain, following the success in Computer Vision and Natural Language Processing (NLP), successful SSL models for speech include Wav2Vec 2.0 [24], HuBERT [25], and WavLM [26]. SSL has also been successful in learning general-purpose audio representations, with examples like COLA [27], CLAR [28], and CLMR [29].

While the idea of finding singer embeddings using contrastive approaches is not new [30], to the best of our knowledge, only one work has employed SSL for singer representations [31]. In their work, contrastive learning is used to acquire feature embeddings of singing voices using data augmentations that disturb a singer's identity to make the embeddings more attentive to timbre or technique. In contrast, our work explores different SSL techniques, focuses on out-of-domain testing, and evaluates on singer similarity as well as singer identification.

3. METHOD

3.1 Goal

Our objective is to obtain, from isolated vocal tracks, unique singer representations that capture the timbre of the singer's voice. These representations must satisfy three criteria: (I) clips from the same singer should have a higher average similarity than clips from different singers; (II) the representation should not be dependent on fundamental frequency or linguistic content variations; and (III) the representations should generalize well to out-of-domain data.

3.2 Overview

The ideal embedding space for singer representations should cluster elements of the same singer while also ensuring semantic consistency by placing similar voice timbres close to each other within the space [4]. In line with the criteria outlined in Section 3.1, we conducted experiments with various self-supervised techniques which force embeddings of similar input data to be close in the embedding space. We experimented with four frameworks: SimCLR [3], Uniformity-alignment [4], VICReg [5], and BYOL [6]. Although these frameworks share a common goal, they differ in their approach (see Section 3.3). We took great care in selecting appropriate data augmentations and used a diverse set of singing voice training data. In the current section, we describe the general training framework common to all our self-supervised experiments.

Data sampling: In our methodology, we use a COLA [27] approach to train our models by sampling audio segments on the fly, from a randomly drawn audio clip coming from a large database. We first extract two segments $(x, x') \in \mathbb{R}^N$ cropped randomly from the audio clip, called the anchor and positive segment. We obtain augmented views of both audio segments of the positive pair via a data augmentation module $\text{Aug}(\cdot)$ that operates in the waveform domain, resulting in an augmented positive pair $(x^{(1)}, x'^{(2)})$. We repeat this process B times for a batch size of B, obtaining a positive pair batch $(\mathbf{x}^{(1)}, \mathbf{x}^{(2)})$, with no repetition of audio clips during a training epoch. The superscript $'$ is further omitted for simplicity.

Model: Our proposed model takes raw audio waveforms sampled at 44.1 kHz as input. Firstly, we compute log-compressed mel-spectrogram features $m \in \mathbb{R}^{F \times L}$ on the fly using the *nnAudio* library [1]. Next, the encoder module $g(\cdot)$ maps the extracted mel-spectrograms to a latent representation $h' = g(m) \in \mathbb{R}^{H \times L}$. At this stage, adaptive average pooling is used to aggregate embedding vectors h' into time-invariant feature embeddings $h \in \mathbb{R}^H$. A projection layer $p(\cdot)$ maps h into a lower dimensional latent space $z = p(h) \in \mathbb{R}^D$ using a shallow neural network.

We denote the full model $f(\cdot)$ by stacking the acoustic feature extraction, encoder, and projection modules. During training, we encode the training batch and obtain projections $\mathbf{z} = f(\mathbf{x})$. After training is completed, we discard the projection layer and use only the feature embeddings h. The similarity between a pair of embeddings is computed using the cosine similarity.

Although there are many specialized speaker verification architectures in the speech domain [32, 33], we use the EfficientNet-B0 [34] architecture as the backbone for the encoder module and a single SiLU non-linearity followed by a fully-connected layer for the projection layer. The projections are ℓ_2 normalized.

3.3 Self-supervised frameworks

The core concept of all used approaches is to leverage big amounts of unlabeled data to build a good representation space by aligning similar elements (and possibly separating dissimilar ones). At training time, model $f(\cdot)$ acts in a Siamese setup by encoding both elements of the augmented pair $\mathbf{z}^{(1)} = f(\mathbf{x}^{(1)})$ and $\mathbf{z}^{(2)} = f(\mathbf{x}^{(2)})$. For BYOL, we have a separate encoder f' with the same architecture as f and we compute $\mathbf{z}^{(1)} = f(\mathbf{x}^{(1)})$ and $\mathbf{z}^{(2)} = f'(\mathbf{x}^{(2)})$. For all setups, we compute a loss function on the batch projections $\mathcal{L}(\mathbf{z}^{(1)}, \mathbf{z}^{(2)})$.

Contrastive Learning: We employ the contrastive loss called *NT-Xent* from SimCLR [3]. The loss maximizes the agreement between positive samples and pushes all other embeddings of the batch (the negative parts) away in the representation space. It does so by maximizing the cosine similarity (sim) between positive samples and minimizing the sum of similarities for all other pairs formed in the batch:

$$\mathcal{L}_{\text{cont}}(\mathbf{z}) = -\sum_i \log \frac{\exp(\text{sim}(\mathbf{z_i^{(1)}}, \mathbf{z_i^{(2)}})/\tau)}{\sum_{\mathbf{j} \neq \mathbf{i}} \exp(\text{sim}(\mathbf{z_i^{(1)}}, \mathbf{z_j^{(2)}})/\tau)}. \quad (1)$$

We decouple the term $\exp(\text{sim}(\mathbf{z_i^{(1)}}, \mathbf{z_i^{(2)}})/\tau)$ from the

[1] https://github.com/KinWaiCheuk/nnAudio

denominator of the original *NT-Xent* [3], which has been shown to make the SSL task easier for smaller batch sizes and less sensitive to the hyperparameter τ [35].

Uniformity-Alignment: Proposed in [4], Uniformity-Alignment aims to align similar examples and distribute elements uniformly in an ℓ_2 normalized embedding space. Instead of using a contrastive loss, the authors propose optimizing directly for these two properties, resulting in two loss functions: alignment ($\mathcal{L}_{\text{align}}$) and uniformity ($\mathcal{L}_{\text{unif}}$).

$$\mathcal{L}_{\text{align}}(\mathbf{z}^{(1)}, \mathbf{z}^{(2)}) = \frac{1}{N} \sum_i \left\| \mathbf{z_i^{(1)}} - \mathbf{z_i^{(2)}} \right\|^2, \quad (2)$$

$$\mathcal{L}_{\text{unif}}^k(\mathbf{z^{(k)}}) = \log \frac{1}{N} \sum_{i,j} \left(\exp(-t \| \mathbf{z_i^{(k)}} - \mathbf{z_j^{(k)}} \|^2) \right), \quad (3)$$

where $t = 2$ and $\mathcal{L}_{\text{unif}} = \sum_{k=1,2} \mathcal{L}_{\text{unif}}^k / 2$.

VICReg: VICReg [5] is an approach that attempts to maximize the information content of the learned embeddings. Three losses are proposed: the variance, invariance, and covariance losses. The invariance loss is the same as the alignment loss (see Equation 2). The variance regularization forces the standard deviation of a batch (in the dimension axis) to be close to the value μ, preventing *collapse* (when embedding dimensions become useless). Let $\mathbf{d_j}(\mathbf{z}) \in \mathbb{R}^B$ be the vector composed of the values of a batch $\mathbf{z}$ at dimension j. The variance regularization is:

$$\mathcal{L}_{\text{var}}(\mathbf{z}) = \frac{1}{D} \sum_{j=1}^{D} \max\left(0, \mu - S\left(\mathbf{d_j}(\mathbf{z}), \epsilon\right)\right), \quad (4)$$

where D is the number of dimensions of z_i, and S is the regularized standard deviation $S(x, \epsilon) = \sqrt{\text{Var}(x) + \epsilon}$.

The covariance regularization decorrelates the dimensions of the embedding, making them orthogonal:

$$\mathcal{L}_{\text{cov}}(\mathbf{z}) = \frac{1}{D_z} \sum_{\mathbf{i} \neq \mathbf{j}} (C(\mathbf{z}))_{\mathbf{i,j}}^2, \quad (5)$$

where $C(\mathbf{z}) = \frac{1}{N-1} \sum_{i=1}^{N} (\mathbf{z}_i - \bar{\mathbf{z}}) (\mathbf{z}_i - \bar{\mathbf{z}})^T$ is the covariance matrix of $\mathbf{z}$, and $\bar{\mathbf{z}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{z}_i$.

BYOL: Bootstrap Your Own Latent (BYOL) [6] employs two neural networks: the online and target networks. Both networks share the same architecture. In addition, BYOL employs an additional predictor network q which computes predictions $q(\mathbf{z})$. BYOL iteratively refines the representation of the online network by minimizing the mean squared error (MSE) between its predictions and the target's projections. If f and f' denote the online and target networks, respectively, the loss function $\mathcal{L}_{\text{BYOL}}$ on the projections $\mathbf{z}^{(1)} = f(\mathbf{x}^{(1)})$, $\mathbf{z}^{(2)} = f'(\mathbf{x}^{(2)})$ is:

$$\mathcal{L}_{\text{BYOL}}(\mathbf{z}^{(1)}, \mathbf{z}^{(2)}) = \frac{1}{N} \sum_i \left\| \mathbf{z_i^{(1)}} - q(\mathbf{z_i^{(2)}}) \right\|^2. \quad (6)$$

The target network f' is not trained using directly the gradients of $\mathcal{L}_{\text{BYOL}}$, but it is updated with an exponential moving average of the weights of the online network.

Corpus	Language	#Hours	#Singers	Type
VCTK [36]	English	44	110	Speech
NUS-48E [37]	English	1.91	12	Speech/Singing
VocalSet [38]	English	10.1	20	Singing
M4Singer [39]	Chinese	29.77	20	Singing

Table 1: Out-of-domain datasets used for testing.

4. EXPERIMENTS

4.1 Data

We used a large private corpus of professionally recorded singing voice data containing approximately 25,000 tracks, totaling 940 hours of audio data. The dataset consists of isolated vocals of re-recordings of popular songs by 5,700 artists and includes a variety of singing styles, voice types, lyrics, and audio effects. We note that the actual number of singers is unknown, as the same artist might have been re-recorded by multiple singers. Therefore, we do not believe that this corpus is appropriate for supervised training. Additionally, we added 6 hours of source-separated vocals to the corpus. All samples were converted to mono 44.1kHz tracks with 16-bit encoding, and any silence lasting more than 1.3 seconds was trimmed to 1.3 seconds. Segments with less than 0.5% amplitude were considered silent, and segments with more than 0.5% amplitude lasting less than 0.2 seconds were silenced. The dataset was partitioned into three distinct sets with ratios of 80% for training, 10% for validation, and 10% for testing, with no artist allocated to more than one set. The length of a track is typically a few minutes.

Out-of-domain evaluation: Four datasets are used to test the out-of-domain generalization of the models. The summary of all datasets is shown in Table 1.

4.2 Experiment setup

We perform a series of experiments to determine the best SSL framework for singer representation learning:

- `CONT`: We train a model on the decoupled version of the contrastive loss $\mathcal{L} = \mathcal{L}_{\text{cont}}$ [35].
- `CONT-VC`: We train a model using $\mathcal{L}_{\text{cont}}$ (contrastive loss) with added variance and covariance regularization $\mathcal{L} = \mathcal{L}_{\text{cont}} + \mu\mathcal{L}_{\text{var}} + \nu\mathcal{L}_{\text{cov}}$ [40].
- `UNIF`: We train a model using the uniformity- alignment loss $\mathcal{L} = \mathcal{L}_{\text{align}} + \gamma\mathcal{L}_{\text{unif}}$ [4].
- `VICReg`: We train a model using the VICReg loss $\mathcal{L} = \lambda\mathcal{L}_{\text{align}} + \mu\mathcal{L}_{\text{var}} + \nu\mathcal{L}_{\text{cov}}$ [5] .
- `BYOL`: We train a model on BYOL configuration, optimizing the MSE $\mathcal{L} = \mathcal{L}_{\text{BYOL}}$ [6].

The contrastive loss has been shown to yield good results in the literature [27, 31], but there is concern that it may break the semantic structure of the embeddings by pushing similar singers away in the representation space [4]. In the `CONT-VC` approach, the addition of variance

and covariance losses from VICReg is tested as a regularization method to mitigate this problem [21,40]. The UNIF approach attempts to optimize directly for uniformity of the space, which has shown links with linear separability [4] and potential for strong singer identification results. While VICReg claims to be an information-theoretic approach to general-purpose representation learning, it has not yet been thoroughly tested in the audio domain. Finally, BYOL is included in the study for comparison as it has shown promising results in several audio downstream tasks, claiming state-of-the-art [41].

4.3 Evaluation procedure

The models are first trained until the validation loss stops decreasing. Validation similarity metrics (Section 4.3.1) are tracked during training, and the best-performing model is selected. This model is evaluated on the test and on out-of-domain sets using cropped 4-second clips of singer recordings (with no overlapping segments). The embeddings h are evaluated in two tasks: singer/speech similarity and singer/speech identification. For simplicity, we refer to singer similarity/identification even when dealing with speech data such as with VCTK/NUS-48E datasets (see Section 5 for details).

4.3.1 Singer similarity

We evaluate singer similarity by measuring two metrics directly on the singer feature embeddings h: the Equal Error Rate (EER) and Mean Normalized Rank (MNR). The EER relates to singer verification. On the other hand, we relate the MNR to singer retrieval by computing the similarities between a query excerpt and a set of candidates. No training is performed for the similarity evaluation.

EER: The EER is a popular metric for verification systems. To compute the EER, the system is exposed to a set of trials consisting of true pairs (two segments coming from the same singer) and fake pairs (two segments coming from different singers), and a similarity metric is computed for both cases (in our case the cosine similarity). False positives (FP) and False Negatives (FN) can be computed by applying a threshold τ on the similarity metric, and the Detection Error Tradeoff (DET) is obtained by varying τ as a function of FP and FN. The EER is the error rate at which FP = FN. We compute the EER following the implementation available as part of the SupERB benchmark [42] [2]. We sample 50,000 speaker pairs for computing the EER on the test set and 20,000 speaker pairs for out-of-domain.

MNR: Denote $q^{(1)}$, $q^{(2)}$ two query audio samples, coming from the same audio recording (and therefore the same singer) drawn at random at each trial. Let S be a set of N audio samples, drawn at random from a dataset, and $q^{(2)} \in S$. The MNR is [40]:

$$\text{MNR} = \frac{1}{K}\sum_{k=1}^{K}\frac{R(q_k^{(1)}, S_k)}{N}, \tag{7}$$

[2] https://github.com/s3prl/s3p

Model	#Params	SR	Dim.	Backbone
GE2E [43] [3]	1.4M	16	256	LSTM
F-ResNet [33] [4]	1.4M	16	512	ResNet-34
H/ASP [44] [4]	8.0M	16	512	ResNet-34
Wav2Vec-base [24] [5]	95M	16	12X768	Wav2Vec 2.0
XLSR-53 [45] [6]	300M	16	24X1024	Wav2Vec 2.0
Ours	5.0M	44.1	1000	EfficientNet-B0

Table 2: Number of network parameters, sampling rate in kHz (SR), the size of the feature embeddings (Dim), and the architecture backbone for the baselines and our models.

where $R(q_k^{(1)}, S_k)$ is the integer position (rank) of $q^{(2)}$ in the sorted list of cosine similarities between $q^{(1)}$ and the samples in S. We perform $K = 1000$ trials for $N = 512$.

Input sample rate: To ensure a fair comparison with the baselines, which operate on 16 kHz, the evaluation is done in two scenarios: at 16 kHz and 44.1 kHz. In the former, the 44.1 kHz inputs are downsampled to 16 kHz and upsampled back to 44.1 kHz before being fed to the models, removing energy above 8 kHz. In the latter, the trained models have access to the full frequency range of the input data.

4.3.2 Singer identification

To evaluate the linear separability of singer classes, we perform singer classification as a downstream task for singer identification on out-of-domain evaluations. We use 5-fold cross-validation to split the audio files of each singer into train, validation, and test subsets (4-fold for NUS-48E). A single feed-forward linear layer is trained with cross-entropy loss on the train subset to predict singer classes from embeddings extracted from frozen models. The best model is selected on the validation subset. Average metrics on the test set over all folds are reported. We limit this task to out-of-domain evaluations since these datasets contain multiple files per singer and the classes are balanced.

4.4 Baselines

In our experiments, we use as baselines three speaker verification networks: GE2E [43], Fast-ResNet34 [33] (hereafter referred to as F-ResNet), H/ASP [44]; and two large general purpose self-supervised models Wav2Vec-base [24], and XLSR-53 [45]. These models have been pre-trained on speech and either achieved state-of-the-art results or have been used for obtaining speaker representations for speech/singing voice synthesis tasks while being publicly available. We provide an overview of the baseline models in Table 2.

Since all baselines operate on 16kHz, we downsample the test signals to 16kHz accordingly. For Wav2Vec-base and XLSR-53, we use adaptive average pooling as the temporal aggregation method for the frame-wise feature embeddings, and we employ a learned, weighted sum of the first three layers for the downstream

[3] https://github.com/resemble-ai/Resemblyzer
[4] https://github.com/clovaai/voxceleb_trainer
[5] https://huggingface.co/facebook/wav2vec2-base
[6] https://huggingface.co/facebook/wav2vec2-large-xlsr-53

Model	In-domain				Out-of-domain															
	Test dataset*				VCTK				NUS-48E				M4Singer*				Vocalset*			
	EER		MNR		EER		MNR		EER		MNR		EER		MNR		EER		MNR	
	16	44.1	16	44.1	16	44.1	16	44.1	16	44.1	16	44.1	16	44.1	16	44.1	16	44.1	16	44.1
GE2E†	27.24	-	18.9	-	13.42	-	5.41	-	28.04	-	18.99	-	25.01	-	15.99	-	40.45	-	35.34	-
F-ResNet†	15.21	-	7.76	-	1.01	-	**0.08**⋆	-	15.36	-	6.63	-	14.21	-	5.98	-	40.64	-	33.82	-
H/ASP†	12.36	-	5.82	-	**0.28**⋆	-	**0.08**⋆	-	**13.99**	-	**5.42**	-	12.31	-	3.93	-	36.27	-	30.79	-
Wav2Vec-base	25.36	-	14.78	-	23.15	-	15.78	-	32.65	-	24.39	-	26.28	-	13.37	-	39.34	-	34.23	-
XLSR-53	25.22	-	15.82	-	25.93	-	19.95	-	36.62	-	28.52	-	26.02	-	16.96	-	40.09	-	35.32	-
VICReg	8.19	3.88	2.29	1.14	25.17	23.88	14.99	14.62	26.11	26.06	15.43	15.34	24.6	22.05	9.78	8.69	34.58	33.12	28.21	26.5
UNIF	9.48	2.86	2.13	0.78	22.51	24.28	12.99	14.67	27.65	26.12	17.08	15.48	20.46	17.03	8.83	6.67	32.4	31.19	25.07	23.19
CONT	6.39	**2.16**	1.33	**0.48**	20.04	22.87	9.34	11.56	23.67	24.51	12.86	12.45	14.28	12.67	5.52	4.51	32.16	30.61	23.64	22.6
CONT-VC	7.39	2.74	1.61	0.52	19.92	21.79	10.35	11.12	24.99	25.4	15.06	13.91	15.97	12.68	6.94	4.81	31.03	**29.74**	22.65	21.87
BYOL	5.88	3.82	1.5	0.68	17.44	19.97	7.8	9.73	26.01	23.9	15.62	12.21	15.65	**12.28**	5.86	**3.77**	31.59	29.76	23.93	**21.25**

Table 3: EER and MNR (%, lower is better) measured on frozen model embeddings. Datasets that contain only singing voice are marked with *, and models which are not self-supervised are indicated with †. Results in bold are the best among all models, for both 44.1 kHz and 16 kHz input sample rates. Underlined results highlight the best on 16kHz input only. For Wav2Vec-base and XLSR-53, we use the embeddings of the first layer and aggregate them using average pooling.

classifier [42]. We empirically found that this approach boosts classification performance compared to using a single layer. Specifically, the first layers of these models are more effective for speaker verification [26] and are more correlated with speaker characteristics [46]. For singer similarity evaluations, we use only the first layer, as there is no training involved to yield weights for a weighted sum.

4.5 Training

To train our models, we used 4-second audio clips that were normalized, augmented, and converted to log-compressed mel-filterbanks with 80 mel bins, a window length of 2048, and a hop size of 512. This results in an FFT frame of 46.4ms and sliding windows of 11.6ms for 44.1 kHz audio. We initialized the EfficientNet-B0 backbone with pre-trained weights on ImageNet [40] and used the ADAM optimizer with a learning rate of 1e-4 and weight decay of 1e-5, with a batch size of 120. For contrastive loss, we used a temperature parameter of $\tau = 0.2$ [4], and whenever we used covariance regularization, we set $\nu = 100$. For variance regularization, we set $\mu = 25$. Additionally, for VICReg experiments, we used an invariance loss factor of $\lambda = 25$, and UNIF, we set $\gamma = 1$. For BYOL, we used a learning rate of 3e-5, a weight decay of 1.5e-6 and an initial moving average value τ of 0.99. We found through empirical analysis that these hyperparameters were effective for convergence and avoiding collapse.

In terms of data augmentation techniques, we applied Gaussian noise, gain with a minimum attenuation of -6 dB, and time masking with at most 1/8 of the clip being masked. We also used formant-preserving pitch shifting with Praat [47, 48] as a method of data augmentation, with the pitch shift ratio and pitch range ratio being sampled uniformly from U(1,3) and U(1,1.5), respectively, with a random choice on whether to take the reciprocal of the sampled ratios or not [46]. All augmentations had a probability of 0.5 of being applied. We avoided using naive pitch-shifting techniques that transpose the formants, which can significantly alter the singers' timbre.

Model	VCTK	NUS-48E	M4Singer*	Vocalset*
GE2E†	97.01	91.13	88.72	45.66
F-ResNet†	99.91	97.36	94.51	49.52
H/ASP†	**99.93**	**98.32**	97.87	74.65
Wav2Vec-base	98.70	96.16	96.52	79.19
XLSR-53	99.66	97.02	**98.62**	**86.05**
VICReg	52.52	78.98	87.34	49.69
UNIF	74.43	93.05	93.55	67.52
CONT	90.24	96.23	95.72	77.42
CONT-VC	86.03	95.14	94.69	75.20
BYOL	96.95	96.56	97.00	81.01

Table 4: Average linear classification accuracy on out-of-domain data (%) over K-fold cross-validation. Datasets that contain only singing voice are marked with *. The best scores are highlighted in bold and the best among the trained models (bottom 5 rows) are underlined. Models which are not self-supervised are indicated with †.

5. RESULTS AND DISCUSSION

Table 3 presents the results of singer similarity evaluation on both in-domain and out-of-domain test sets, reporting the best Equal Error Rate (EER) and Mean Normalized Rank (MNR) for trained models and baselines in all test datasets. Table 4 shows the accuracies for downstream singer identification task on out-of-domain datasets. We also share in supplementary material additional qualitative visual evaluations of the embeddings [7], and release code and models to encourage reproducibility and facilitate its use in future projects [8].

5.1 Results on pre-trained models on speech

The results indicate that models pre-trained on speech in a supervised manner (using speaker labels) exhibit good generalization to out-of-domain speech datasets. H/ASP achieves an impressive 0.28% EER on the VCTK, and all models score higher than 88% accuracy on VCTK, NUS-

[7] https://sites.google.com/view/singer-representation-learning
[8] https://github.com/SonyCSLParis/ssl-singer-identity

48E, and M4Singer datasets. Their similarity performance on singing voice datasets, however, is much worse than on speech, but the best models still score below 10% EER on NUS-48E and 12.31% and 14.21% EER on M4Singer for H/ASP and F-ResNet, respectively.

This suggests that important features of the singing voice can also be learned directly from speech. However, the results show the pre-trained models perform worse on heavily processed data that includes uncommon effects and vocal techniques. This is evident, in particular, in the last columns of Tables 3 and 4 (VocalSet), with all baselines scoring around 40% EER and from 20% to 40% worse accuracy when compared to the other datasets.

5.2 Results on Self-Supervised Models

Models trained with contrastive loss (CONT and CONT-VC) achieved the best EER and MNR on the test set. These models were able to learn highly discriminative features for the task of in-domain singer similarity. For instance, the CONT model had the lowest overall EER and MNR (2.16% and 0.48% respectively) on the test set.

It can also be seen in Table 3 that in-domain test performance did not necessarily translate to good generalization to out-of-domain data. By adding variance and covariance regularizations (CONT-VC), the model achieved better generalization to out-of-domain data on some datasets (such as the VocalSet, with approximately 1% EER difference). However, in the VICReg scenario, which has both regularizations but lacks the contrastive part, the results were worse. In fact, VICReg had the worst overall results of all the tested self-supervised frameworks. UNIF, while better than VICReg, also performed worse on average when compared to the other approaches.

CONT and BYOL achieved the best accuracy over all our trained models on singer identification (Table 4), achieving the highest scores of 77.42% and 81.01%, respectively (the VocalSet paper [38] reports 60-70% accuracy on a supervised singer identification task).

BYOL achieved the best generalization on similarity, performing best on out-of-domain data, even though its scores were worse on the in-domain test set. Interestingly, of all explored self-supervised techniques, BYOL is the only one that does not explicitly force any kind of feature distribution on the embedding space. In addition, BYOL was able to learn best how to leverage the information present at 16 kHz sample rate, with an EER of 5.88 on the test set. It also performed best on out-of-domain speech data (VCTK). In general, our models struggled with speech, performing generally better when they only had access to a reduced frequency band. This suggests that in speech, high-frequency information the models rely on hinders their ability to generalize.

44.1 vs 16 kHz: Using 44.1 kHz inputs consistently improved the similarity results on singing voice datasets (e.g., M4Singer) for all models, highlighting the models' ability to efficiently use high-frequency information. Moreover, most models showed a marked decline in the in-domain dataset results when tested with 16 kHz inputs (the CONT model, for example, shows a drop from 2.16% EER to 6.39% EER). While the 16 kHz inputs could be considered out-of-domain, this effect shows that high-frequency information is important for the trained models to achieve better performance.

Comparison to baselines: The trained models show better results than baselines on in-domain test sets and the VocalSet dataset for singer similarity tests, although they fall behind F-ResNet and H/ASP on the mixed speech/singing dataset NUS-48E and VCTK. Nonetheless, on M4Singer, some self-supervised models outperformed the supervised baselines, with BYOL showing the best performance (12.28% EER and 3.77% MNR), and CONT and CONT-VC also being superior to F-ResNet.

The trained models have substantially better singer similarity results compared to Wav2Vec-base and XLSR-53. These results indicate the potential of training models on the proposed SSL tasks specifically on singing voice data. Further improvements could be made by fine-tuning the embeddings on verification tasks, as has been demonstrated in previous work on Wav2Vec 2.0 [49].

Moreover, BYOL outperformed Wav2Vec-base for both VocalSet and M4Singer on classification. Among all models, XLSR-53 achieved the best overall performance for singer identification of singing voice. However, is noteworthy that our models have significantly fewer parameters than the self-supervised Wav2Vec-base (19 times less) and XLSR-53 (63 times less).

6. CONCLUSION

In conclusion, we have shown that self-supervised learning is an effective approach for learning representations of singers. The self-supervised models trained on a large corpus of singing voice data demonstrated a performance that either matched or surpassed publicly available supervised speech models, without resorting to specialized architecture designs. Additionally, our models outperformed general-purpose self-supervised counterparts even with a significantly reduced parameter count. When applied to singer identification, our models exhibited superior performance over Wav2vec-base on singing voice datasets but fell somewhat short in comparison to the considerably more expansive XLSR-53.

Furthermore, our results suggest that these models hold promise for singer identification and similarity downstream tasks. BYOL showed the most promise for generalizing to out-of-domain data, while the contrastive approaches were more effective for in-domain data.

However, we note that our models' representations do not yet fully capture a singer's identity when confronted with unique singing techniques, such as those found in the VocalSet [38]. This underscores the need for further research on robust SSL frameworks capable of accommodating such variations. Our findings also suggest that employing a higher sampling frequency can be advantageous for singing voice tasks, but optimal frequency for generalizing to both singing and speech tasks remains to be determined.

7. ACKNOWLEDGMENTS

This work was partly funded by the European Union (ERC, HI-Audio, 101052978). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them.

The authors also thank Alain Riou for the BYOL implementation.

8. REFERENCES

[1] S. Wang, J. Liu, Y. Ren, Z. Wang, C. Xu, and Z. Zhao, "MR-SVS: Singing voice synthesis with multi-reference encoder," *CoRR*, vol. abs/2201.03864, 2022.

[2] S. Nercessian, "End-to-End Zero-Shot Voice Conversion Using a DDSP Vocoder," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, Oct. 2021, pp. 1–5.

[3] 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.

[4] F. Wang and H. Liu, "Understanding the Behaviour of Contrastive Loss," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Nashville, TN, USA: IEEE, Jun. 2021, pp. 2495–2504.

[5] A. Bardes, J. Ponce, and Y. LeCun, "VICReg: Variance-invariance-covariance regularization for self-supervised learning," in *ICLR*, 2022.

[6] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. Á. Pires, Z. 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 *NeurIPS*, 2020.

[7] S. Ternström, "Hi-Fi voice: Observations on the distribution of energy in the singing voice spectrum above 5 kHz," in *Acoustics' 08, Paris, France, Jun 29-Jul 4, 2008*, 2008, pp. 3171–3176.

[8] B. B. Monson, E. J. Hunter, A. J. Lotto, and B. H. Story, "The perceptual significance of high-frequency energy in the human voice," *Frontiers in psychology*, vol. 5, p. 587, 2014.

[9] Md. Sahidullah, S. Chakroborty, and G. Saha, "On the use of perceptual Line Spectral pairs Frequencies and higher-order residual moments for Speaker Identification," *IJBM*, vol. 2, no. 4, p. 358, 2010.

[10] L. Regnier and G. Peeters, "Singer verification: Singer model .vs. song model," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Kyoto, Japan: IEEE, 2012, pp. 437–440.

[11] T. Nakano, K. Yoshii, and M. Goto, "Vocal timbre analysis using latent Dirichlet allocation and cross-gender vocal timbre similarity," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 5202–5206.

[12] A. Mesaros, T. Virtanen, and A. Klapuri, "Singer identification in polyphonic music using vocal separation and pattern recognition methods." in *Proc. of the 8th International Society for Music Information Retrieval Conference (ISMIR)*, 2007, pp. 375–378.

[13] M. Lagrange, A. Ozerov, and E. Vincent, "Robust singer identification in polyphonic music using melody enhancement and uncertainty-based learning," in *Proc. of the 13th International Society for Music Information Retrieval Conference (ISMIR)*, 2012.

[14] B. Sharma, R. K. Das, and H. Li, "On the Importance of Audio-Source Separation for Singer Identification in Polyphonic Music," in *Interspeech 2019*. ISCA, Sep. 2019, pp. 2020–2024.

[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 *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 1–5.

[16] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2010.

[17] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-Vectors: Robust DNN Embeddings for Speaker Recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, Apr. 2018, pp. 5329–5333.

[18] J.-w. Jung, Y. J. Kim, H.-S. Heo, B.-J. Lee, Y. Kwon, and J. S. Chung, "Pushing the limits of raw waveform speaker recognition," in *Interspeech*. ISCA, 2022, pp. 2228–2232.

[19] M. Sang, H. Li, F. Liu, A. O. Arnold, and L. Wan, "Self-supervised speaker verification with simple siamese network and self-supervised regularization," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 6127–6131.

[20] W. Xia, C. Zhang, C. Weng, M. Yu, and D. Yu, "Self-supervised text-independent speaker verification using prototypical momentum contrastive learning," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6723–6727.

[21] T. Lepage and R. Dehak, "Label-efficient self-supervised speaker verification with information maximization and contrastive learning," in *Proc. Interspeech 2022*. ISCA, Sep. 2022, pp. 4018–4022.

[22] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9729–9738.

[23] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[24] 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, pp. 12 449–12 460, 2020.

[25] 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.

[26] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao *et al.*, "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, 2022.

[27] A. Saeed, D. Grangier, and N. Zeghidour, "Contrastive learning of general-purpose audio representations," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 3875–3879.

[28] H. Al-Tahan and Y. Mohsenzadeh, "Clar: Contrastive learning of auditory representations," in *International Conference on Artificial Intelligence and Statistics*, 2021, pp. 2530–2538.

[29] J. Spijkervet and J. A. Burgoyne, "Contrastive learning of musical representations," in *Proc. of the 22nd International Society for Music Information Retrieval Conference (ISMIR)*, 2021, pp. 673–681.

[30] C.-i. Wang and G. Tzanetakis, "Singing Style Investigation by Residual Siamese Convolutional Neural Networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2018, pp. 116–120.

[31] H. Yakura, K. Watanabe, and M. Goto, "Self-Supervised Contrastive Learning for Singing Voices," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 30, pp. 1614–1623, 2022.

[32] B. Desplanques, J. Thienpondt, and K. Demuynck, "ECAPA-TDNN: Emphasized channel attention, propagation and aggregation in TDNN based speaker verification," in *Interspeech*. ISCA, 2020, pp. 3830–3834.

[33] J. S. Chung, J. Huh, S. Mun, M. Lee, H. S. Heo, S. Choe, C. Ham, S. Jung, B.-J. Lee, and I. Han, "In defence of metric learning for speaker recognition," in *Interspeech 2020*, Oct. 2020, pp. 2977–2981.

[34] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International Conference on Machine Learning*, 2019, pp. 6105–6114.

[35] C.-H. Yeh, C.-Y. Hong, Y.-C. Hsu, T.-L. Liu, Y. Chen, and Y. LeCun, "Decoupled contrastive learning," in *ECCV (26)*, ser. Lecture Notes in Computer Science, vol. 13686. Springer, 2022, pp. 668–684.

[36] J. Yamagishi, C. Veaux, and K. MacDonald, "CSTR VCTK Corpus: English Multi-speaker Corpus for CSTR Voice Cloning Toolkit (version 0.92)," 2019.

[37] Z. Duan, H. Fang, B. Li, K. C. Sim, and Y. Wang, "The NUS sung and spoken lyrics corpus: A quantitative comparison of singing and speech," in *APSIPA*. IEEE, 2013, pp. 1–9.

[38] J. Wilkins, P. Seetharaman, A. Wahl, and B. Pardo, "VocalSet: A Singing Voice Dataset." in *Proc. of the 19th International Society for Music Information Retrieval Conference (ISMIR)*, 2018, pp. 468–474.

[39] 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," *Advances in Neural Information Processing Systems*, vol. 35, pp. 6914–6926, 2022.

[40] S. Lattner, "SampleMatch: Drum sample retrieval by musical context," in *Proc. of the 23rd International Society for Music Information Retrieval Conference (ISMIR)*, 2022, pp. 781–788.

[41] D. Niizumi, D. Takeuchi, Y. Ohishi, N. Harada, and K. Kashino, "BYOL for audio: Exploring pre-trained general-purpose audio representations," *IEEE ACM Trans. Audio Speech Lang. Process.*, vol. 31, pp. 137–151, 2023.

[42] S.-W. Yang, P.-H. Chi, Y.-S. Chuang, C.-I. J. Lai, K. Lakhotia, Y. Y. Lin, A. T. Liu, J. Shi, X. Chang, G.-T. Lin, T.-H. Huang, W.-C. Tseng, K.-t. Lee, D.-R. Liu, Z. Huang, S. Dong, S.-W. Li, S. Watanabe, A. Mohamed, and H.-y. Lee, "SUPERB: Speech processing universal PERformance benchmark," in *Interspeech*. ISCA, 2021, pp. 1194–1198.

[43] L. Wan, Q. Wang, A. Papir, and I. L. Moreno, "Generalized end-to-end loss for speaker verification," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4879–4883.

[44] Y. Kwon, H.-S. Heo, B.-J. Lee, and J. S. Chung, "The ins and outs of speaker recognition: Lessons from VoxSRC 2020," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 5809–5813.

[45] A. Conneau, A. Baevski, R. Collobert, A. Mohamed, and M. Auli, "Unsupervised cross-lingual representation learning for speech recognition," in *Interspeech*. ISCA, 2021, pp. 2426–2430.

[46] H.-S. Choi, J. Lee, W. Kim, J. Lee, H. Heo, and K. Lee, "Neural analysis and synthesis: Reconstructing speech from self-supervised representations," *Advances in Neural Information Processing Systems*, vol. 34, pp. 16 251–16 265, 2021.

[47] P. Boersma and D. Weenink, "Praat: Doing phonetics by computer (Version 5.1.13)," 2009.

[48] Y. Jadoul, B. Thompson, and B. de Boer, "Introducing Parselmouth: A Python interface to Praat," *Journal of Phonetics*, vol. 71, pp. 1–15, Nov. 2018.

[49] Z. Fan, M. Li, S. Zhou, and B. Xu, "Exploring wav2vec 2.0 on Speaker Verification and Language Identification," in *Interspeech 2021*. ISCA, Aug. 2021, pp. 1509–1513.

ON THE EFFECTIVENESS OF SPEECH SELF-SUPERVISED LEARNING FOR MUSIC

Yinghao Ma [,1*] **Ruibin Yuan** [,2,3*] **Yizhi Li** [,4*] **Ge Zhang** [,2,5*] **Xingran Chen** [6] **Hanzhi Yin**[3] **Chenghua Lin**[4†]
Emmanouil Benetos[1†] **Anton Ragni**[4] **Norbert Gyenge**[4] **Ruibo Liu**[7] **Gus Xia**[8] **Roger Dannenberg**[3] **Yike Guo**[9] **Jie Fu**[2†]

[] Multimodal Art Projection Research Community [1] Queen Mary University of London [2] Beijing Academy of Artificial Intelligence
[3] Carnegie Mellon University [4] University of Sheffield [5] University of Waterloo [6] University of Michigan Ann Arbor
[7] Dartmouth College [8] New York University Shanghai [9] Hong Kong University of Science and Technology

{yinghao.ma, emmanouil.benetos}@qmul.ac.uk, ruibiny@andrew.cmu.edu
{yizhi.li, c.lin}@sheffield.ac.uk, gezhang@umich.edu, fujie@baai.ac.cn

ABSTRACT

Self-supervised learning (SSL) has shown promising results in various speech and natural language processing applications. However, its efficacy in music information retrieval (MIR) still remains largely unexplored. While previous SSL models pre-trained on music recordings may have been mostly closed-sourced, recent speech models such as wav2vec2.0 have shown promise in music modelling. Nevertheless, research exploring the effectiveness of applying speech SSL models to music recordings has been limited. We explore the music adaption of SSL with two distinctive speech-related models, data2vec1.0 and Hubert, and refer to them as music2vec and musicHuBERT, respectively. We train 12 SSL models with 95M parameters under various pre-training configurations and systematically evaluate the MIR task performances with 13 different MIR tasks. Our findings suggest that training with music data can generally improve performance on MIR tasks, even when models are trained using paradigms designed for speech. However, we identify the limitations of such existing speech-oriented designs, especially in modelling polyphonic information. Based on the experimental results, empirical suggestions are also given for designing future musical SSL strategies and paradigms.

1. INTRODUCTION

Deep learning (DL) techniques have shown promising results in a wide range of auditory tasks, including speech and music information retrieval (MIR). However, the quantity and quality of labelled data is a bottleneck for developing algorithms with better generalisation in complex real-world settings for machine listening. To address this issue, self-supervised learning (SSL) such as BERT [1] has emerged as a solution to leverage diverse and representative unlabelled data to train a deep feature extractor with better generalisation. By combining this pre-trained SSL encoder with a naive classifier, typically a multi-layer perceptron (MLP) or long short-term memory (LSTM) with limited hidden layers, the model can achieve strong or state-of-the-art (SOTA) performance in various downstream tasks including NLP [1–3], computer vision [4], and audio [5, 6], where well-labelled datasets are limited. For music, larger datasets can be more expensive due to copyright and annotation costs, making SSL essential for developing effective MIR systems. Investigating versatile SSL approaches in MIR can further improve the performance on many MIR tasks, benefitting the music industry, music education, and heritage preservation. Although SSL has significantly improved the performance of models in tasks such as speech recognition, sentiment analysis, and language modelling, its effectiveness in MIR remains largely unexplored.

There has been much work on SSL for audio representation learning, including speech, sound events or music. But most results are difficult to evaluate or fine-tune due to limited access to training data, pre-trained parameters or training codes. PANN [7] is trained on noisy/weak-label classification and does not provide promising results in music tasks such as pitch classification and instrument classification [8]. Besides, it can hardly be re-trained on music datasets given that the MIR community does not have a weekly labelled large music dataset. MusiCoder [9], Music PASE [10], and MAP-Music2Vec [11] use strategies mainly based on masked prediction, where training models predict the audio waveform manually-designed feature or learnable deep feature of input removed randomly from the ground truth. Such models trained on music are not open-sourced except MAP-Music2Vec, which provides pre-trained parameters on hugging-face [1] . Jukebox [12] uses similar strategies for pop-song recording generation and demonstrates good potential for multiple MIR tasks [6]. But the training code for it is unavailable and is hard to fine-tune given its 6 billion parameters. MAP-MERT v0 [13] mimicks HuBERT [14], which regards the clustering results of audio as a pseudo label or pseudo spectrum to be reconstructed rather

*The authors contributed equally to this work.
† Corresponding authors.

© Y. Ma, R. Yuan, Y. Li, G. Zhang et al.. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).
Attribution: Y. Ma, R. Yuan, Y. Li, G. Zhang et al., "On the Effectiveness of Speech Self-Supervised Learning for Music", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

[1] https://huggingface.co/m-a-p/music2vec-v1

than a cluster assignment. But it does not provide training codes for further model evaluation. Furthermore, there are some music SSL models based on instance discrimination. In this family of approaches, each instance is considered its class, and models are trained to distinguish among different instances. CLMR [15] is trained with a limited number of parameters and shows limited capacity [6]. PEMR [16] does not show promising results besides tagging and is not open-source for further evaluation.

Although not designed for MIR tasks, some speech SSL models provide promising results on music tasks, and their training codes are available for fine-tuning or re-training on musical audio. Mockingjay [17], and PASE [18] use masked waveform / audio-feature prediction for pre-training. COLR [19] uses EfficientNet with a limited number of parameters and is designed for general audio, though it has a promising result on instrument classification. SF-NFNet-F0 [20] also uses an architecture based on convolution neural networks, a SlowFast Normalizer-Free ResNet, for audio pre-training. Furthermore, apart from providing good results on automatic speech recognition (ASR), Wav2Vec2.0 [5], HUBERT and data2vec [21] also provide much better results on pitch estimation and instrumental classification than PANN, though they are still far from perfect [8]. All of the speech SSL models are helpful for music SSL model development.

Previous work on re-training speech SSL systems with music recordings is limited to the size of training datasets or model structure. Ragano et al. [22] re-trained wav2vec2.0 on music audio and improved performance on pitch estimation and instrument classification significantly. But the training set is less than 100 hours which may be less representative, and the downstream tasks are limited and not universal. MusiCoder and Music PASE can be regarded as re-training speech SSL models on music recordings, but the model performance is not promising. Besides, these models are evaluated with a limited number of downstream tasks, making the learned embedding less persuasive. SF-NFNet-F0 is trained on music recordings and provides better results on multiple music tagging tasks [23]. But its model architecture is based on CNN, without much room for further scale-up and longer sequence modelling.

The missing science in the previous studies is as follows. All of the existing models trained on music are either with a limited number of parameters and capacity for MIR tasks other than tagging or not open-source for further evaluation. Some of the systems developed on speech or general audio recordings demonstrate promising but not satisfying results on MIR tasks. Besides, previous investigations on the efficacy of applying speech-related SSL models to music recordings are limited by the size of the training set, not enough universality on the downstream tasks, or paying less attention to powerful transformer structures.

Our key contributions are four-fold: (1) exploring two speech-related SSL models based on transformer structures, data2vec and HuBERT, and comparing the results with those models pre-trained in speech recordings; (2) carrying out ablation studies for pre-training, thus providing more intuition for further music SSL system design; and (3) systematically comparing the performance on 13 downstream tasks, which facilitates comprehensive model evaluation on a wide range of MIR tasks.

2. METHOD

In order to keep the pre-training and representation evaluation protocols comparable, we focus on adapting from the speech self-supervised learning frameworks that support direct audio input and end-to-end pre-training. Given our intent of exploring the influence of the pre-training design itself, we choose two SSL frameworks mainly distinguished by their self-supervised learning targets while sharing very similar training settings, including model architecture, training datasets, and evaluation protocols. In this section, we briefly describe the two selected SSL models – data2vec-1.0 [21] and HuBERT [14] – in the unified auto-encoding framework (cf. Fig. 1) and discuss the similarities and differences under music audio pre-training.

2.1 Music2Vec: Continuous Target Prediction

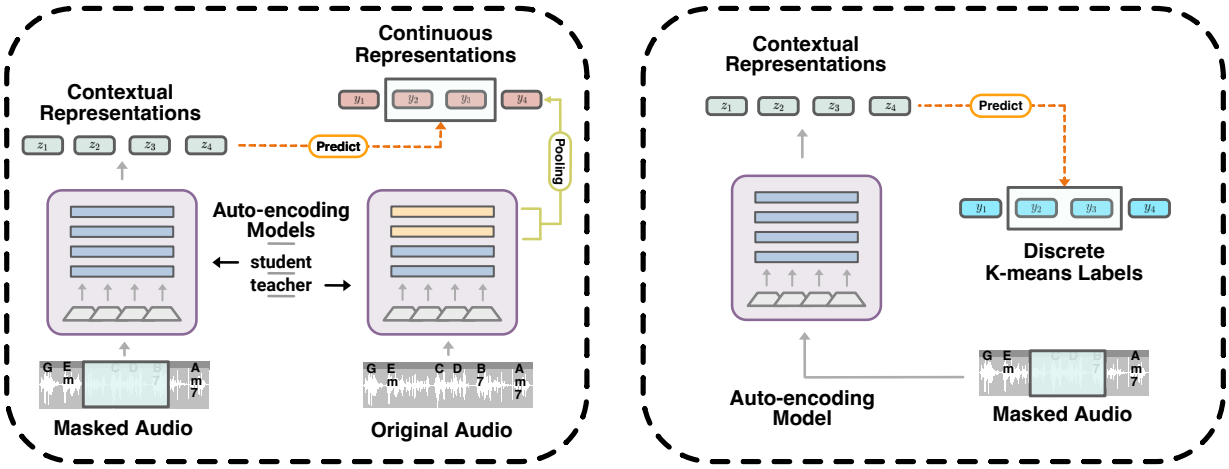
We adapt the pre-training paradigm from the speech version of the multi-modal framework data2vec-1.0 [21], where the prediction targets during pre-training are continuous representations. We refer to this continuous prediction model adapted with music recordings as Music2Vec.

Modified from the design of bootstrap your own latent (BYOL) [24], Music2Vec aims to predict continuous latent representations from the teacher model for the masked input audios, which is illustrated in Fig. 1a. The teacher model and student model share the same architecture, and the parameters of the teacher model are updated according to the exponential moving average of the student [21]. The student model takes the partially masked input and is asked to predict the average pooling of top-K layer outputs from the Transformer [25] in the teacher model. In contrast, the teacher model takes the unmasked input and provides contextual prediction targets in the pre-training.

Following the data2vec [5] setting, we train the Music2Vec of 95M parameters with a comparable 1k hours of music recordings. Since pre-trained speech models can barely benefit music representation learning [22], we instead train the base model from scratch to verify its effectiveness in modelling music audio recordings.

2.2 MusicHuBERT: Discrete Target Prediction

Another efficient speech SSL model, HuBERT [14], is chosen as the representative of discrete target prediction design. We referred to the music adaption version as MusicHuBERT. It takes masked music audios as input (Similar to Music2Vec) and predicts pre-processed discrete labels corresponding to the masked area, as shown in Fig. 1b. The discrete targets are pseudo labels provided by K-means that are trained on the MFCC features of the training audios. The number of clusters K of the K-means model is a hyperparameter, and all the centroids are assigned with randomly initialised embeddings and learned during the



(a) Music2Vec (b) MusicHuBERT

Figure 1: Pre-training Paradigms of Selected Models. Both of the models are fed with masked audio inputs and predict given targets without supervised information.

MusicHuBERT pre-training. MusicHuBERT can also be trained for an extra n iterations, where K-means clustering is learned from model outputs' previous iteration. We follow the original HuBERT [14] setting to train a model with 95M parameters of the same size as Music2Vec.

2.3 similarities & Differences of SSL frameworks

This subsection will examine the similarities and differences between the SSL frameworks mentioned above.

Both Music2Vec and MusicHuBERT are annotation-free and utilise SSL techniques; their most common characteristic is the training task of "reconstructing" information from masked inputs, making them auto-encoding models. During the denoising process, these models learn the semantics contained in the audio. Furthermore, they share similar model architecture designs, which are inherited from wav2vec-2.0 [5], wherein the audio is initially encoded by a multi-layer 1-D CNN feature extractor that maps a 16 kHz waveform to 50 Hz representations. The encoded tokens are then fed into a 12-layer transformer block with a hidden dimension of $H = 768$.

Regarding the differences in the designs, the most notable one is that Music2Vec is required to predict continuous latent variables, whereas MusicHuBERT predicts discrete pseudo-labels. The time cost of SSL target preparation bottleneck varies according to their mechanism. In Music2Vec, the pre-training consumes twice the model forward time since the target representations from the teacher model are inferred on-the-fly. In contrast, MusicHuBERT trains the K-means model and infers all the pseudo-labels before training, which requires high parallel processing ability when the dataset is scaled-up.

3. DATASET & EVALUATION

3.1 Training

We use a private dataset with 1000 hours of music audio recordings for pre-training; each sample is a 30s-long excerpt from pop-song or instrumental music. The size of the pre-training dataset is roughly the same as the pre-training for HuBERT-base and data2vec-audio-base models.

3.2 Evaluation

We evaluate the models on 13 downstream tasks, including timbre classification tasks such as genre and instrumental classification, singing, playing technique classification, singer classification, and music tagging; emotion-related tasks like music mood classification and regression; and note-related tasks such as pitch estimation, key detection; and sequential tasks like beat tracking.

Music Tagging is a multi-label classification task. We used MagnaTagATune (MTT) [26] and MTG-Jamendo [27] for this task, tag categories of which include genre, instrumentation, mood, and tempo (e.g. fast) etc. For both datasets, we limit the tag vocabulary to the 50 most common tags. We use all clips in MTT and MTG-Jamendo for evaluation. Since many of the audio recordings among 5.5k MTG-Jamendo excerpts are longer than the 30s, we averaged the multiple embeddings computed with a 30s sliding window as the overall embedding. The metrics are the macro-average of ROC-AUCs and the average precision (AP) / PR-AUC among all top-50 tags.

Key detection. We use a commonly-used subset of Giantsteps-MTG-keys [28] as the training and validation set following the data splitting [6], and Giantsteps (GS) [29] as the test set. The metric is a refined accuracy that gives partial credit to reasonable errors [30].

Genre classification. We report the multi-class classification accuracy of the GTZAN [31] dataset, along with ROC and AP on MTG-Genre for multi-label. We used the standard "fail-filtered" split [32] for GTZAN.

Emotion score regression. The Emomusic dataset [33] contains 744 music clips of 45 seconds, each reported on a 2-D valence-arousal plane after listening. We use the same dataset split as [6]. The evaluation metric is the determination coefficient (r^2) between the model regression results

and human annotations of arousal (EmoA) and valence (EmoV) [33]. We split the 45-second clip into a 5-second sliding window for inference and averaged the prediction.

Instrument classification. We use the Nsynth [34] and MTG-instrument datasets. The former is a multi-class task on 306k audio samples in 11 instruments with accuracy as an indicator. The latter is a subset of MTG-Jamendo, containing 25k audio tracks and 41 instrument tags; each track can contain multiple instruments and is evaluated on ROC and AP.

Pitch classification. Given these audios are short monophonic audio, this task is multi-class to determine which of the 128 pitch categories, and the accuracy is used as an evaluation metric.

Vocal technique detection. We use the VocalSet dataset [35], which is the only publicly available dataset for the study of singing techniques. The dataset contains the vocals of 17 different singing techniques in various contexts for a total of 10.1 hours. As the audio clips are divided into 3 seconds, the task only requires a judgment on the type of technique and not on the start and end of the technique. We used the same 10 different singing techniques as in [36] as a subset and used the same 15 singers as the training and validation sets and 5 singers as the test set. Since there is no accepted division between training and validation sets, we selected 9 singers as the training set and 6 singers as the validation set. All the 3-second segments originate from the same recording are allocated to the same part of the split (e.g. all in the testing set).

Singer identification is to identify the vocal performer from a given recording. We randomly divided the VocalSet dataset, which contains 20 different professional singers (9 female and 11 male), into a training set, validation set and testing set based on a ratio of 12:8:5, all containing the same 20 singers.

Beat tracking. We use an offline approach to the binary classification, i.e. the model can use the following information from each frame to help with inference. The model needs to output frame-by-frame predictions at a certain frequency and post-process them using a dynamic Bayesian network (DBN) [37], the same methods with supervised SOTA. The DBN is implemented using `madmom` [38]. The dataset we use is GTZAN Rhythm [39]. We also label the two adjacent frames of each label as beat, a common way of smoothing in beat tracking. The model is evaluated using the f_measure implemented in `mir_eval` [30], and the prediction is considered correct if the difference between the predicted event and the ground truth does not exceed 20ms. In this task, some models were trained on other datasets, and the full GTZAN set was used as the test set. For all cases, however, we use GTZAN-train as the training set and GTZAN-test as the test set.

Emotion Tagging. We use MTG-MoodTheme, another subset of MTG-Jamendo [27] that contains 18.5k audio tracks and 59 tags. Unlike Emomusic, this is a multi-label task, with ROC and AP as metrics.

4. EXPERIMENTAL RESULTS

We use the `fairseq` framework [2] from Meta to train MusicHuBERT and Music2Vec models. All the MusicHuBERT and Music2Vec models are trained for 400k steps with 8 $\times$ NVIDIA A100-40GB GPUs. Training with 8 GPUs takes around $2-3$ days. The experimental results are chiefly as follows.

Our findings suggest these SSL models pre-trained on speech can be helpful for MIR tasks, but pre-trained on music is generally more helpful, besides some exceptions. In section 4.2, we identify the strengths along with weaknesses of training strategies, revealing areas for further improvement. In section 4.3, we discuss the effect of hyperparameters in pretext tasks.

4.1 Pre-trained on Speech and Music

Table 1 demonstrates the performance of HuBERT [3] and data2vec [4] SSL models that were pre-trained on speech recordings and music recordings separately. Here, we only consider the SOTA performance trained with the same dataset train/valid/test split. All of the models are used as parameter-frozen feature extractors. The weighted sum of one output of the CNN tokeniser as well as the 12 outputs of all the transformer layers, are combined with an MLP as the back end. The MLP has only one single 512-dimension hidden layer. The learning rate of the probing is set to 1e-3.

For the HuBERT model, the results pre-trained on speech recordings are comparable with SOTA on tasks like music tagging, beat tracking, pitch estimation and singing technique classification etc., and are surpassed by the results pre-trained on music audio on most of the downstream tasks besides pitch estimation on Nsynth and key detection on GS. For pitch detection, the data samples in Nsynth are a single note played by one single monophonic instrument, which is similar to speech data. So it is reasonable that HuBERT pre-trained on speech data is capable of modelling a single pitch. Although HuBERT surpasses the vanilla MusicHuBERT on GS and Nsynth-pitch, it is surpassed by the results of MusicHuBERT with an ablation study on pre-training hyperparameters (shown in Table 2).

For data2vec, the data2vec-audio results are also comparable with SOTA on many tasks and have a large gap on others, and overall surpassed by Music2Vec or its ablation study shown in Table 3 on most of the tasks as well. But the data2vec results of beat tracking on GTZAN-Rhythm and singer identification on Vocalset surpassed all Music2Vec. Vocalset includes singing of different phonemes with different singing techniques by different singers. The speech SSL system is capable of modelling diverse phonemes in ASR and various timbres of speakers but has less focus on timbre in speaking techniques you may find in opera. On the contrary, the music SSL models may focus more on phonemes (lyrics) and singing timbre (techniques) but include less focus on the singer itself. For beat tracking, we observe that the performance is reduced significantly when

[2] https://github.com/facebookresearch/fairseq
[3] https://huggingface.co/facebook/HuBERT-base-ls960
[4] https://huggingface.co/facebook/data2vec-audio-base

Table 1: Experimental performance of the SSL baseline systems on all downstream tasks

Downstream dataset	MTT		GS key	GTZAN Genre	EMO		Nsynth Instr	Nsynth pitch	VocalSet tech	VocalSet singer	GTzAN Rhythm	MTG Instrument		MTG MoodTheme		MTG Genre		MTG Top50	
Metrics	ROC	AP	Refined Acc	Acc	Emo_V	Emo_A	Acc	Acc	Acc	Acc	F1 (beat)	ROC	AP	ROC	AP	ROC	AP	ROC	AP
HuBERT base	89.8	36.4	15.0	64.8	31.0	57.5	68.2	79.4	61.0	58.8	83.5	73.2	17.0	74.0	11.6	85.0	16.3	81.8	26.5
MusicHuBERT base	90.2	37.7	14.7	70.0	42.1	66.5	69.3	77.4	65.9	75.3	**88.6**	75.5	17.8	76.0	13.9	86.5	18.0	82.4	28.1
data2vec audio base	88.4	33.6	15.5	60.7	23.0	49.6	69.3	77.7	64.9	74.6	36.4	73.1	16.9	73.3	11.0	83.5	14.5	80.6	24.8
Music2vec vanilla	89.1	35.1	19.0	59.7	38.5	61.9	69.4	88.9	**68.3**	69.5	33.5	73.1	16.3	74.3	12.2	85.2	16.5	81.4	26.2
SOTA	**92.0** [40]	**41.4** [6]	**74.3** [28]	**82.1** [41]	**61.7**	**72.1** [6]	**78.2** [20]	**89.2** [23]	65.6 [36]	**80.3** [42]	80.6 [43]	**78.8**	**20.2** [44]	**78.6**	**16.1** [23]	**87.7**	**20.3** [44]	**84.3**	**32.1** [23]

the number of transformer layers increases from 0 to 12. This shows that the data2vec structure may not be useful for learning temporal information.

4.2 Pre-trained with Different Paradigms

From Table 1, we can tell that MusicHuBERT is more promising than Music2vec given that it provides better results in most of the downstream tasks, especially genre classification on GTZAN, emotion regression on EMO and beat tracking on GTZAN. But it is worse on single-pitch estimation on Nsynth, along with key detection on GS.

These phenomena suggest pre-training with the HuBERT paradigm is strongly correlated with the MFCC feature information used for k-means. Therefore, the quantisation results lack multi-pitch information, including harmony or chord modelling, that is essential to key detection. The following research can use the chroma feature to replace MFCCs [5] . On the contrary, the mask prediction for the deep feature in the data2vec pre-training paradigm is clearly better but still has much room for improvement compared to the SOTA. Although the deep feature still lacks sufficient harmonic information for key detection, it already contains enough information for single-pitch estimation, and the MFCCs may focus more on the timbre of instruments instead of the fundamental frequency. Apparently, Music2Vec can learn pitch information more freely. Besides, data2vec is generally a bit worse for tagging than Music2vec, and both are significantly worse on beat tracking compared to HuBERT and MusicHuBERT.

4.3 Ablation Studies on Pretraining Hyperparameters

Here, we carry out an ablation study of hyperparameter search under both pre-training paradigms. Given the time limitation, we did not extract features on MTG datasets and only calculated the results in another 9 downstream tasks.

4.3.1 Ablation Study on MusicHuBERT

We use the number of clusters k =500 and k=2000. For the case k=500, we increase the dimension of MFCC features from 13, which is commonly used in the speech community, to 20, which is widely used in sound event detection. Thus, the dimension of MFCCs combined with their delta features and delta-delta features have 39 and 60 dimensions respectively. For the case of k=2000, we use the 768-dimension deep feature learned from the first iteration experiment to carry out the second iteration k-means.

From Table 2, we can see that MusicHuBERT with k=2000 is better than the k=500 case for most of the tasks. Given HuBERT is good for speech when k=100 or k=500, which is roughly the number of human phonemes, this implies music tokens or notes are much richer than speech and therefore need a larger number for quantisation.

The results on k-means for deep features are better than the vanilla MusicHuBERT besides genre classification on GTZAN, singer identification on vocalset, and singing techniques classification on vocalset. This implies the MFCCs features are good for modelling the human voice, regardless of speech or singing. The results of GTZAN may be due to the randomness as the dataset is very small.

Besides, increasing the dimension of MFCCs provides no significant difference among most of the tasks other than tasks on Nsynth and GS. Increased dimensionality for MFCC features can provide more detailed information on impulse response for a sound event. Thus, monophonic instrumental notes can be better modelled with 60-dimension MFCC features. Furthermore, the emotion regression also provides different results, but the average of the two metrics is nearly the same, providing no significant improvement.

4.3.2 Ablation Study on Music2Vec

We use audio files with 30s length, mask span length 10, mask probability 65%, target top-8 transformer layer the teacher model as a deep feature, and training step 400K as the vanilla setting. We conduct parameter searching and correlation analysis for Music2Vec pretraining, including masking strategy, training steps, the learning target layers, and recording length; the results are shown in Table 3.

We revise the masking strategy by changing the **mask span length** and **mask token probability** in the data2vec-audio-base setting. Mask token probability is the probability for each token to be chosen as the start of the span to be masked, the length of which can also be adapted for different data modalities. The results in Table 3 show that the other span value and other mask token probability provide a bit worse results on nearly all the tasks. This suggests that the data2vec hyperparameters for speech pre-training are generally helpful for music pre-training.

Given the fact that early transformer layer representations generally perform well on key detection and beat

[5] For more information on this, please refer to our following paper MERT: Acoustic Music Understanding Model with Large-Scale Self-supervised Training at `https://arxiv.org/abs/2306.00107`

Table 2: Ablation study on MusicHuBERT hyperparameters (k is the number of MFCC clusters)

Downstream dataset	MTT		GS key	GTZAN Genre	EMO		Nsynth Instr	Nsynth pitch	VocalSet tech	VocalSet singer	GTZAN Rhythm	Average Score
Metrics	ROC	AP	Refined Acc	Acc	Emo_V	Emo_A	Acc	Acc	Acc	Acc	F1 (beat)	score
HuBERT	89.8	36.4	15.0	64.8	31.0	57.5	68.2	79.4	61.0	58.8	83.5	59.8
k=2000 MFCC dim=39	90.2	37.7	14.7	**70.0**	42.1	66.5	69.3	77.4	65.9	75.3	88.6	64.4
k=2000 iter2	**90.4**	37.5	13.8	68.3	**43.3**	67.4	70.0	**80.3**	63.6	70.4	**88.8**	63.8
k=500 MFCC dim=39	89.6	36.1	15.7	64.5	41.0	67.7	66.7	76.8	60.5	72.3	87.5	62.4
k=500 MFCC dim=60	90.3	**38.0**	**17.6**	69.7	40.8	**67.5**	**70.3**	79.0	**66.2**	**75.5**	88.6	**65.0**

Table 3: Ablation study on Music2Vec hyperparameters (span is mask span, prob is mask probability, step is training steps, target=12 uses all 12 transformer layers, and crop5s uses 5s music excerpts)

Downstream dataset	MTT		GS key	GTZAN Genre	EMO		Nsynth Instr	Nsynth pitch	VocalSet tech	VocalSet singer	GTZAN Rhythm	Average Score
Metrics	ROC	AP	Refined Acc	Acc	Emo_V	Emo_A	Acc	Acc	Acc	Acc	F1 (beat)	score
data2vec	88.4	33.6	15.5	60.7	23.0	49.6	69.3	77.7	64.9	**74.6**	**36.4**	55.2
vanilla	89.1	35.1	19.0	59.7	38.5	61.9	69.4	88.9	68.3	69.5	33.5	57.8
span=5	87.3	32.0	15.7	47.6	22.7	41.2	64.2	84.8	56.7	53.8	33.2	49.7
span=15	88.7	34.3	16.4	56.6	39.0	58.8	67.1	88.1	63.1	61.9	33.1	55.2
prob=50	88.5	34.0	23.7	59.3	40.6	55.0	66.8	87.7	64.9	61.7	33.9	56.3
prob=80	88.2	33.9	18.4	50.3	36.7	55.7	67.9	88.9	64.2	65.2	33.7	55.1
step=800k	87.7	32.7	20.3	54.5	34.9	47.3	66.9	87.5	65.6	65.1	33.4	55.0
target=12	89.7	35.2	**26.5**	64.5	41.7	64.2	**71.1**	**89.2**	71.0	73.2	34.1	60.6
crop5s	**90.0**	**36.6**	18.5	**76.6**	**53.4**	**71.6**	68.3	88.9	**71.3**	72.4	33.9	**61.8**

tracking, we modify the **prediction target** provided by the teacher model. We change the prediction target in Music2Vec from the original one, that is, the average of the top-8 layer representations, to all the 12 layers. The results in Table 3 show that Music2Vec actually benefits, not only from the potentially preserved key information shown by a significant increase on GS but all the other tasks as well.

Furthermore, we use **audio length cropping** to shorten music excerpts since longer sequences are more difficult to model. Note that the combined audio length in a batch on a single GPU is not altered, and the hardware environment remains the same, making a single training batch contain a larger number of music samples when clips are cropped. Due to the position embedding in the SSL systems, the model can get information more than 5 seconds after pre-training on only 5-second music recordings. But the key detection provides worse results which may lead to the fact that a local key within a 5-second song may not be identical to the global key in the whole music sentence.

5. CONCLUSION & DISCUSSION

In this paper, we explore the music variants of two distinctive speech-related transformer-based SSL models, data2vec and HuBERT. Our findings suggest that pre-training with music recordings rather than speech can generally improve performance on a wide range of MIR tasks, even when the models and training are designed for speech. There are exceptions for data2vec, however, such as singer identification, the dataset of which is similar to the speech dataset used to pre-train. Thus, when resources are limited, our suggestion is to use speech pre-training models, given that they can provide helpful information about music already. Speech data can be beneficial if lacking a sufficient vocal dataset with different singers, but one should use mainly music data if possible.

Furthermore, we can use the same speech training hyperparameters for masked span and masked probability in music pre-training. But some other hyperparameters, such as the number for pseudo label clustering, might be the shortage of pretext strategies. We identified some limitations of existing speech SSL systems, especially in the case of harmonic information and diversity of music notes. One suggestion is to emphasise key or harmonic in the pretext task for music SSL models by using more than just MFCC features. Also, the number of categories for quantisation in k-means should be much larger if necessary, given the number of pitch, chord, and timbre categories is much larger than the number of human speech phones. This diversity in music might be a bottleneck for both speech SSL systems to learn good music features. For one thing, the larger number of clusters for k-means in HuBERT is expensive to calculate, making it harder to scale up, preventing transformer-based models from reaching their potential for better performance and longer sequence modelling. In addition, it may not be easy for data2vec to jointly learn deeper features. We may need curriculum learning skills or manually-designed features to increase training stability.

Another general suggestion for pre-training recognises that batch size should be as diverse as possible. Given that the memory of one single machine is limited, it is a good idea to shorten the length of audio to be modelled at first, allowing for an increase in batch size, and then train another language model for long sequence modelling.

We believe the findings in this paper to be of value in understanding the potential for SSL speech models applied to music, and we have offered some general insights about music modelling that resulted from this study.

6. ACKNOWLEDGEMENTS

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]. Yizhi Li is fully funded by an industrial PhD studentship (Grant number: 171362) from the University of Sheffield, UK. This work is supported by the National Key R&D Program of China (2020AAA0105200). We acknowledge IT Services at The University of Sheffield for the provision of services for High-Performance Computing. We would also like to express great appreciation for the suggestions from faculties Dr Chris Donahue, and Dr Roger Dannenberg, as well as the facility support from Mr. Yulong Zhang in the preliminary stage.

7. REFERENCES

[1] 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.

[2] S. Gururangan, A. Marasović, S. Swayamdipta, K. Lo, I. Beltagy, D. Downey, and N. A. Smith, "Don't stop pretraining: Adapt language models to domains and tasks," *arXiv preprint arXiv:2004.10964*, 2020.

[3] J. Sarzynska-Wawer, A. Wawer, A. Pawlak, J. Szymanowska, I. Stefaniak, M. Jarkiewicz, and L. Okruszek, "Detecting formal thought disorder by deep contextualized word representations," *Psychiatry Research*, vol. 304, p. 114135, 2021.

[4] A. Newell and J. Deng, "How useful is self-supervised pretraining for visual tasks?" in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 7345–7354.

[5] 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, pp. 12 449–12 460, 2020.

[6] R. Castellon, C. Donahue, and P. Liang, "Codified audio language modeling learns useful representations for music information retrieval," *arXiv preprint arXiv:2107.05677*, 2021.

[7] 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, 2020.

[8] J. Turian, J. Shier, H. R. Khan, B. Raj, B. W. Schuller, C. J. Steinmetz, C. Malloy, G. Tzanetakis, G. Velarde, K. McNally *et al.*, "Hear: Holistic evaluation of audio representations," in *NeurIPS 2021 Competitions and Demonstrations Track*. PMLR, 2022, pp. 125–145.

[9] Y. Zhao and J. Guo, "Musicoder: A universal music-acoustic encoder based on transformer," in *International Conference on Multimedia Modeling*. Springer, 2021, pp. 417–429.

[10] H.-H. Wu, C.-C. Kao, Q. Tang, M. Sun, B. McFee, J. P. Bello, and C. Wang, "Multi-task self-supervised pre-training for music classification," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 556–560.

[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 *ISMIR late braking demo*, 2022.

[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] Y. Li, R. Yuan, G. Zhang, Y. Ma, C. Lin, X. Chen, A. Ragni, H. Yin, Z. Hu, H. He *et al.*, "Large-scale pretrained model for self-supervised music audio representation learning," in *Digital Music Research Network (DMRN) workshop*, 2022.

[14] 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.

[15] J. Spijkervet and J. A. Burgoyne, "Contrastive learning of musical representations," *arXiv preprint arXiv:2103.09410*, 2021.

[16] D. Yao, Z. Zhao, S. Zhang, J. Zhu, Y. Zhu, R. Zhang, and X. He, "Contrastive learning with positive-negative frame mask for music representation," in *Proceedings of the ACM Web Conference 2022*, 2022, pp. 2906–2915.

[17] A. T. Liu, S.-w. Yang, P.-H. Chi, P.-c. Hsu, and H.-y. Lee, "Mockingjay: Unsupervised speech representation learning with deep bidirectional transformer encoders," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6419–6423.

[18] M. Ravanelli, J. Zhong, S. Pascual, P. Swietojanski, J. Monteiro, J. Trmal, and Y. Bengio, "Multi-task self-supervised learning for robust speech recognition," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6989–6993.

[19] A. Saeed, D. Grangier, and N. Zeghidour, "Contrastive learning of general-purpose audio representations," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 3875–3879.

[20] L. Wang, P. Luc, Y. Wu, A. Recasens, L. Smaira, A. Brock, A. Jaegle, J.-B. Alayrac, S. Dieleman, J. Carreira *et al.*, "Towards learning universal audio representations," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 4593–4597.

[21] A. Baevski, W.-N. 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*. PMLR, 2022, pp. 1298–1312.

[22] A. Ragano, E. Benetos, and A. Hines, "Learning music representations with wav2vec 2.0," *arXiv preprint arXiv:2210.15310*, 2022.

[23] M. C. McCallum, F. Korzeniowski, S. Oramas, F. Gouyon, and A. F. Ehmann, "Supervised and unsupervised learning of audio representations for music understanding," in *ISMIR*, 2022.

[24] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar *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.

[25] 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.

[26] E. Law, K. West, M. I. Mandel, M. Bay, and J. S. Downie, "Evaluation of algorithms using games: The case of music tagging." in *ISMIR*. Citeseer, 2009, pp. 387–392.

[27] D. Bogdanov, M. Won, P. Tovstogan, A. Porter, and X. Serra, "The mtg-jamendo dataset for automatic music tagging." ICML, 2019.

[28] 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.

[29] 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 *Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR); 2015 Oct 26-30; Málaga, Spain.[Málaga]: International Society for Music Information Retrieval, 2015. p. 364-70.* International Society for Music Information Retrieval (ISMIR), 2015.

[30] 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, pp. 367–372.

[31] 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.

[32] 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.

[33] 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.

[34] 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.

[35] J. Wilkins, P. Seetharaman, A. Wahl, and B. Pardo, "Vocalset: A singing voice dataset." in *ISMIR*, 2018, pp. 468–474.

[36] Y. Yamamoto, J. Nam, and H. Terasawa, "Deformable cnn and imbalance-aware feature learning for singing technique classification," *arXiv preprint arXiv:2206.12230*, 2022.

[37] S. Böck, F. Krebs, and G. Widmer, "Joint beat and downbeat tracking with recurrent neural networks." in *ISMIR*. New York City, 2016, pp. 255–261.

[38] 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*, Amsterdam, The Netherlands, 10 2016, pp. 1174–1178.

[39] U. Marchand and G. Peeters, "Swing ratio estimation," in *Digital Audio Effects 2015 (Dafx15)*, 2015.

[40] Q. Huang, A. Jansen, L. Zhang, D. P. Ellis, R. A. Saurous, and J. Anderson, "Large-scale weakly-supervised content embeddings for music recommendation and tagging," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 8364–8368.

[41] 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.

[42] M. Modrzejewski, P. Szachewicz, and P. Rokita, "Transfer learning with deep neural embeddings for music classification tasks," in *Artificial Intelligence and Soft Computing: 21st International Conference, ICAISC 2022, Zakopane, Poland, June 19–23, 2022, Proceedings, Part I*. Springer, 2023, pp. 72–81.

[43] M. Heydari, F. Cwitkowitz, and Z. Duan, "Beatnet: Crnn and particle filtering for online joint beat downbeat and meter tracking," *arXiv preprint arXiv:2108.03576*, 2021.

[44] P. Alonso-Jiménez, X. Serra, and D. Bogdanov, "Music representation learning based on editorial metadata from discogs," in *ISMIR*, 2022.

TRANSFORMER-BASED BEAT TRACKING WITH LOW-RESOLUTION ENCODER AND HIGH-RESOLUTION DECODER

Tian Cheng **Masataka Goto**
National Institute of Advanced Industrial Science and Technology (AIST), Japan
{tian.cheng, m.goto}@aist.go.jp

ABSTRACT

In this paper, we address the beat tracking task which is to predict beat times corresponding to the input audio. Due to the long sequential inputs, it is still challenging to model the global structure efficiently and to deal with the data imbalance between beats and no beats. In order to meet the above challenges, we propose a novel Transformer-based model consisting of a low-resolution encoder and a high-resolution decoder. The encoder with low temporal resolution is suited to capture global features with more balanced data. The decoder with high temporal resolution is designed to predict beat times at a desired resolution. In the decoder, the global structure is considered by the cross attention between the global features and high-dimensional features. There are two key modifications in the proposed model: (1) adding 1D convolutional layers in the encoder and (2) replacing positional embedding by the upsampled encoder features in the decoder. In the experiment, we achieved the state-of-the-art performance and showed that the decoder produced more precise and stable results.

1. INTRODUCTION

Beat tracking is an important task in Music Information Retrieval (MIR) area with a long history. The task is to predict beat times, a periodic sequence of time instants which people can tap along with, from musical pieces. The first attempt of beat tracking for polyphonic musical audio signals can date back to around 30 years ago [1]. In the past three decades, we see the techniques shifting from signal processing to machine learning. In the most recent deep-learning-based methods, sequence models have been used to produce beat probabilities for each input frame, with the final beat times detected by the HMM on the beat probabilities in the post-processing step. In these models, various sequence models have been used, including Recurrent Neural Network (RNN) [2–4], Temporal Convolutional Network (TCN) [5–7], and Transformers [8–10]. Convolutional Neural Networks (CNNs) are also commonly combined in the models for front-end feature embedding [9, 11, 12].

© T. Cheng and M. Goto. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** T. Cheng and M. Goto, "Transformer-based beat tracking with low-resolution encoder and high-resolution decoder", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

To produce good beat tracking results, the model needs to consider both local timing and global consistency. This brings a contradiction on choosing the temporal resolution. The problem of using low temporal resolution (i.e., low frame rate) is that we cannot predict beats with precise times. On the other hand, using high temporal resolution (i.e., high frame rate) results in long sequential inputs and imbalanced output labels. The current commonly-used 10ms temporal resolution enables an easy comparison on the results. With such high temporal resolution, the sequential inputs are already relatively long for the RNN, causing the gradient vanishing problem. Using TCN and Transformer helps to solve the gradient vanishing problem, while modeling long sequences can still be challenging. To model long sequences more efficiently, more compact models have been proposed, such as dilated self-attention [9] and linear Transformer [10]. Another problem caused by the high temporal resolution is the data imbalance issue between beats and no beats. Given the same tempo, the higher the temporal resolution is, the more the no-beat labels exist between the beat labels. In order to solve this problem, smoothed labels [7, 9, 13] and weighted loss functions designed for the data imbalance problem [14–16] are applied to achieve more efficient training. The above long sequence modelling issue and data imbalance issue can be more challenging if a higher temporal resolution than 10ms is needed. In fact, there are some commercial music applications that potentially require more temporally precise beat tracking for sample-wise audio editing/mixing/mashups based on beat timings and highly rigid music synchronization.

In order to tackle the contradiction between high and low temporal resolutions, we propose a novel beat tracking model based on the Transformer with low-resolution encoder and high-resolution decoder. With the low temporal resolution, the sequential inputs become shorter and the training data become more balanced, which makes the global structure easier to model by the encoder. At the same time, the beat time precision in the output can still be preserved by the decoder with the high temporal resolution. The Transformer is a good architecture for joining the two parts because the encoder and decoder are not required to be the same length, and features of different dimensions can be jointly learned by the cross attention in the decoder. We modify the original Transformer in several ways to make it work for beat tracking with the proposed combination of the encoder and decoder. First, we

stack 2D convolutional layers for feature learning from the spectral inputs and 1D convolutional layers inside the encoder layers for feature smoothing and dimension adjustment. Second, we use the upsampled encoder feature to replace the position encoding in the decoder. In the experiments, we produced results comparable to the state-of-the-art performance. The analysis of experimental results showed that the decoder not only produced more precise results, but also helped to recover the missing beats and to filter out unwanted peaks between beats, making the beat tracking more stable.

The rest of paper is organised as follows. Section 2 summarises the related work on Transformer-based beat tracking models and multi-scale models. In Section 3, we give a detailed description of the proposed model, especially focusing on the proposed modifications. Section 4 presents the experiments with ablation study, results, and attention visualisation. In the last section, we conclude the paper and show aspects for future improvements.

2. RELATED WORKS

2.1 Transformer in beat tracking

Recently, Transformers have been used for many MIR tasks with promising performance, such as music transcription [17–19], music tagging [20], and beat tracking [8–10]. In the SpectTNT model, Transformer encoders are used for modeling both the spectral and temporal features [8]. The model also combines the Temporal Convolutional Network (TCN) model for better beat tracking results. Since the Transformer is computationally expensive for long sequences, the inputs are divided in 6-second chunks to process. For modelling the long sequences efficiently, more compact Transformers have been applied, including the dilated Transformer in the Beat Transformer model [9] and the linear Transformers for singing beat tracking [10].

These existing methods are based on Transformer encoders, while in our model, we use both the encoder and decoder, which is an important contribution of this paper. By adding the decoder layers, we can set a more reasonable temporal resolution for the encoder input, more specifically, low-temporal-resolution inputs. In other words, in the proposed model, the temporal resolution of the encoder can be independent of the high temporal resolution of the beat tracking output. With the low-resolution encoder, we are able to model the sequences more efficiently and obtain more balanced training data.

2.2 Multi-scale structure

In the proposed model, we leverage features at different scales: low-dimensional features for modeling the global structure and high-dimensional features for predicting precise beat times. Such multi-scale structure has also been used in related domains. In our previous work, we proposed a multi-scale beat tracking model based on the Wave-U-Net, which learned features at different scales from waveform and spectral inputs with downsampling and upsampling blocks [21]. Schreiber et al. achieved tempo estimation by concatenating multi-scale features learned from a series of convolutional layers with different filter size from 32 to 256 [22]. Sun et al. [23] propose a multi-scale structure for tempo estimation by downsampling/upsampling the feature to different scales and combining multi-scale features repeatedly.

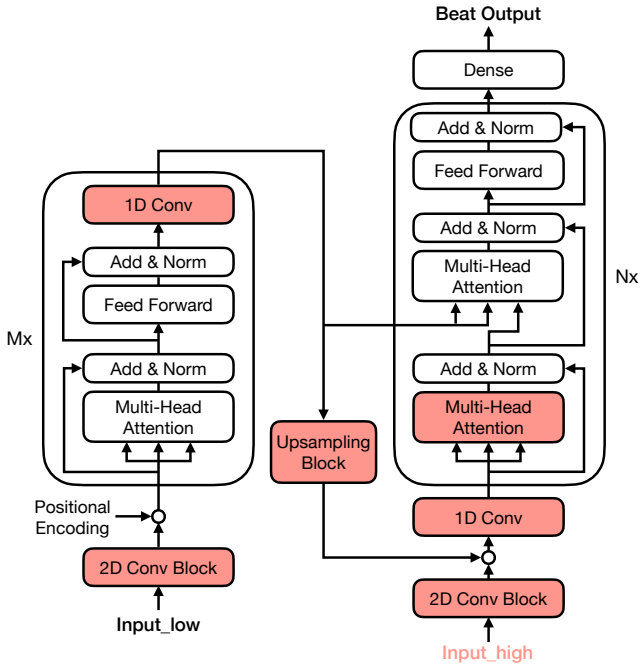


Figure 1: The model architecture of the beat tracking model. The coloured parts indicate the modifications from the original Transformer.

3. MODEL ARCHITECTURE

The proposed model architecture is shown in Figure 1. Our model is based on the Transformer, consisting of both encoder and decoder layers. As we have already written, the key idea is to use a low-resolution encoder for modelling the global structure, depicted on the left side of the figure (starting from "Input_low" denoting the low-resolution input), and a high-resolution decoder for predicting beats more precisely, depicted on the right side of the figure (starting from "Input_high" denoting the high-resolution input).

To make it work for beat tracking, we make some modifications on the original Transformer. The modified parts are coloured in Figure 1, including adding 1D convolutional layers ("1D Conv") in the encoder, replacing the positional embedding by upsampled encoder features (from "Upsampling Block") in the decoder, and stacking 2D convolutional layers for feature learning from the spectral inputs ("2D Conv Block") in both the encoder and decoder.

In the following subsections, we illustrate the proposed model in detail.

Network parameter	Setting
filter size	3×3, 3×3, 3×3, 1×3
maxpooling size	1×3, 1×3, 1×3
activation function	ReLU

Table 1: Parameters used in the convolution block.

Encoder	
Conv Block filter number	48, **64**, 72
encoder layer number	3, 4, **5**, 6
head number	4, 8, 12, **16**
key dim.	8, 16, 24, **32**
inner-layer dim. in feed-forward	32, 48, **64**, 72, 128
Conv 1D filter number	32, **64**, 96
Conv 1D filter size	5, **15**

Decoder	
Conv Block filter number	**32**
decoder layer number	1, **2**
head number	**4**

Table 2: Hyperparameters in the proposed beat tracking model.

3.1 Input features and 2D convolutional layers

We use the Mel-spectrogram as the input features. For the low-resolution encoder, we computed 80-dimensional Mel-spectrogram with a 22050 sample rate and a hop size of 1024, roughly corresponding to a 46 ms temporal resolution (more precisely, 46.44 ms). The high-resolution Mel-spectrogram is computed the same way but in a hop size of 256, roughly corresponding to 12 ms temporal resolution (more precisely, 11.61ms). The dimensions of the inputs are (T, 80) and (4T, 80), respectively, where T is the frame length of the low-resolution input. We choose such resolutions so that the low-resolution can still distinguish beat and no beat frames for fast-tempo pieces, and the high-resolution outputs can be easily compared to those of other methods. In the 2D convolutional block, we stack four 2D convolutional layers and three maxpooling layers for feature embedding, with details show in Table 1.

3.2 Encoder with 1D convolutional layers

The encoder consists of identical encoder layers which process the low-dimensional features. As shown on the left side in Figure 1, each encoder layer includes a multi-head attention sub-layer and a fully connected feed-forward network with residual connections. Before the encoder, we concatenate the features with the positional encoding. Since in the Transformer, the input dimension is not changeable within the encoder layers, we stack a 1D convolutional layer after the feed-forward network for feature smoothing and channel number adjustment.

3.3 Upsampling Block

Another important change of the proposed model is that we replace the original positional encoding by upsampled encoder features for the decoder. In the upsampling block, there are two upsampling layers with linear interpolation. The upsampled features are then concatenated with the high-dimensional features. We also stack a 1D convolutional layer to re-dimension the concatenated features. In the preliminary experiment, we confirmed that the original positional encoding does not work well and the upsampled features worked for indicating rough beat positions. The ablation study for this replacement is presented in Section 4.3.

3.4 Decoder

The decoder processes the high-dimensional features for predicting more precise beats. The decoder layer consists of three components. As shown on the right side in Figure 1, between the multi-head attention sub-layer and a fully connected feed-forward network, there is another multi-head attention sub-layer which computes the cross-attention between the low- and high-dimensional features. We use the decoder as a discriminative model for predicting the output based on the input, rather than a generative model as the original Transformer decoder. Hence we do not need to use the causal mask in the first multi-head attention sub-layer.

3.5 Output Layer and Post-Processing

As shown in Figure 1, we stack a dense layer with the sigmoid activation at the end of the decoder for producing beat outputs. Then, we apply the DBN from Madmom [2] for post-processing. We first take the nearest integer of frame per second (fps) for the post-processing, and then map the results to the original fps.

3.6 Complete Architecture

We apply random search to find the best hyperparameters for the model. We set up a grid of hyperparameter values according to Table 2, and randomly select a subset to compare. The decoder uses the same parameters as the encoder if not present. The finally chosen parameters are shown in bold.

4. EXPERIMENTS

4.1 Data

We train, validate, and test the proposed model by using the standard music datasets with beat annotations as shown in Table 3. For training and validation sets, all musical pieces are segmented into 30-second clips with 50% overlap. Segments from the same musical piece appear only in either the training or validation set to ensure that there is no overlap between the training and validation sets. Test results are obtained on the whole pieces without segmentation.

We do data augmentation for better tempo balance. We follow the strategy in [7] to generate input features for less

Usage	Datasets
training only	Beatles [24], Harmonix [25], 5 RWC datasets [26, 27], tapcorrect [28]
8-fold cross-validation	Ballroom [29, 30], Hainsworth [31], SMC [32]
testing only	GTZAN [33, 34]

Table 3: The usage of the standard datasets for beat tracking evaluation.

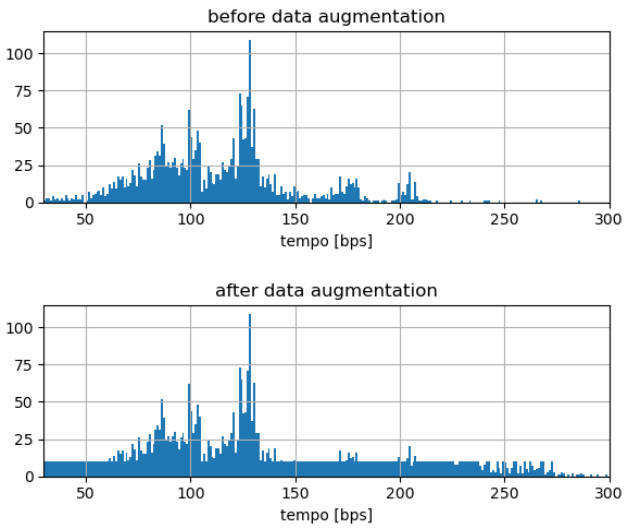


Figure 2: The tempo distribution before and after the data augmentation for the model training.

representative tempos by changing the hop size when computing the mel-spectrogram. The tempo distribution before and after the data augmentation is shown in Figure 2.

Inspired by [9], we also process the input mel-spectral features by Harmonic Percussive Source Separation (HPSS) and obtain the original mel-spectrogram S, the harmonic part H, and the percussive part P. In the preliminary experiment, we compare two way of using HPSS: one way is as data augmentation which triples the training data (i.e., we can use all of S, H, and P with the same beat annotations); the other way is to concatenate three parts, S, H, and P, as the more informative input features. Since the results showed that using HPSS as the data augmentation works better, we decided to take that way.

4.2 Training

In order to train the model effectively, we compare three training methods as shown in Table 4. The first method is training the model (i.e., both the encoder and decoder) from scratch.

For the other two methods, we first temporarily stack a dense layer at the end of the encoder and pre-train the encoder only with the low-resolution labels. Then we initialize the encoder with this pre-trained model and start training the decoder. The second method trains the parameters of the decoder only, by freezing the parameters of this pre-traind encoder. The third method trains the parameters of both the encoder and decoder after the above initialization of the encoder.

Method	Initialization	Encoder parameters
1	None	Trained with decoder
2	Pre-trained encoder	Not trainable in training decoder
3	Pre-trained encoder	Trainable in training decoder

Table 4: Three training methods (the third method was the best).

We choose the third method for training the model because it worked best in our preliminary experiments. The model is trained with binary cross-entropy by using the RMSprop optimiser [35] with a learning rate of 0.0002. The batch size is set to 16 for pre-training the encoder, and 4 for training the whole model.

4.3 Ablation Study

To illustrate the influence of replacing the positional encoding by the upsampled encoder features in the decoder, we show the differences on the training with and without the proposed modification (replacement) in Figure 3. We see that using the upsampled encoder features decreased the validation loss slightly and decreased the training loss in a large degree in comparison to using the positional encoding. This shows that the modified model can learn better from the training data and generalise well in the validation set, resulting in better beat tracking results.

4.4 Evaluation

We evaluate the proposed method with three standard metrics: F-measure with a tolerance window of 70ms, continuity-based metrics CMLt (tracking accuracy on the correct metrical level), and AMLt (tracking accuracy with alternate metrical levels allowed) [24].

4.4.1 The proposed method

In order to validate our model design, besides results on the decoder outputs, we also show results on the pre-trained encoder outputs. In Table 5, “Encoder (Th)” indicates the results obtained by applying a threshold of 0.1 without using the DBN. “Encoder” and “Decoder (Proposed)” results are processed by the DBN in the post-processing step, with the encoder outputs linear interpolated. If we compare the results of “Encoder (Th)” and “Encoder”, we observe that the DBN post-processing step increased the performance in all the four datasets, especially for the continuity-based results (CMLt and AMLt). Furthermore, if we compare them with the “Decoder (Proposed)” results, which corresponds to the proposed model, we see performance further increased on the Ballroom, SMC, and GTZAN datasets.

Figure 3: The training and validation losses for training the decoder with the original positional embedding or with upsampled encoder features.

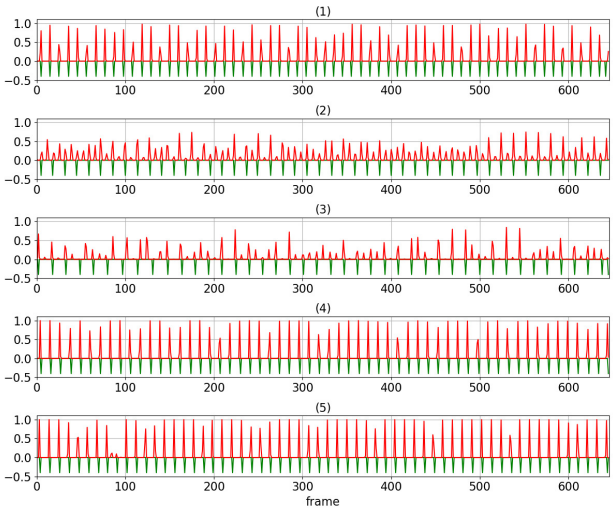
Method	F-measure	CMLt	AMLt
Dataset: Ballroom			
Encoder (Th)	90.7	80.1	85.7
Encoder	93	87.4	96.1
Decoder (Proposed)	95	91.1	96.4
Beat trans [9]	96.8	95.4	96.6
TF trans [8]	96.2	93.9	96.7
TCN [7]	96.2	94.7	96.1
Dataset: Hainsworth			
Encoder (Th)	84.4	66.7	81.8
Encoder	88.2	81	93.4
Decoder (Proposed)	87	76.2	93.6
Beat trans [9]	90.2	84.2	91.8
TF trans [8]	87.7	86.2	91.5
TCN [7]	90.4	85.1	93.7
Dataset: SMC			
Encoder (Th)	53.9	32.9	45.6
Encoder	55	45.8	64.1
Decoder (Proposed)	55.4	45.1	65.6
Beat trans [9]	59.6	45.6	63.5
TF trans [8]	60.5	51.4	66.3
TCN [7]	55.2	46.5	64.3
Dataset: GTZAN			
Encoder (Th)	87.1	72.8	85.5
Encoder	87.8	78.5	93.7
Decoder (Proposed)	88.4	80.8	94
Beat trans [9]	88.5	80	92.2
TF trans [8]	88.7	81.2	92
TCN [7]	88.5	81.3	93.1

Table 5: Testing results for comparing the proposed method with three state-of-the-art beat tracking models [7–9]. The GTZAN dataset is held out for testing only; other datasets are used in the 8-fold cross-validation. (Th) means results obtained with a threshold of 0.1 without using the DBN post-processing step.

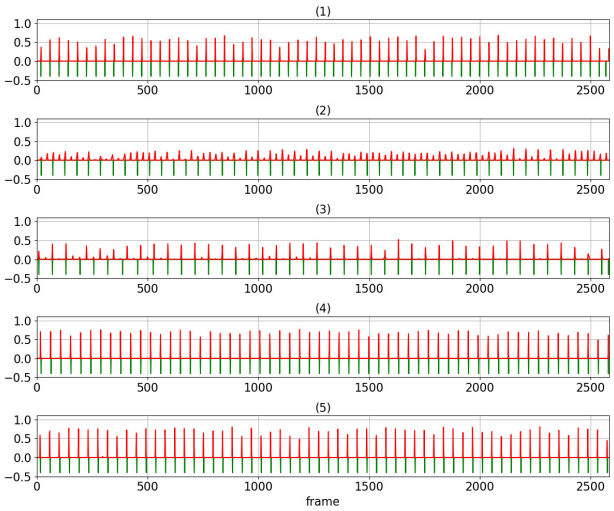
In order to understand the effect of the proposed decoder better, we show the outputs examples from the pre-trained encoder and the decoder in Figure 4. We can see that the encoder outputs are large at beat times, which is benefit from the more balanced training data in shorter sequences. On the other hand, the decoder outputs are small and even (i.e., more stable), as we expected. As we design, the decoder basically predicts the beat times at a higher resolution in comparison to the encoder. The decoder also helped to recover missing beats as shown in the 5th example, where some beats are missing in the encoder output but they are recovered in the decoder output. Moreover, the decoder helped to filter out peaks between beats as shown in the 3rd example, where peaks are more regularly placed in the decoder output. With the above effects, using the proposed decoder generally improved results except for the Hainsworth dataset. For the Hainsworth dataset, we see the CMLt decreased, but the AMLt remained the same level, which means that the decrease on the performance is caused by phase and octave errors. Since the peaks from the decoder are evener, in some cases it would be more difficult for the DBN to exclude the peaks between beats.

4.4.2 Comparison to state-of-the-art beat tracking models

As shown in Table 5, results of the proposed model ("Decoder (Proposed)") were comparable to the state-of-the-art results obtained by three beat tracking models [7–9], despite not the best. Since our goal is not to achieve better performances than all the state-of-the-art models, these results are satisfactory since we can show the high performances of the proposed model with different temporal resolutions. We see noticeable gap on the CMLt in comparison to other methods, which means the proposed model encountered more phase and octave errors. We hope this could be improved by including related topics in the multi-task learning as in [7, 9]. In addition, for the testing-only dataset GTZAN, the F-measures achieved by thresholding the encoder outputs ("Encoder (Th)") are better than what we expected, given the fact that it did not use the DBN post-processing step.



(a) Encoder outputs



(b) Decoder outputs

Figure 4: Output examples from the pre-trained encoder and the decoder for five different pieces. The ground-truth beat annotations are indicated by lines pointing down.

4.5 Attention Visualisation

In order to understand how low- and high-dimensional features are jointly learned, we show the cross attention matrix between low- and high-dimensional features in the second decoder layer in Figure 5. We found that for high-dimension features at beat times (frames), it got attention at each beat on the low-dimensional features. For no-beat times (frames), all attentions were drawn to the frames after the corresponding beat times, which formed horizontal lines in this figure. With such attentions, the final high-dimensional beat outputs were predicted with the captured global beat structure considered.

5. CONCLUSIONS AND FUTURE WORK

We present a novel Transformer-based model for beat tracking. The proposed model consists of both encoder layers and decoder layers which work on low- and high-dimensional features, respectively. We obtained beat tracking performances which are comparable to the state-of-the-art beat tracking results. The experimental results showed that the proposed model worked well as designed: with the low-dimensional (low-temporal-resolution) encoder for capturing the global beat structure and high-dimensional (high-temporal-resolution) decoder for predicting more precise beats. Thus, the proposed Transformer-based encoder and decoder structure succeeds in providing a new framework for handling multi-scale features for beat tracking. Beyond beat tracking, the advantage of this framework can be summarized as follows.

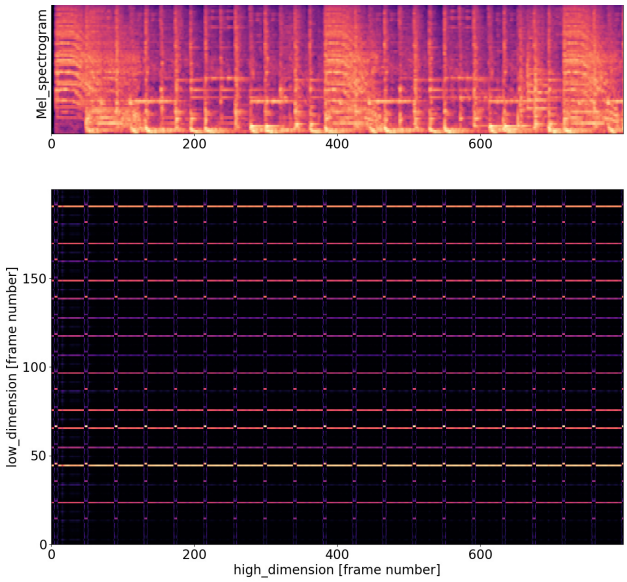


Figure 5: Cross attention matrix between low dimensional features and high dimensional features in the decoder layer.

- The encoder and decoder do not require inputs to be the same length (same temporal resolution), they can be used to handling features at different scales, which enables us to sample the features with more reasonable time resolutions.

- We can make use of features at different scales jointly learned by the cross attention in the decoder.

We therefore believe that this framework is also adaptable for other MIR tasks, such as musical structure boundary detection.

As the analysis of our experimental results showed, phase and octave errors are relatively high in our results. As future work, we would like to tackle the problems by combining downbeat tracking and tempo estimation in the proposed model by using multi-task learning. In addition, we also plan to use our model to produce beat outputs in a higher temporal resolution, which is demanded by some practical music applications as we discussed in Section 1. Yet another advantage of our model is that such precise beats can be achieved by using transfer learning with a higher-temporal-resolution input for the decoder.

6. ACKNOWLEDGMENTS

This work was supported in part by JST CREST Grant Number JPMJCR20D4 and JSPS KAKENHI Grant Number JP21H04917, Japan.

7. REFERENCES

[1] M. Goto and Y. Muraoka, "A Beat Tracking System for Acoustic Signals of Music," in *Proc. ACM Multimedia*, 1994, pp. 365–372.

[2] S. Böck, F. Krebs, and G. Widmer, "A Multi-Model Approach to Beat Tracking Considering Heterogeneous Music Styles," in *Proc. ISMIR*, 2014.

[3] ——, "Joint Beat and Downbeat Tracking with Recurrent Neural Networks," in *Proc. ISMIR*, 2016.

[4] F. Krebs, S. Böck, and G. Widmer, "Downbeat Tracking Using Beat-Synchronous Features and Recurrent Networks," in *Proc. ISMIR*, 2016.

[5] M. E. P. Davies and S. Böck, "Temporal Convolutional Networks for Musical Audio Beat Tracking," in *Proc. EUSIPCO*, 2019.

[6] S. Böck, M. E. P. Davies, and P. Knees, "Multi-Task Learning of Tempo and Beat: Learning One to Improve the Other," in *Proc. ISMIR*, 2019, pp. 486–493.

[7] S. Böck and M. E. Davies, "Deconstruct, Analyse, Reconstruct: How to Improve Tempo, Beat, and Downbeat Estimation," in *Proc. ISMIR*, 2020.

[8] Y. Hung, J. Wang, X. Song, W. T. Lu, and M. Won, "Modeling beats and downbeats with a time-frequency transformer," in *Proc. ICASSP*, 2022, pp. 401–405.

[9] J. Zhao, G. Xia, and Y. Wang, "Beat Transformer: Demixed Beat and Downbeat Tracking with Dilated Self-Attention," in *Proc. ISMIR*, 2022, pp. 169–177.

[10] M. Heydari and Z. Duan, "Singing Beat Tracking With Self-supervised Front-end and Linear Transformers," in *Proc. ISMIR*, 2022, pp. 617–624.

[11] M. Fuentes, B. Mcfee, H. C. Crayencour, S. Essid, and J. P. Bello, "Analysis of Common Design Choices in Deep Learning Systems for Downbeat Tracking," in *Proc. ISMIR*, 2018.

[12] T. Cheng, S. Fukayama, and M. Goto, "Joint Beat and Downbeat Tracking Based on CRNN Models and a Comparison of Using Different Context Ranges in Convolutional Layers," in *Proc. ICMC*, 2020.

[13] ——, "Convolving Gaussian Kernels for RNN-based Beat Tracking," in *Proc. EUSIPCO*, 2018, pp. 1919–1923.

[14] F. Pedersoli and M. Goto, "Dance Beat Tracking from Visual Information Alone," in *Proc. ISMIR*, 2020, pp. 400–408.

[15] C. J. Steinmetz and J. D. Reiss, "WaveBeat: End-to-end beat and downbeat tracking in the time domain," in *151st AES Convention*, 2021.

[16] T.-P. Chen and L. Su, "Toward postprocessing-free neural networks for joint beat and downbeat estimation," in *Proc. ISMIR*, 2022, pp. 27–35.

[17] C. Hawthorne, I. Simon, R. Swavely, E. Manilow, and J. Engel, "Sequence-to-sequence piano transcription with transformers," in *Proc. ISMIR*, 2021, pp. 246–253.

[18] L. Oua, Z. Guo, E. Benetos, J. Han, and Y. Wang, "Exploring transformer's potential on automatic piano transcription," in *Proc. ICASSP*, 2022, pp. 776–780.

[19] J. Gardner, I. Simon, E. Manilow, C. Hawthorne, and J. H. Engel, "Mt3: multi-task multitrack music transcription," in *Proc. of the Tenth International Conference on Learning Representations (ICLR)*, 2022.

[20] M. Won, K. Choi, and X. Serra, "Semi-supervised music tagging transformer," in *Proc. ISMIR*, 2021, pp. 769–776.

[21] T. Cheng and M. Goto, "U-Beat: A multi-scale beat tracking model based on Wave-U-Net," in *Proc. ICASSP*, 2023.

[22] H. Schreiber and M. Meinard, "A single-step approach to musical tempo estimation using a convolutional neural network," in *Proc. ISMIR*, 2018, pp. 98–105.

[23] X. Sun, Q. He, Y. Gao, and W. Li, "Musical Tempo Estimation Using a Multi-scale Network," in *Proc. ISMIR*, 2021.

[24] M. E. P. Davies, N. Degara, and M. D. Plumbley, "Evaluation Methods for Musical Audio Beat Tracking Algorithms," Queen Mary University of London, London, United Kingdom, Tech. Rep. C4DM-TR-09-06, 2009.

[25] 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 *Proc. ISMIR*, 2019.

[26] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "RWC Music Database: Popular, Classical, and Jazz Music Databases," in *Proc. ISMIR*, 2002, pp. 287–288.

[27] ——, "RWC Music Database: Music Genre Database and Musical Instrument Sound Database," in *Proc. ISMIR*, 2003, pp. 229–230.

[28] J. Driedger, H. Schreiber, W. B. de Haas, and M. Müller, "Towards automatically correcting tapped beat annotations for music recordings," in *Proc. ISMIR*, 2019.

[29] F. Gouyon, A. Klapuri, S. Dixon, M. Alonso, G. Tzanetakis, C. Uhle, and P. Cano, "An Experimental Comparison of Audio Tempo Induction Algorithms," *IEEE Trans. Audio, Speech, Language Process.*, vol. 14, no. 5, pp. 1832–1844, 2006.

[30] F. Krebs, S. Böck, and G. Widmer, "Rhythmic Pattern Modeling for Beat and Downbeat Tracking in Musical Audio," in *Proc. ISMIR*, 2013, pp. 227–232.

[31] S. W. Hainsworth and M. D. Macleod, "Particle Filtering Applied to Musical Tempo Tracking," *EURASIP Journal on Applied Signal Process.*, vol. 15, 2004.

[32] A. Holzapfel, M. E. P. Davies, J. R. Zapata, J. a. L. Oliveira, and F. Gouyon, "Selective Sampling for Beat Tracking Evaluation," *IEEE Trans. Audio, Speech, Language Process.*, vol. 20, no. 9, pp. 2539–2548, 2012.

[33] G. Tzanetakis and P. Cook, "Musical Genre Classification of Audio Signals," *IEEE Speech Audio Process.*, vol. 10, no. 5, 2002.

[34] U. Marchand and G. Peeters, "Swing Ratio Estimation," in *Proc. DAFx*, 2015.

[35] T. Tieleman and G. Hinton, "Lecture 6.5—RMSProp: Divide the Gradient by a Running Average of its Recent Magnitude," COURSERA: Neural Networks for Machine Learning, 2012.

ADDING DESCRIPTORS TO MELODIES IMPROVES PATTERN MATCHING: A STUDY ON SLOVENIAN FOLK SONGS

Vanessa Nina Borsan **Mathieu Giraud**
Univ. Lille, CNRS, Centrale Lille
UMR 9189 CRIStAL, F-59000 Lille, France
{vanessa,mathieu}@algomus.fr

Richard Groult **Thierry Lecroq**
Univ Rouen Normandie, INSA Rouen Normandie,
Université Le Havre Normandie, Normandie Univ
LITIS UR 4108, F-76000 Rouen, France
{thierry.lecroq,richard.groult}@univ-rouen.fr

ABSTRACT

The objective of pattern-matching topics is to gain insights into repetitive patterns within or across various music genres and cultures. This approach aims to shed light on the recurring instances present in diverse musical traditions. The paper presents a study analyzing folk songs using symbolic music representation, including melodic sequences and musical information. By examining a corpus of 400 monophonic Slovenian tunes, we are releasing annotations of structure, contour, and implied harmony. We propose an efficient algorithm based on suffix arrays and bit-vectors to match both music content (melodic sequence) and context (descriptors). Our study reveals that certain descriptors, such as contour types and harmonic "stability" exhibit variations based on phrase position within a tune. Additionally, combining melody and descriptors in pattern-matching queries enhances precision for classification tasks. We emphasize the importance of the interplay between melodic sequences and music descriptors, highlighting that different pattern queries may have varying levels of detail requirements. As a result, our approach promotes flexibility in computational music analysis. Lastly, our objective is to foster the knowledge of Slovenian folk songs.

1. INTRODUCTION

Music pattern analysis in the field of Music Information Retrieval (MIR) is extensively studied. The challenges of this topic extend beyond algorithms, encompassing diverse music forms, representation (signal, symbolic, or textual), music content, and cultural metadata.

© VN. Borsan, M. Giraud, R. Groult and T. Lecroq. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** VN. Borsan, M. Giraud, R. Groult and T. Lecroq, "Adding Descriptors to Melodies Improves Pattern Matching: a Study on Slovenian Folk Songs", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

1.1 Content and Context in Ethnomusicology

Ethnomusicologists analyze recordings, live performances, and transcriptions (in various notations) to understand the composition of music. While transcriptions reveal the *what* of the music, cultural context is essential for comprehending the *how* and *why* behind these musical structures.

Initiated by Merriam [1], and many others [2–7], the music is to be observed *in* culture (in his later work, *as* culture), or as a multi-dimensional object, a direct consequence of the organization of social structures, and vice versa. Some studies [4, 6, 8–10] have primarily compared folk tunes based on their music content. In others, including the Slovenian Folk Song Collection [11], the categorization of the collection is organized according to the elements, such as lyrics and other textual content.

Considering *music material*, it can be explored as a general outline for music analysis [12], or through specific music descriptors, such as melodic contour (the melodic arch shape) [13]. Recent studies have expanded the use of descriptors to analyze folk songs, incorporating a broader range of attributes. De la Ossa [14] suggests basic music descriptors be included, such as scale types, range, several levels of rhythmic information, and so on. *Cantometrics*, introduced by Lomax [15, 16], proposed 37 descriptors, (almost) independent from usual Western music theory. His idea of representing datasets as a digital "Global Jukebox" was recently completed. [17–19]. Computational methodologies encourage us to process more data, including multiple layers of music *content*, and *context* as descriptors (music and/or metadata). Serra [20] exposed the presence of musical entities (performer, music, instrument, etc.) that "are linked by various types of relationships," which contribute to the understanding of music as a whole. Conklin and Neubarth also stressed the importance of non-musical information, such as region and genre [21], extended Densmore's and observed (super)area and (super)type information [22]. These non-musical phenomenons, although limited, were always correlated with different types of music content (rhythm, melody, pattern, and antipattern [23]) of folk songs. Although focusing on musical content, especially on cadences, van Kranen-

burg also considers lyric, perceptual information, as well as other information [24–26].

1.2 Pattern Inference and Matching

Music pattern searching or matching is most commonly approached from a music analysis perspective [12, 27–30], by addressing music structures, such as melody, harmony, and rhythm. Previous contributions focused on a single or a couple of features or the computational representations and matching of multi-feature music patterns. The Mongeau-Sankoff algorithm [31] simultaneously explores multidimensional music features, as it defines the distance between any two melodies depending on the pitch, tonal contour and rhythmic structure. The pattern similarity is ranked by the number of transformations, including consolidation and fragmentation.

Other dynamic programming methods for melodic sequence alignment were proposed [6, 32, 33], as well as other methods on the general melodic or pitch-related queries [33–36]. Some research added the rhythmic [37], or, especially with multipart music, the harmonic information [38]. In another solution, Marsden adapts the *hierarchical or tree structures* for representing and comparing melodies [39]. Lartillot, conversely, matched melodic sequence (or motives) by using *heterogeneous patterns* [30], whose occurrences can be located through multiple parametric dimensions – including contextual ones, such as implied harmony.

1.3 Motivation and Contents

These studies indicate that a melodic pattern is not isolated from other musical elements, such as a phrase, rhythmic or harmonic structure, ornamentation, and so on. While most distinguish between *music material* and *cultural metadata*, we instead split the first into *melodic sequence* and *descriptors* (see Table 1). Our objective is that *melodic phrase should never be detached from its context.* Hence, we focus on segmented melodic phrases that never lose their identifier (connecting them to all supplementary (meta)data), nor their position within a tune (first, middle, last). This enables the *tune description by phrase position, contours, labels, and rough harmonic tendencies*, and to easily access and apply any combination of other (meta)data information to the pattern-matching process.

In Section 2 we introduce the corpus as well as our annotation methodology on metadata and descriptors, including structure, contour, and implied harmony. Section 3 introduces a pattern-matching algorithm that utilizes suffix arrays (for all melodies) and bit arrays (a selection of descriptors) to return matched results based on melodic *content* and controlled descriptor context. Sections 4 and 5 discuss the implementation and the results of examples of combined melody/descriptor queries, and Section 6 provides concluding remarks and addresses open perspectives.

2. THE ANNOTATED CORPUS OF SLOVENIAN FOLK SONGS

2.1 The Corpus

We are expanding the digital world of folk songs ([17, 40–42] and others) with a limited selection of tunes from the largest collection of Slovenian folk songs – SLP, *Slovenske ljudske pesmi*), which consists of 5 critically edited physical books, issued between 1970 and 2007 [11, 43–46]. Tunes belong to *narrative song* genre (Figure 1) or hybrids between narrative and lyric genres (resemble narrative form, but much shorter). These are divided into types (by lyric resemblance) and their variants.

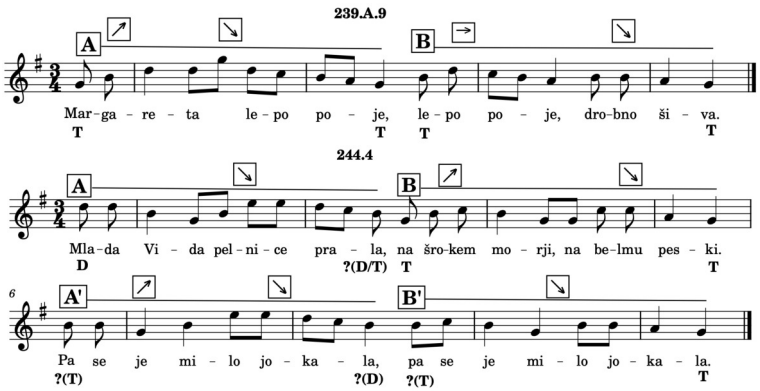
The fifth and last book [11] was edited and issued by the Slovenian Ethnomusicological Institute (Research Centre of the Slovenian Academy of Sciences and Arts, consecutively mentioned as GNI), which later digitized (OCR-ed pdf, musicXML, sib) by Matija Marolt and Matevž Pesek (FRI, University of Ljubljana). Most tunes are transcriptions of field recordings collected members of the institute, and external colleagues, such as Franc Kramar (1890–1959), Josip Dravec (1912–1996), and Stanko Vraz (1810–1851). Most tunes were collected and/or transcribed between the 1950s and 1970s. The earliest two (notated) transcriptions date back to 1819 and 1839, while the most recent transcription was completed in 2001. Together, there are 650 tune variants of 54 types and belong to family fates and conflicts topic, which represents about 34% of all Slovenian ballads [11]. Some tunes have one or two, while the most popular have 100+ variants (*Infanticide Bride*, *A Widower at His Wife's Grave*, *Step-mother and an Orphan*, and *Convicted Infanticide*). which mostly belong to Styria (166), followed by Upper Carniola (133) and Lower Carniola (99) regions.

Out of those, 418 are transcribed as monophonic, 218 [1] as homophonic, and 8 as mixed. About 70% were performed by a solo female singer. Instrumentally-accompanied examples are very rare. All tunes were transposed to G (major/minor) by the editors. Recent annotations include special notation symbols indicating deviations, such as slight disparity in pitch (higher/lower), duration (shorter/longer), and more [11].

2.2 Descriptors and Corpus Annotation

Out of the 418 monophonic tunes, 18 were excluded due to incomplete score information or incompatible encoding. Our final corpus contains 400 monophonic tune variants with detailed manual annotations, made by the first author and reviewed by all contributors. An average tune has about 9 bars and 30 notes. *Phrase boundaries* were annotated by curating the output of a simple heuristic relying

[1] Some melodies may have been harmonized by the annotator, making it unclear which singing line represents the original tune. This is common in transcriptions by Franc Kramar (1890-1959). The opposite can be true for monophonic transcriptions.



ID	POS	LBL	CT	H_S	H_E
239.A.9.1	F	A	↗↘	T	T
239.A.9.2	L	B	→↘	T	T

ID	POS	LBL	CT	H_S	H_E
244.4.1	F	A	↘	D	?(D/T)
244.4.2	M	B	↗↘	T	T
244.4.3	M	A'	↗↘	?(T)	?(D)
244.4.4	L	B'	↘	?(T)	T

Figure 1. Tunes from the SLP corpus with structural, contour, and implied harmony annotations. (Top) *The Death of the Bride Before Her Wedding*, 9th variant of tune type 239(A), transcribed by GNI in 1960. The first phrase (F), labeled A has a convex (↗↘) contour, whereas the second phrase has a horizontal-descending (→↘) contour as the first pitch of B phrase is about at the average compared to consecutive pitches of the phrase. Starting (upbeat) and ending implied harmonies (H_S, H_E) can be clearly labeled as a tonic, even though the strong beat on the first full measure is a D pitch. (Bottom) *The Widower at His Wife's Grave*, 44th variant of tune type 252, transcribed by Franc Kramar in 1913. Phrases A and B are approximately repeated as phrases A' and B', with changes in melody, contours, and harmonic functions. Harmonically, the first phrase starts on a clear dominant but is somewhat ambivalent at the end. The second phrase is the most stable on a tonic. The rest is slightly more ambivalent between the two degrees again, while, at the very end, the tune concludes on a tonic. (Right) A few descriptors that are associated with these tunes (see Table 1).

on pauses and punctuation marks, yielding 1502 phrases (median of 4 per tune, min. 2, max. 8).

Musical descriptors, listed in Table 1, describe either the full tune or phrases. The descriptors can carry both, *non-musical* and *musical* information.

Tune metadata [4] relies on transcribers' information, and the format aligns with the original sources, with some conversions made for data analysis convenience [11]. Tune phrases were annotated with *descriptors* across the following categories:

- *Phrase position.* This central annotation category establishes the relationship between other descriptors. Each phrase is annotated with a sequence number and position, such as first, middle, or last.
- *Structure.* Each phrase is assigned a label that describes the repetition of its melodic material within the verse. The first label is always A, followed by A, A', A+ (similar to B), A(X) (refrain-like A), or B. The same alphabetical progression is applied to subsequent phrases. The tunes have an average of 2.82 different labels (1 to 6 with symbols or 4 with letters only) and infrequent repetitions (Table 2). Each label appears only 1.34 times on average per tune.
- *Implied Harmony.* Using Western harmony to describe folk songs may be biased and controversial, but it is likely that Slovenian folk tunes and their transcribers have been exposed to the Western music system to some extent [47]. Approximate functions of tonic (T) or dominant (D) were annotated for about 60% of phrase beginnings and 50% of endings, with ambiguous cases marked as "?". To evaluate the validity of individual annotations, the inclusion of scale information (the count of distinct pitch classes) is provided.
- *Contour.* Diverging from only comparing the unreliable note-to-note melodic representation of oral music tradition, we use Huron's 9 types of melodic arches [13] (the most frequent being the convex contour ↗↘), where the starting and ending MIDI pitch value is compared against an average value of all intermediate MIDI pitch values.

[4] Metadata was collected for all 650 songs, including polyphonic compositions.

3. MATCHING MELODY WITH DESCRIPTORS

3.1 Pitch and Descriptor Representation

Each tune is subdivided into *phrases* as *pitch sequences* with *descriptors*, which are considered as a set of n phrases $P = \{p_1, p_2, ..., p_n\}$, and m phrase descriptors $\Delta^1, \ldots, \Delta^m$. Each descriptor Δ^t has a finite set of values $V(\Delta^t)$. A phrase p_i is associated with a *descriptor sequence* $d_i = (d_i^1, d_i^2, \ldots, d_i^m)$, where each d_i^t is in $V(\Delta^t)$.

For example, the following options:

$$\begin{cases} \{\Delta^1, \ldots, \Delta^5\} = \{\text{POS}, \text{LBL}, \text{CT}, \text{H}_S, \text{H}_E\} \\ V(\text{POS}) = \{\text{F}, \text{L}, \text{M}\} \quad V(\text{H}_E) = \{\text{T}, \text{D}, ?\} \\ V(\text{LBL}) = \{\text{A}, \text{B}\} \quad V(\text{H}_E) = \{\text{T}, \text{D}, ?\} \\ V(\text{CT}) = \{\nearrow\searrow, \rightarrow\searrow, \nearrow, \ldots\} \end{cases}$$

can describe phrase 239.9.A.1 (Figure 1) as:

$$p_{239.9.A.1} = gbddgdcbag \quad d_{239.9.A.1} = (\text{F}, \text{A}, \nearrow\searrow, \text{T}, \text{T}).$$

		Non-musical Metadata
tune ID	▷	Type, variant, other
type title	▷	Title or label
region	▷, ⋆	Region
annotator	▷	Name of the initial collector/transcriber
year	▷, ⋆	Year [2] of initial annotation.
singer	▷, ⋆	Singer sex and ensemble size
lyric	▷, ⋆	First line [3], first verse, structure
		Musical Descriptors
POS	⋆	Phrase position (**F**irst, **M**iddle, **L**ast)
NUM	⋆	Phrase number (1, 2, 3, 4, ...)
LBL	⋆	Phrase label (A, B, C, ...)
SYM	⋆	Phrase symbol (A', A", A +, ...)
CT_SPEC	◇	Huron's contours (↗, ↘, ↗↘, ↘↗, →, ↗→, ↘→, →↗, →↘)
H_S	⋆	Starting harmony (T, D, ?, ...)
H_E	⋆	Ending harmony (T, D, ?, ...)
TS	◇	Time signature (simple duple/triple ...)
SCALE	◇	Scale (8, 7, 6, ...)
		Musical Content
MEL		Melodic sequence (example: *gbddg*)

Table 1. Metadata, musical descriptors, and musical content. Manual annotations were done for phrase boundaries and descriptors marked with ⋆, while computed descriptors are marked with ◇. Descriptors marked with ▷ were collected by the initial transcribers and/or the GNI. Other descriptors, such as general contour, tone set, and leading tone, are also present in the dataset but not discussed here.

3.2 Melody and Descriptor Pattern Matching

The goal of *melody-and-descriptor matching* is to find all phrases (associated with their descriptors) matching in both the given *pitch pattern* and selected variation of *descriptor pattern*.

A *pitch pattern* pp is also a sequences of pitches. It matches a phrase p_i when pp is a factor of p_i, matching note-to-note. This definition currently permits no kind of deviation. For example, $pp = dcb$ matches $p_{239.9.A.1} = gbddg\underline{dcb}ag$.

A *descriptor pattern* is $dp = (dp^1, \ldots, dp^m) \in (V(\Delta^1) \cup \{\star\} \times \cdots \times V(\Delta^m) \cup \{\star\})$, where $\star$ is a "don't-care" symbol. It determines which descriptors are to be checked, and matches a descriptor sequence $d_i = (d_i^1, \ldots, d_i^m)$ if, for every $t = 1 \ldots m$, either $dp^t = \star$ or $dp^t = d_i^t$. For instance, the descriptor pattern $dp = (\star, \mathrm{A}, \nearrow\searrow, \star, \star)$ checks if the phrase label matches A, and contour matches ↗↘, but ignores the phrase position and harmonic functions. Thus dp matches $d_{239.9.A.1} = (\mathrm{F}, \mathrm{A}, \nearrow\searrow, \mathrm{T}, \mathrm{T})$ but does not match $(\mathrm{F}, \mathrm{B}, \nearrow\searrow, \mathrm{T}, \mathrm{D})$.

NP	instances		
2	20% (78)	AB (60)	AA (18)
3	6% (23)	ABC (11)	ABB (6)
4	**65% (261)**	**ABCD (112)**	ABAB (45)
5	2% (8)	–	–
6	6% (24)	ABCDCD (7)	AABABA (5)
8	1% (6)	ABCBCBCB (2)	–

Table 2. Out of 400 tunes, sorted according to the number of phrases (NP), the most common structure "ABCD" is present in 28% of all tunes. Label variants are ignored (A' is considered as A). There are no tunes with 7 or more than 8 phrases. Unique structures (–) are not reported.

3.3 Algorithm

For this matching problem, we first retrieve phrases and positions from a suffix array, in linear time, then filter these matches with bit-wise operators.

Pitch sequence matching with suffix array. Pitch sequences of P are concatenated to one sequence, separated by a symbol, such as $S_P = p_1\$p_2\$\ldots p_n\$$. An index data structure such as a compressed suffix array is computed and stored to retrieve all occurrences of a pitch sequence. When a query is matched at position k in S_P, the corresponding phrase p_i and its position in p_i is retrieved using a (pre-computed) bit-vector $\overline{S_p}$, and functions $rank_1(x, k)$ and $select_1(x, k)$ (respect. the number of occurrences of 1 in the prefix of length k of a bit-vector x, and the index of the k-th 1 in x). The bit-vector $\overline{S_p} = b_1 \ldots b_{|S_p|}$ is defined as $b_i = 1$ if $S_p = \$$, otherwise $b_i = 0$. Hence, the query occurs in phrase p_i at position j (within p_i) with $i = rank_1(\overline{S_p}, k)$, and $j = k - select_1(\overline{S_p}, i)$. Retrieving the list of phrases and positions of a query, pitch sequence q of length m is done in time $O(m + occ)$, where occ is the number of occurrences of q in S_p provided that $rank$ and $select$ operations are performed in constant time.

Descriptor pattern matching with bitwise operators. Each descriptor Δ^t can be represented by b^t bits, with $b^t = \lceil \log_2 |V(\Delta^t)| \rceil$, and each value $v \in V(\Delta^t)$ is associated to a bit-vector $\overline{v}$. Each descriptor sequence $d_i = (d_i^1, d_i^2, \ldots, d_i^m)$ is then stored as a bit-vector $\overline{d_i} = \overline{d_i^1} \ldots \overline{d_i^m}$. A descriptor pattern $dp = (dp^1, dp^2, \ldots, dp^m)$ is associated to two bit-vector masks $\mu(dp) = \mu^1 \ldots \mu^m$ and $\pi(dp) = \pi^1 \ldots \pi^m$, where

$$\begin{cases} \mu^t = \pi^t = 0 \ldots 0 \ (b^t \text{ bits}) & \text{if } dp^t = \star \\ \mu^t = 1 \ldots 1 \text{ and } \pi^t = \overline{dp^t} & \text{otherwise.} \end{cases}$$

Then, a descriptor pattern dp matches one descriptor d if and only if $(\overline{d} \text{ xor } \pi(dp))$ and $\mu(dp) = 0$. For example, if $dp = (\star, \mathrm{A}, \nearrow\searrow, \star, \star)$, then

$$\begin{aligned} \overline{d_{239.9.1.A}} &= 00 \cdot 0 \cdot 0010 \cdot 01 \cdot 01 \\ \mu(dp) &= 00 \cdot 1 \cdot 1111 \cdot 00 \cdot 00 \\ \pi(dp) &= 00 \cdot 0 \cdot 0010 \cdot 00 \cdot 00 \\ \hline \overline{d_{239.9.1.A}} \text{ xor } \pi(dp) &= 00 \cdot 0 \cdot 0000 \cdot 01 \cdot 01 \\ \overline{d_{239.9.1.A}} \text{ xor } \pi(dp) \text{ and } \mu(dp) &= 00 \cdot 0 \cdot 0000 \cdot 00 \cdot 00 \end{aligned}$$

Checking whether a descriptor dp matches a descriptor d is done in $O(1)$ time, provided that the bit-vectors fit in one machine word.

	↗↘	↘	↗	↘↗	↗→	↘→
First	24%	16%	**30%**	15%	6%	1.2%
Middle	**36%**	21%	15%	8%	10%	1.3%
Last	**44%**	32%	2%	5%	4%	11%
Total	**35%**	23%	16%	9%	7%	1.5%

Table 3. We show the most frequent Huron's contour types according to phrase position. Regarding general contour (data not shown), first phrases are mostly ascending (53%), while middle and last phrases are predominantly descending (63% and 89% respectively).

4. IMPLEMENTATION AND AVAILIBILITY

The descriptors and the algorithm were implemented in Python using `music21` [48] for music data manipulation, `bitarray` library for descriptors matching and a C++ library `sdsl-lite` [49] (used in Python with `pysdsl` library) for melodic matching. From the latter, we used `rank` and `select` methods, and `BitVector` and `SuffixArrayBitcompressed` classes. On a standard laptop from 2022, building suffix arrays and bit vectors on 1502 phrases takes less than 0.5 s. A single suffix array takes 49.4 KB, and bit vectors 1.7 KB. The longest melody/descriptor queries take about 100 ms. The annotations and the code are available on a git repository through open licences (Open Database License, Database Contents License, GPLv3+) at `algomus.fr/data` and `algomus.fr/code`. We are collaborating with partners in Slovenia to prepare the release of 400 melodies and additional ethnomusicological research.

5. RESULTS AND DISCUSSION

Almost all tunes (84%) in our collection revolve around a minimum of six distinct pitch classes, indicating the likely utilization of Western major/minor scales and modes. Approximately 67% of the tunes fall within the range of a major sixth (M6) to an octave (P8). Additional statistics in the annotated corpus examine phrase positions (Section 5.1). Specifically, the analysis focuses on the prevalent tune subtype 286.T1, annotated for melodic similarity. Findings from combined melody and descriptor queries are presented in Section 5.2.

5.1 Descriptors and Phrase Positions

The 400 tunes consist of 1502 phrases, split into first (400), middle (702), and last (400) positions. The labels, contours, and implied harmonies are strongly influenced by the phrase positions, as evident from Tables 2, 3, and 4. In general, phrases are convex or descending (see Table 3), while the *first phrases* are mostly ascending or convex, and less harmonically stable at their ends than beginnings. Conversely, the *last phrase* is almost never ascending, and more harmonically stable at its ends than beginnings. The *middle phrase* group is more divided and is more unstable.

		T	D	$?_T$	$?_D$	?
First	H_S	25%	**54%**	9%	<1%	12%
	H_E	22%	27%	15%	7%	**29%**
Middle	H_S	16%	**36%**	14%	6%	28%
	H_E	21%	16%	18%	6%	**40%**
End	H_S	19%	32%	10%	5%	**34%**
	H_E	**60%**	<1%	20%	None	19%
Total	H_S	19%	**40%**	11%	5%	25%
	H_E	**32%**	15%	18%	5%	32%

Table 4. Starting (H_S) and ending (H_E) harmonic functions in relation to phrase positions demonstrate a consistent pattern. Phrases typically initiate on a dominant (D) and conclude on a tonic (T). However, there is ambiguity with the functions ?, $?_T$, and $?_D$, as they can be interpreted as either T or D, which arises from the influence of previous and following pitch values or bars, making the exact annotation spot unclear.

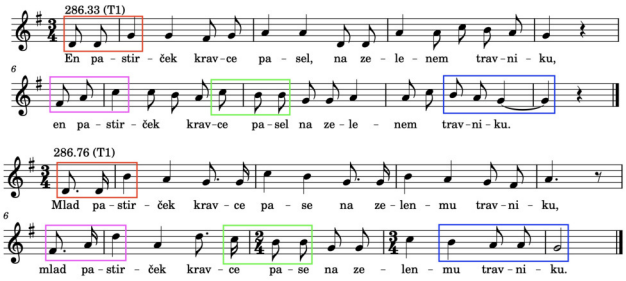


Figure 2. Two variants (out of 34) of the subtype 286.T1 with similar melodies with short melodic patterns in coloured squares.

We assume that the contrasting beginnings and endings of each verse offer pitch orientation for the singers, given the repetitive structure of narrative songs. The contour relationship between the first, middle, and last phrases supports the notion that "what goes up is likely to come down," as proposed by Huron [13].

5.2 Case Study: Subtype 286.T1, *Infanticide Bride*

Our dataset includes 103 monophonic variants of the widely known "Infanticide Bride" theme in European folk song tradition. Subtype 286.T1 consists of 34 tunes selected for their melodic similarity (Figure 2), which often exhibit similar patterns, such as the *fad* as a start middle phrase pattern or the *bag* as a last phrase ending pattern.

We have developed combined melody/descriptor queries to represent certain phrases of subtype 286.T1. These queries are evaluated as a binary classification problem: Can we accurately identify the 34 initial phrases of 286.T1 and distinguish them exclusively from others?

Pattern design and matching. Table 5 demonstrates that simple melody queries with 1 to 3 notes achieve reasonable recall rates (50%-80%) but limited precision. Refining the queries with descriptors improves precision and relevance,

leading to enhanced F_1 measures. The ddb melody query alone produces 93 matches, but 75 of them are "false positives" unrelated to the first phrase of 286.T1 tunes. Incorporating a phrase position descriptor (F, first) improves the query, while adding relevant contour (↗) and starting harmonic information further enhances specificity. This comprehensive query results in only 2 false positives, achieving a precision of up to 0.88, with minimal sensitivity loss. The ag pattern in the last phrase, characterized by a convex contour and a harmonic ending, is a noteworthy example. Given the enhanced harmonic stability typically found in verse endings, the inclusion of the H_ST as a stable harmony descriptor proves to be effective in this context. Including too many or irrelevant descriptors leads to poor results. For instance, the cbb pattern is primarily found at the end of the middle phrase. However, requiring a stable harmonic framework (H_ET) for middle endings reduces precision, as it is less common in those positions (Table 4). Another interesting instance is the fad, occurrences of which are almost evenly split into two contours. If we matched (fad, M, ↗↘ or ↗→), we would get 23 true positives and a precision of 0.82 with a recall of also 0.82. The algorithm should be extended to accommodate the matching of a subset of multiple descriptors within the same category, rather than solely relying on one descriptor.

Patterns as building blocks. Melody/descriptor patterns have versatile applications beyond classification. In our case, the most effective queries incorporate position descriptors, indicating that we should view phrase *building blocks* as patterns. It is noteworthy, that studying the "false positives" (matches outside of 286.T1) is expected to yield intriguing results, shedding light on the transmission of music material among tunes and vice versa. For instance, the ag pattern in the last phrase, exhibiting a ↗↘ contour and ending with H_ET, is not only specific to 286.T1 but also appears in 14 tunes of type 252 (*A Widower at His Wife's Grave*). The shared section of the melodic line in the two tunes has identical descriptors, although its positions may vary (scores not shown). Comparing outcomes across multiple corpora would provide insight into the unique musical characteristics of (Slovenian) folk songs.

5.3 Discussion

Our study explored tune structures and melodic patterns, finding that combining melodic content with descriptors provides valuable insights into the characteristics of Slovenian folk songs. However, not all descriptors fit universally to describe all content, and vice versa. In contrast to the usual transferability of folk song melodies, our case study indicates that the melody of 286.T1 was not easily transferable, possibly due to its popularity in multiple regions. Further inter- and intra-corpus research is needed to investigate this distinctive characteristic. We also show, that individual melodic extracts alone lack the specificity required for a comprehensive description of a full phrase or tune. Strong correlations were found for descriptors like contour. Expanding the dataset is needed to comprehensively explore the relationship between lyrics and melodies, including the observed descending shape in the last phrase, potentially reflecting or corresponding with speech characteristics [50, 51].

Query (melody + descriptors)			TP	FP	FN	Prec.	Rec.	F_1
d	None	1 (34)	28	1191	6	0.02	**0.82**	0.04
d	F		28	327	6	0.08	0.82	0.14
d	F, H_SD		27	189	7	0.12	0.79	0.22
d	F, ↗, H_SD		21	48	13	0.30	0.62	**0.41**
ddb	None	1 (34)	18	**75**	16	0.19	**0.53**	0.28
ddb	F		18	32	16	0.36	0.53	0.43
ddb	F, H_SD		17	21	17	0.45	0.50	0.47
ddb	F, ↗, H_SD		14	**2**	20	**0.88**	0.41	**0.56**
fad	None	3 (33)	24	24	9	0.50	**0.73**	0.59
fad	M		24	14	9	0.63	0.73	**0.68**
fad	M, ↗↘		**11**	2	22	**0.85**	0.33	0.48
fad	M, ↗→		**12**	3	21	**0.80**	0.36	0.50
cbb	None	3 (33)	25	71	8	0.26	**0.76**	0.39
cbb	M		25	39	8	0.39	0.76	**0.52**
cbb	M, ↗→		11	3	22	0.79	0.33	0.47
cbb	M, H_ET		**1**	3	32	0.25	**0.03**	0.05
ag	None	4 (34)	27	481	7	0.05	**0.79**	0.10
ag	L		27	165	7	0.14	0.79	0.24
ag	L, ↗↘		23	54	11	0.30	0.68	0.41
ag	L, ↗↘, H_ET		23	51	11	0.31	0.68	**0.43**

Table 5. Evaluation of melody/descriptor queries seen as classification queries intended to match phrases 1, 3, and 4 of the melodic tune subtype 286.1 (34 first and last phrases, 33 third phrases) against all 1502 phrases of the dataset. We computed True Positives (TP), False Positives (FP), False Negatives (FN), and from those, precision, recall, and F_1-score. Bold values are discussed in the text.

6. CONCLUSION AND PERSPECTIVES

By integrating descriptor information into melodies, we gain a deeper understanding of the observed music. Our findings indicate a strong dependency of many descriptors on phrase positions, and that combining melody and descriptors enhances precision compared to using melody alone. Our algorithm efficiently matches melodies and descriptors, which can be extended beyond our proposed selection. Lastly, we released annotations of Slovenian folk songs, a yet underrepresented corpus in the MIR community.

Our current plans primarily involve releasing the corpus of these tunes, accompanied by comprehensive ethnomusicological commentary. In addition, future work should prioritize improving the algorithm's usability for non-computational users, expanding the existing annotations of descriptors, and implementing the capability to perform combined query searches with approximate matching for melodies and descriptors. Our study (and corpus) may be used as supporting data for new algorithms of phrase segmentation, tune structure analysis, and harmony tasks including semi-automatic annotation.

Acknowledgements. We extend our gratitude to Matija Marolt, Matevž Pesek, and current as well as past members of the Institute of Ethnomusicology ZRC SAZU for their pivotal role in digitizing and curating the corpus from field recordings, notations and notes. We also appreciate the valuable input and comments provided by Dinh-Viet-Toan Le and the Algomus team, Patrick E. Savage and the members of Comp Music Lab, and the anonymous reviewers, which greatly contributed to the development of this paper.

7. REFERENCES

[1] A. P. Merriam, *The Anthropology of Music*. Northwestern University Press, 1964.

[2] K. Blaukopf, *Musical Life in a Changing Society: Aspects of Music Sociology*. Hal Leonard Corporation, 1992.

[3] I. Mills, "The heart of the folk song," *Canadian Folk Music Journal*, vol. 2, pp. 29–34, 1974.

[4] S. P. Bayard, "Prolegomena to a study of the principal melodic families of British-American folk song," *The Journal of American Folklore*, vol. 63, no. 247, pp. 1–44, 1950.

[5] Z. Kumer, *Vloga, zgradba, slog slovenske ljudske pesmi*. Založba ZRC, 1996.

[6] P. van Kranenburg, A. Volk, and F. Wiering, "On operationalizing the musicological concept of tune family for computational modeling," *Proceedings of Supporting Digital Humanities: Answering the unaskable*, 2011.

[7] T. Rice, *Modeling Ethnomusicology*. Oxford University Press, 2017.

[8] J. R. Cowdery, "A fresh look at the concept of tune family," *Ethnomusicology*, vol. 28, no. 3, pp. 495–504, 1984.

[9] C. Pendlebury, "Tune families and tune histories: Melodic resemblances in British and Irish folk tunes," *Folk Music Journal*, vol. 11, no. 5, pp. 67–158, 2020.

[10] 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.

[11] M. G. Kaučič, M. Klobčar, Z. Kumer, U. Šivic, and M. Terseglav, *Slovenske ljudske pesmi V.: Pripovedni pesmi*. Založba ZRC, 2007, vol. 5.

[12] S. Ahlbäck, "Melody beyond notes: A study of melody cognition," Ph.D. dissertation, Göteborgs Universitet, 2004.

[13] D. Huron *et al.*, "The melodic arch in Western folksongs," *Computing in Musicology*, vol. 10, pp. 3–23, 1996.

[14] S. d. l. Ossa, *A Basic Guide to Folksong Analysis*. Budapest: Liszt Academy of Music, 2019.

[15] A. Lomax, *Folk Song Style and Culture*. Routledge, 2017.

[16] A. Lomax and N. Berkowitz, "The evolutionary taxonomy of culture: A few behavioral factors account for the regional variation and evolutionary development of culture," *Science*, vol. 177, no. 4045, pp. 228–239, 1972.

[17] A. Wood, K. R. Kirby, C. Ember, S. Silbert, S. Passmore, H. Daikoku, J. Mcbride, F. Paulay, M. Flory, J. Szinger *et al.*, "The Global Jukebox: A public database of performing arts and culture," *PLOS One*, vol. 17, no. 11, 2021. [Online]. Available: https://doi.org/10.1371/journal.pone.0275469

[18] A. L. Wood, *Songs of Earth: Aesthetic and Social Codes in Music*. University Press of Mississippi, 2021.

[19] P. E. Savage, "Alan Lomax's cantometrics project: A comprehensive review," *Music & Science*, vol. 1, 2018.

[20] X. Serra, "The computational study of a musical culture through its digital traces," *Acta Musicologica*, vol. 89, no. 1, pp. 24–44, 2017.

[21] K. Neubarth, I. Goienetxea, C. G. Johnson, and D. Conklin, "Association mining of folk music genres and toponyms," *International Society for Music Information Retrieval Conference (ISMIR 2012)*, pp. 7–12, 2012.

[22] K. Neubarth and D. Conklin, "Contrast pattern mining in folk music analysis," in *Computational Music Analysis*. Springer, 2016, pp. 393–424.

[23] D. Conklin, "Antipattern discovery in folk tunes," *Journal of New Music Research*, vol. 42, no. 2, pp. 161–169, 2013.

[24] P. van Kranenburg, "Computational approach to content-based retrieval of folk song melodies," Ph.D. dissertation, Utrecht University, 2010.

[25] P. van Kranenburg and F. Karsdorp, "Cadence detection in western traditional stanzaic songs using melodic and textual features," in *International Society for Music Information Retrieval Conference (ISMIR 2014)*, 2014, pp. 391–396.

[26] C. McKay, J. Cumming, and I. Fujinaga, "jSymbolic 2.2: Extracting features from symbolic music for use in musicological and MIR research," in *International Society for Music Information Retrieval Conference (ISMIR 2018)*, 2018, pp. 348–354.

[27] D. Conklin and M. Bergeron, "Discovery of contrapuntal patterns," in *International Society for Music Information Retrieval Conference (ISMIR 2010)*, vol. 2010, 2010, pp. 201–206.

[28] I. Y. Ren, H. V. Koops, A. Volk, and W. Swierstra, "In search of the consensus among musical pattern discovery algorithms," in *International Society for Music Information Retrieval Conference (ISMIR 2017)*, 2017, pp. 671–678.

[29] E. Cambouropoulos, "Musical rhythm: A formal model for determining local boundaries, accents and metre in a melodic surface," in *Joint International Conference on Cognitive and Systematic Musicology*, 1996, pp. 277–293.

[30] O. Lartillot, "Automated motivic analysis: An exhaustive approach based on closed and cyclic pattern mining in multidimensional parametric spaces," in *Computational Music Analysis*, 2016, pp. 273–302.

[31] M. Mongeau and D. Sankoff, "Comparison of musical sequences," *Computers and the Humanities*, vol. 24, no. 3, pp. 161–175, 1990.

[32] P. E. Savage, "Measuring the cultural evolution of music: Cross-cultural and cross-genre case studies," in *PsyArXiv*, 2020 (preprint).

[33] P. E. Savage, S. Passmore, G. Chiba, T. E. Currie, H. Suzuki, and Q. D. Atkinson, "Sequence alignment of folk song melodies reveals cross-cultural regularities of musical evolution," *Current Biology*, vol. 32, no. 6, pp. 1395–1402, 2022.

[34] T. Eerola and M. Bregman, "Melodic and contextual similarity of folk song phrases," *Musicae Scientiae*, vol. 11, no. 1, pp. 211–233, 2007.

[35] C. Anagnostopoulou, M. Giraud, and N. Poulakis, "Melodic contour representations in the analysis of children's songs," in *International Workshop on Folk Music Analysis (FMA 2013)*, 2013, pp. 40–43.

[36] J. McBride, A. T. Tierney, P. Pfordresher, J. Six, S. Fuji, and P. E. Savage, "Are pitches discrete? an information-theoretic framework and a corpus study," in *International Conference on Music Perception and Cognition and Triennial Conference of the European Society for the Cognitive Sciences of Music (ICMPC-ESCOM 2021)*, 2021.

[37] C. Finkensiep, K. Déguernel, M. Neuwirth, and M. Rohrmeier, "Voice-leading schema recognition using rhythm and pitch features," in *International Society for Music Information Retrieval Conference (ISMIR 2020)*, 2020, pp. 520–526.

[38] E. Cambouropoulos, "The harmonic musical surface and two novel chord representation schemes," in *Computational Music Analysis*. Springer, 2016, pp. 31–56.

[39] A. Marsden, "Representing melodic patterns as networks of elaborations," *Computers and the Humanities*, vol. 35, no. 1, pp. 37–54, 2001.

[40] H. Schaffrath, "The Essen Associative Code: A code for folksong analysis," in *Beyond MIDI: The handbook of musical codes*. MIT Press, 1997, p. 343–361.

[41] T. Eerola and P. Toiviainen, "Suomen kansan esävelmät – finnish folk song database," 1999. [Online]. Available: http://esavelmat.jyu.fi/

[42] P. van Kranenburg, M. De Bruin, and A. Volk, "Documenting a song culture: the Dutch song database as a resource for musicological research," *International Journal on Digital Libraries*, vol. 20, no. 1, pp. 13–23, 2019.

[43] Z. Kumer, M. Matičetov, B. Merhar, and V. Vodušek, *Slovenske ljudske pesmi I: Pripovedni pesmi*. Ljubljana: CGP Delo, 1970, vol. 1.

[44] Z. Kumer, M. Matičetov, and V. Vodušek, *Slovenske ljudske pesmi II: Pripovedni pesmi*. Ljubljana: Slovenska matica, 1981, vol. 2.

[45] M. Terseglav, I. Cvetko, M. Golež, and J. Strajnar, *Slovenske ljudske pesmi III: Pripovedni pesmi*. Ljubljana: Slovenska matica, 1992, vol. 3.

[46] ——, *Slovenske ljudske pesmi IV: Pripovedni pesmi*. Ljubljana: Slovenska matica, 1992, vol. 4.

[47] V. Vodušek, *Etnomuzikološki članki in razprave*. Založba ZRC, 2003.

[48] M. S. Cuthbert and C. Ariza, "music21: A toolkit for computer-aided musicology and symbolic music data," *International Society for Music Information Retrieval Conference (ISMIR 2010)*, pp. 637–642, 2010.

[49] S. Gog, T. Beller, A. Moffat, and M. Petri, "From theory to practice: Plug and play with succinct data structures," in *International Symposium of Experimental Algorithms (SEA 2014)*, 2014, pp. 326–337.

[50] Y. Ozaki, A. Tierney, P. Pfordresher, J. Mcbride, E. Benetos, P. Proutskova, G. Chiba, F. Liu, N. Jacoby, S. Purdy *et al.*, "Similarities and differences in a global sample of song and speech recordings [stage 2 registered report]," *PsyArXiv*, 2022 (preprint).

[51] P. Albouy, S. A. Mehr, R. S. Hoyer, J. Ginzburg, and R. J. Zatorre, "Spectro-temporal acoustical markers differentiate speech from song across cultures," *bioRxiv*, 2023 (preprint).

HOW CONTROL AND TRANSPARENCY FOR USERS COULD IMPROVE ARTIST FAIRNESS IN MUSIC RECOMMENDER SYSTEMS

Karlijn Dinnissen
Utrecht University, The Netherlands
k.dinnissen@uu.nl

Christine Bauer
Paris Lodron University Salzburg, Austria
christine.bauer@plus.ac.at

ABSTRACT

As streaming services have become a main channel for music consumption, they significantly impact various stakeholders: users, artists who provide music, and other professionals working in the music industry. Therefore, it is essential to consider all stakeholders' goals and values when developing and evaluating the music recommender systems integrated into these services. One vital goal is treating artists fairly, thereby giving them a fair chance to have their music recommended and listened to, and subsequently building a fan base. Such artist fairness is often assumed to have a trade-off with user goals such as satisfaction. Using insights from two studies, this work shows the opposite: some goals from different stakeholders are complementary. Our first study, in which we interview music artists, demonstrates that they often see increased transparency and control for users as a means to also improve artist fairness. We expand with a second study asking other music industry professionals about these topics using a questionnaire. Its results indicate that transparency towards users is highly valued and should be increased.

1. INTRODUCTION

Music is most consumed on streaming services nowadays [1]. These services often have music recommender systems (MRS) integrated to provide personalized recommendations to users. Unfortunately, those systems might disadvantage some artists due to biases in the data, the system, and society. This could lead to unfairness for artists, e.g., through reduced visibility and opportunities [2–5].

To mitigate such issues, it is essential to understand and involve the stakeholders affected by MRS, and assess whether we are solving the right problems [6]. However, researchers have rarely directly reached out to these stakeholders inquiring how exactly they are affected, and what they value and desire in these systems. Limited work has been done on the user side, e.g., interviewing users about fairness in recommender systems (RS) in general [7], diversity in MRS [8], and the impact of MRS on listeners [9].

© K. Dinnissen and C. Bauer. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** K. Dinnissen and C. Bauer, "How Control and Transparency for Users Could Improve Artist Fairness in Music Recommender Systems", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.*, Milan, Italy, 2023.

Work considering the artists' view is equally scarce. Exceptions are two interview-based studies on artists' perspective on fairness in MRS [10], and on playlists in music streaming services [11]. Moreover, to date, no research consults other music industry professionals. Such indirect stakeholders are often ignored when designing and evaluating systems such as MRS, even though these systems affect them as well [12]. In the case of MRS, some professionals come into direct contact with streaming services and embedded MRS, e.g., when working in a publishing or concert booking role. More indirectly, the success of a professional in an artist's team depends on that of the artist, which in part depends on streaming services.

This work focuses on two topics that are not described in existing artist-focused work: (i) *transparency* for users and (ii) giving more *control* to users. These topics emerged unprompted in interviews with artists (Study 1), which aimed to understand what artists consider to be fair in music streaming services and embedded MRS, and which role artists envision for music streaming platforms with regard to fairness, diversity, and transparency. While the general results of this study are described in Dinnissen & Bauer [13], in the work at hand, we zoom in on transparency and control for users because artists frequently mentioned these concepts as a means to increase artist fairness. This suggests that—contrary to what is often suggested [14–16]—there is not necessarily a trade-off between user and artist goals; they could even be complementary. Inspired by these insights, we subsequently query industry professionals through a questionnaire (Study 2).

We address the following research questions (RQs):

- **RQ1**: How do (i) artists and (ii) other music industry professionals view the current level of transparency and control for users on music streaming services?
- **RQ2**: Which role do artists see for user transparency and control in improving artist fairness?
- **RQ3**: What are artists' user interface (UI) suggestions to improve transparency and control?

This work offers insight into several perspectives on transparency and control of MRS for users. Artists think both should be increased and give concrete UI suggestions to achieve this. Industry professionals agree that transparency for users should be increased but offer a more nuanced view on control. We emphasize two key points: (i) user and artist goals should not always be viewed as

trade-offs, as some can be complementary, and (ii) directly involving a diverse set of stakeholders is essential in music information retrieval (MIR) research and development, to integrate their values and needs adequately.

2. RELATED WORK

2.1 Fairness in Music Recommender Systems

Fairness in MRS is increasingly receiving attention in the RS and MIR research communities [17, 18], as music streaming services and their integrated MRS significantly influence the music landscape [19]. One challenge here is that fairness is a human judgment value with many definitions and factors at play that do not directly translate into RS evaluation metrics [3]. Hence, research generally focuses on specific, often demographics-based fairness dimension(s) [5], such as nationality (e.g., [20]) and gender (e.g., [4, 21]). Here, a system is generally considered fair on a dimension if it upholds *group fairness*, a concept in which several groups of people are defined (e.g., based on their nationality), and the system should not give anyone a lesser experience based on their belonging to one group.

Fairness research in the music domain covers users (i.e., consumers), artists (i.e., item providers), or both simultaneously (for an overview, see [2]). Frequently mentioned issues for artists are *popularity bias*, a phenomenon where already popular items are recommended more often than others (e.g., [22, 23]), and the *item cold-start problem*, denoting difficulty in accurately recommending new items due to lack of previous interactions (e.g., [24]). These issues particularly affect new or less well-known music acts. For users, goals such as satisfaction are often considered rather than their fairness desires. Still, users indicate that in RS in general, provider fairness is important to them [7].

Other stakeholders to consider are platforms offering MRS [25, 26], music labels [27], and other music industry professionals who come into contact with or are impacted by music streaming services (e.g., concert bookers, artist managers, event producers). To the best of our knowledge, no work directly addresses the latter stakeholder group's view on how their values should be integrated into MRS.

2.2 Transparency for Users

Like fairness, RS transparency is increasingly valued by users and item providers alike [10, 28], with its societal relevance resulting in EU-wide legislation [29]. Transparency is often offered through interpretable or explainable RS, which can educate users on inner RS workings [7, 30]. To properly gauge RS fairness, transparency is considered a prerequisite. In Sonboli et al. [7], users indicate desiring insight into the fairness goals of organizations that offer RS. Ferwerda et al. [31] show that MRS users were more satisfied if they perceived a playlist as fair (here, focusing on artist popularity), even if they could not identify which playlist was more fair according to objective measures.

Insight into the inner workings of MRS, and therein considered fairness dimensions, could be offered on multiple levels and with different amounts of detail [30, 32]. On the broadest level, music streaming services could share relevant business rules. On an abstract RS model level, global explanations can show overall tendencies on different dimensions (including fairness) [33]. Local model explanations are also possible, e.g., on the level of specific songs, artists, or playlists [33–35]. Different user personalities and cognition needs should be considered here [36], ideally allowing users to choose the type of explanations [7]. When promoting lesser-known artists, persuasive explanations might increase how users rate their recommendations [16]. However, users indicate they do not want to be (unintentionally) manipulated through explanations, even in a fairness context [7]. Finally, visualizations might also bring model logic to light [32], though textual explanations might be more effective in the MRS domain [34]. Yet, such transparency-enhancing functionality has rarely been implemented into user-facing parts of music streaming services, especially on the model level.

2.3 Giving Users Control

User control is considered an essential quality of an effective RS, as it positively affects user trust and satisfaction [28]. While transparency can provide insight into fairness, control gives users the agency to change their recommendations based on their values and goals, which could include fairness. Users indicate they want to choose whether they want more personalized recommendations that might be less fair for item providers, or less personalized in favor of fairness [7]. In the music domain specifically, research on user control often aims at exploration, discovery, or diversification tasks (e.g., [8, 32, 37–41]), with no works to date focusing on fairness. Still, if users' listening behavior becomes more diverse (i.e., they start engaging with a broader range of music items), this could also contribute to artist fairness if that range includes less popular or historically underrepresented artists [15, 42].

Like transparency, user control can be implemented on different system levels in a MRS. Literature differentiates between low-level control on recommendation data level (playlist, play and like buttons, rating), middle-level control on user profile level (e.g., with tags or sliders), and high-level control on algorithm parameter level [32]. When enabling such functionality, the extent to which a user can take control should be personalized to keep cognitive load and complexity at an acceptable level [32, 37].

Even though research demonstrates how users can be given control and how this contributes to recommendation acceptance and user satisfaction, in practice, users typically have little control in widely used MRS. Essentially, RS providers are in the main position to control the system, and with it, the items recommended [43, 44]. In the music industry, some item suppliers (e.g., major labels) may be in a strong position to shift the control to their side [43].

3. METHODS

We employed two studies: Study 1 with artists and Study 2 with other music industry professionals. Here, we describe

Code	Age	Gender	Audience reach	Genre
P1	26–35	Male	Local	Hip-Hop
P2	26–35	Male	National	Rock/Pop
P3	26–35	Male	Local	Rock/Punk/Metal
P4a,b	26–35	Male	Local (a) National, Local (b)	Hardcore/Rock/Blues (a), Indie/Metal (b)
P5	26–35	Male	Internat.	Dance
P6	18–25	Non-binary	Local	Pop
P7a,b	46–55	Female (a), Male (b)	National	Alt. Pop
P8a,b	56–65	Female	N/A	Folk/World
P9	18–25	Non-binary	Local	Rock/Pop/Folk
P10	26–35	Male	Local	Neoclassical
P11	36–45	Female	Local	80's Alt. Synthpop
P12	18–25	Female	Local	Metal
P13	26–35	Female	(Inter)nat.	Indie-pop Alt.
P14	36–45	Male	National	Many

Table 1. Study 1 self-reported participant information.

both studies' methods and outline our analysis approach.

3.1 Study 1: Interviews with Artists

We conducted 14 interviews with currently active music artists in the Netherlands from January to March 2022. We reached out to new participants until we reached a high level of thematic saturation [45]. For music groups, we offered the opportunity to join the interview with two members. This resulted in 3 interviews with two members and 11 interviews with individual artists (Table 1).

The research setup was based on the one used by [10], starting with a metadata questionnaire and a short presentation about MRS. Then, we conducted a semi-structured interview (52 minutes on average). Questions, outlined in detail in Dinnissen & Bauer [13], covered a broad range of topics: transparency for artists, artist control over recommendations, reaching an audience, popularity bias, diversity, gender balance, influencing users' behavior, localization, repertoire size, royalty distribution, and impact of the COVID-19 pandemic. By using open questions, we encouraged an open conversation rather than a predefined one, leaving space for artists to add insights.

We recorded, transcribed, and pseudonymized the audio of the interviews. We used a Qualitative Content Analysis [46] for which we based the annotation scheme on the codes used in [10] (deductive) and then adapted the codes based on the interview content (inductive). Three annotators coded the transcripts, with two inter-annotator sessions indicating a high level of inter-annotator agreement. In the work at hand, we focus on the interview parts related to transparency and control for users. Therefore, we analyze results under respective codes 'Transparency' (top-level)—'Towards user', and 'Control' (top-level)—'For users'. As mentioned in Section 1, neither topic was explicitly addressed in the questions. Both organically came up when discussing experiences and fairness.

3.2 Study 2: Questionnaires

To reach a considerable sample of music industry professionals, we used questionnaires as our data collection approach in Study 2. [1] We collected 35 responses, all filled in on tablets, from attendees at Eurosonic Noorderslag, a major European conference for music industry professionals held in January 2023. [2] 12 participants identified as women, 22 as men, and 1 participant refrained from stating their gender. Participants were from 7 European countries. When asked about their current professional role(s), participants indicated education (10), technology (7), event production (6), bookings (5), research/science (4), marketing/PR (4), artist (3), legal/policy (3), artist representation (2), and other (7). 11 participants indicated more than one role, and 'artist' was not the sole role for any participant.

In the questionnaire, we address the wide variety of topics from Study 1, this time from several points of view (i.e., artist, MRS user, and participants' own as an industry professional). For this work, again, we focus on the questions that relate to transparency and control for users (see Table 2). We used a 5-point Likert-scale answering format and also offered the options to indicate 'Don't know / prefer not to answer', skip a question if desired, and add comments. For the topics at hand, no comments were added. As one participant skipped all four questions on these topics, we present results for 34 participants.

4. RESULTS AND DISCUSSION

We outline the results of our studies going from our research questions. Insights from both Study 1 and Study 2 are used to answer RQ1, whereas RQ2 and RQ3 zoom in further on the results of Study 1.

4.1 RQ1: Transparency & Control for Users

For RQ1, we present insights from artists and other music industry professionals about current transparency and control for users within music streaming services.

Transparency—Artist view. All participants indicated using music streaming services both as a consumer and artist, distinguishing clearly between those two roles in their answers. In some cases, their views as an artist were similar to those from a user perspective. On transparency, they remarked that MRS deployed in streaming services are opaque to both artists and users. Here, we focus on the latter, which came up in several interviews despite no question being dedicated to this point of view. Opaqueness of MRS towards users was often stated as a fact: *"As a user [...] you really have no idea what is recommended to you. You are kind of cool with all of it because they do a pretty good job, those algorithms."* (P6)

Artists called for more transparency on especially fairness objectives and diversity in recommendations. Some also noted that if users lack insight into MRS, they have no way of knowing whether and how their taste is being influenced: *"It would be proper for a platform to show how it works, and that you as a listener would also... know? [...] As I think it influences [...] our listening behavior—which is not necessarily our taste—a lot."* (P13)

[1] Study 2 materials can be accessed at [47].
[2] `https://esns.nl/en/`

Transparency—Music industry professional view. From Study 2, we show results for 34 industry professionals for the two questions dedicated to user transparency (Table 2, Q1+2). Responses on whether personalized MRS are transparent to users were spread ($SD = 1.37$), with 32% of participants somewhat agreeing, but also 26% of participants strongly disagreeing. Participants tend toward a negative view ($Mean = 2.79$) on current transparency towards users, displaying similar tendencies as artists.

A stronger consensus can be found on whether personalized MRS should be made more transparent. Here, no participants opted for (slightly) disagree; only one participant chose the neutral option, and the other participants either somewhat agreed (56%) or strongly agreed (41%). This shows a clear, almost unanimous call for more transparency of MRS, mirroring this call from artists in Study 1.

Discussion. From these results, we deduce that both stakeholder groups highly value transparency for users in MRS, think it is currently lacking, and desire improvement. All in all, artists mainly focus on system-level information (versus, e.g., song-level), which could be shared through global explanations [34]. As in Sonboli et al. [7], artists mention an educational component, noting users do not know how MRS work, which objectives are incorporated, and how MRS influence their listening behavior.

Control—Artist view. In Study 1, the topic of control for users frequently surfaced when artists discussed their own experiences as streaming service users: being unhappy with their recommendations, and having no options to modify them: *"Truly every week [certain act] is added to my Release Radar, every week I dislike it, I disliked [their profile], and it still appears every week. How?"* (P2)

"You start [on YouTube] with a very small band, and eventually you always end up, let's say, at... Metallica, at Rock Im Park, you know. So I never find that useful." (P3)

Increased user control was not only mentioned as a solution to such problems, but also as a means to generally improve MRS by allowing users to provide more information on their preferences: *"I know plenty of people who really enjoy listening to the same music all the time and especially don't want to hear anything new. [...] If they'd solely be presented with a lot of different styles, they'd probably think: 'that's it, I'll go somewhere else, this is too much for me'. So it would be very cool if you could indicate [yourself], from zero to ten, 'I am very experimental'."* (P4a)

Nevertheless, some artists viewed actively searching for specific music on streaming services as effectively also 'taking control': *"If I want to listen to super obscure Hip-Hop or something, then there are [play]lists for that. You do need to know those exist [...], but they contain all kinds of new things I didn't know about before."* (P1)

P2 did note that not all users know how to find such lists or want to put in such effort: *"I feel like people very often simply listen to what they are told. [...] 30 years ago, radio dictated what people could listen to in the car, so people just listened to that, and now Spotify is doing it."* (P2)

Control—Music industry professional view. Resulting from Study 2, we show responses from 34 participants in Table 2 (Q3+4). Contrary to artists, industry professionals were less unanimously negative about the current extent to which users can control their recommendations in MRS. Regarding users' influence on general recommendations, participants were divided on the topic, with 38% indicating being somewhat or strongly dissatisfied, and 41% indicating they were somewhat or strongly satisfied.

On the extent to which users can influence their personal playlists, participants responded slightly more positively ($Mean = 3.38$), with 29% indicating they were somewhat or strongly dissatisfied, and 56% indicating they were somewhat or strongly satisfied. Overall, dissatisfaction with current user control seems less pronounced in this stakeholder group than for artists.

Discussion. Artists expressed dissatisfaction with current control over recommendations, as well as the lack of agency for users to change them. Some works (e.g., [44]) suggest that streaming services' business interests are a possible reason for limited control. Still, the added value artists see for increased control aligns with frequently discussed user goals, such as exploration [37, 39, 41] and discovery [32]. The industry professionals' view was more nuanced, with about half of the participants being satisfied with the extent to which users can currently control MRS.

4.2 RQ2: Role of Transparency & Control in Improving Artist Fairness

For RQ2, we focus on why artist fairness could be improved by increasing transparency and control for users, according to our participants. As this emphasis was initiated by participants from Study 1 but out of scope for Study 2, we focus on insights from Study 1.

Artists mainly mentioned transparency towards users in the context of algorithmically generated and 'curated' (i.e., created by an editor) playlists. These could be created with specific fairness goals in mind, which should be clearly communicated to the user. Such playlists could counter biases in MRS by presenting the user with more diverse music, e.g., highlighting music from (historically) underrepresented artists. P7b mentioned this could also be a way to highlight older repertoire, and P9 noted: *"An older album can, of course, still be new to someone."* (P9)

However, P8a+b and P13 remarked that solely offering playlists with a designated fairness goal would not bring any lasting shift in user behavior: *"If you put all [women] in one list, then it is a list again. Then it effectively becomes some kind of subgenre, while gender actually transcends genre, and... it should not be an issue at all anyway."* (P13)

A second suggestion was giving insight into all current playlists regarding certain ratios such as artist gender or ethnicity. With such insights, users could be made more aware of current inequalities, and make more informed and fairer decisions based on their own values: *"I think it is important for creators that users know what they are choosing. [...] Such transparency is missing completely. So I think it would be better if [streaming services] were transparent, like: all [play]lists contain this many women, this many men, this many black people, this many white peo-*

No.	Question	Min	Max	Median	Mean	SD
Q1	For users of streaming services, I feel like it is clear for which reason(s) specific music is recommended to them.	1	5	3	2.79	1.37
Q2	For users of streaming services, I feel like it is important to make it more clear for which reason(s) specific music is recommended to them.	3	5	4	4.38	0.54
Q3	For users of streaming services, I am happy with the extent to which they can influence which music is in their general recommendations.	1	5	3	3	1.26
Q4	For users of streaming services, I am happy with the extent to which they can influence which music is in their personalized playlists.	1	5	4	3.38	1.19

Table 2. Questionnaire responses (1 = 'Strongly disagree', 5 = 'Strongly agree').

ple, this many... and they would do this for all lists. End of story. And all of them would do it." (P8a)

Other than more transparency, giving users more control over their recommendations was also suggested as a way to address unfairness issues. When discussing popularity bias and cold start problems, several participants suggested letting users manually adapt their playlists or general recommendations. Users could then, e.g., indicate they want to receive more songs that they have not listened to before: *"Ideally, I would like to see not one band name I already know. As a small artist, I would appreciate that a lot as well, as it would make chances a little higher you'd maybe be recommended for once."* (P3)

Alternatively, users could indicate their preferred level of adventurousness: *"Maybe it would be nice to make a specific setting for [more music outside of usual taste], 'I feel adventurous' or something like that."* (P6)

As a final insight, we note that most artists desired stronger measures, such as actively making playlists more diverse or balanced. P8a remarked that if users are given the choice, only those who already wish to contribute to a balanced and diverse music landscape would use such functionality: *"[Recommending more diverse music] should just be a standard, [...] because else, it just won't happen. Because that's not the way people are. We are herd animals! We do what we know! And we* also *do what we know if we say: 'I want to discover something new'. [...] It is a choice you think suits you. [...] So if you offer it as a choice, you will keep fishing in the same pond."* (P8a)

P13 also suggests offering more balanced playlists to start with: *"[I would prefer] a playlist which contains a certain number—that it is more balanced."* (P13)

P5 remarked this could be achieved while taking user type (e.g., inclination towards diverse music) into account, and adapting recommendations based on that: *"So it might start with... 30, 33%, and if it is a hit, the percentage becomes higher, and if [the more diverse songs are] skipped, it becomes lower... something like that."* (P5)

Discussion. Artists identified a connection between transparency for users and artist fairness. They indicated that users need insight into how MRS work to make informed decisions matching their fairness needs, which literature suggests could benefit artist fairness [7, 31]. Even though persuasive explanations might increase users' satisfaction with lesser-known artist recommendations [16], we note that such persuasion is not necessarily appreciated [7].

Artists especially emphasized insight into platform fairness goals (confirming results for RS in general [7]), and fairness metrics within playlists. To some extent, playlist channels intended to address fairness are currently offered: e.g., Spotify's *EQUAL* [48] and Deezer's *Women's Impact* [49] initiatives both aim to combat gender disparity, while other curated playlists are dedicated to a certain niche, e.g., Tidal's *Diversity & Tradition: New Black Americana* [50]. However, while dedicating playlists to an underrepresented group addresses the overall fairness issue, each standalone playlist is not necessarily a fair one, as it features only one group. Lastly, we note that collecting data on, or giving users insights into, sensitive attributes such as gender and ethnicity, is a debated topic [51].

Concerning giving users more control, artists noted it would help increase fairness on certain aspects if desired. This corresponds to previous findings [7], where users expressed their wish to adapt personalized RS on diversity aspects. Control over diversity in MRS could also contribute to artist fairness if less popular or (historically) underrepresented artists are recommended as a result [15, 42]. Still, some artists believed that increased control alone would not make a significant impact, as they expected that only users whose listening behavior is already diverse would increase their playlists' diversity.

4.3 RQ3: UI Suggestions

For RQ3, we cover concrete ideas for implementing transparency- and control-enhancing UI functionalities, as brought up in Study 1. These came up when discussing (desired) artist fairness improvements, and integrating those in a manner that users would perceive positively. Ideas focused on influencing either MRS in general, or specific streaming service pages (e.g., playlists).

Artists mentioned some approaches to increase transparency and agency for users simultaneously. Those could be implemented on: (1) a user profile page, where users can modify their general recommendations, and (2) specific playlists so that users could modify recommendations within each playlist. For example, P6 and P11 mentioned sliders to adapt, e.g., how many new artists versus established artists should be recommended, or whether songs should be new to the user: *"Perhaps just a percentage, a slider, saying how many recommended songs you'd know already, and how many you wouldn't know, which you could adjust according to which mood you're in."* (P6)

P7a+b and P11 suggested adding tags or filters so that

users could indicate what they want to be recommended, e.g., only songs from a specific genre or region: *"As if you are logging in from France, for example."* (P7b)

Lastly, P10 mentioned addressing users through a prompt suggesting to increase the listener-artist connection. This could be achieved by proposing that the user visits the profile of artists they often listen to but have not yet looked up. Prompts could also suggest trying something more adventurous: *"Let's say someone is listening to the same things* constantly, *after one week you could also say: 'hey, [user], is it time for something else?' [...] And then you could indicate: 'nah, I don't really feel like it, go back to what I was listening to, let's just play The Beatles and The Rolling Stones again'... Or you like it."* (P10)

Discussion. Our results offer a first insight into transparency- and control-increasing design for MRS, from the artist's perspective. As a whole, artists focused on control in their answers rather than on transparency-increasing design or explanations. They mainly focused on mid-level controls on user profiles, playlists, or through prompts, rather than low- and high-level controls [32]. These responses correspond to previous research (e.g., sliders [32,37–39] and tags [32]) but had not yet been noted in an artist fairness context. From the user perspective, such controls could help influence MRS to better fit the current user goals and mindset (i.e., focused, open, or exploratory) [52]. In Sonboli et al. [7], users emphasize the importance of design practices to promote fair treatment. From the streaming services' side, new functionality by YouTube Music might address the need for more control by allowing users to customize radio channels, e.g., by indicating what percentage of songs should be new to the user [53]. Deezer has also introduced a 'Country Selector' allowing users to switch the 'home country' on which their music and shows recommendations are based [54]. Regarding transparency, Spotify recently introduced an AI-generated 'DJ' feature offering personalized playlists with item-based explanations [55]. Further new functionalities and redesign should be extensively researched and tested to minimize user change aversion [56], and to verify whether they correspond with other stakeholder values.

5. CONCLUSION

5.1 Insights for the MIR Community and Beyond

Our work contributes insights into artists' and music industry professionals' perspectives on MRS transparency towards users. Our results suggest neither stakeholder is positive about the current transparency, despite its importance in MIR systems [18]. *"Transparency can serve to empower artists and listeners to challenge AI systems"* [18]. In the literature, there is a strong agreement that transparency is fundamental for MIR [5, 17, 18] and MRS specifically [10, 36], which is supported by our findings. Our results also show that the transparency towards users is considered insufficient and requires improvement.

Regarding control, artists indicate clearly that they desire increased user control over MRS, deeming the current level insufficient. They argue that the combination of transparency toward users and giving them control will, in turn, help increase fairness for artists, which is a novel and complementary view going beyond existing work. By contrast, music industry professionals are interestingly divided on this matter, for which the cause is yet to be explored.

On a broader level, we learn that there is not necessarily a trade-off between user, item provider, and industry goals (as extensively discussed in multi-stakeholder systems research): indeed, there is some overlap. In our work's context, users, artists, and other music industry professionals essentially want the same (i.e., more transparency and control for users) for similar reasons (i.e., better artist fairness and more recommendation diversity), though our study does not deliver insights concerning industry professionals' reasoning. Hence, it is imperative that MIR involves different stakeholders to understand better what the various actors need and value, and integrates those needs and values in MRS. While trade-offs will keep existing, we need to delve into, and focus on, overlaps and joint goals.

Our work also contributes UI suggestions addressing control and transparency. We note that making only user-facing design changes is insufficient; they should be supported by MIR measures (e.g., data enhancement for retrieval and filtering, fair ranking). We emphasize the significance of combining algorithms and UI research alike.

Concluding, in MIR research, we need to support artists better. Taking a multi-stakeholder approach will accelerate this because some goals and needs are complementary. Essentially, supporting users (in transparency and control) can help artists (in terms of fairness).

5.2 Limitations and Future Work

One constraint of this work is that we aim for exploration with our sample and therefore do not offer an exhaustive, generalizable overview. In future work, both studies could be extended with participants from different cultural, musical, and professional backgrounds to paint a more generalizable overall picture. The perspectives of streaming service providers and other additional stakeholders could also be further addressed. Additionally, as we did not explicitly address transparency and control for users in Study 1, we might have missed views from participants where those topics did not come up. Still, this ensures that responses were spontaneous and unprompted. Lastly, Study 2 mostly contained closed questions that did not allow in-depth analysis, though participants had the possibility to add remarks in the optional free text fields.

A promising future direction is to implement the suggested UI functionalities in a music streaming service, and compare user behavior to that in a system without such functionalities. It would be especially worthwhile to conduct user studies evaluating those functionalities with various stakeholder groups and measure differences in perceived transparency, control, and fairness of such a system. Such a study should cover what RS should deliver if a user indicates not wanting fair recommendations, and how to personalize any fairness-aimed explanations to user needs.

6. ACKNOWLEDGEMENTS

We thank all participating artists and music industry professionals for sharing their invaluable insights. We also express our gratitude to all who supported us in recruiting participants, and especially to Marloes Vredenborg and Isabella Saccardi, both of whom helped conduct Study 2.

7. REFERENCES

[1] IFPI, "Global music report 2023: State of the industry," London, UK, 2023. [Online]. Available: https://www.ifpi.org/wp-content/uploads/2023/03/Global_Music_Report_2023_State_of_the_Industry.pdf

[2] K. Dinnissen and C. Bauer, "Fairness in music recommender systems: a stakeholder-centered mini review," *Frontiers in Big Data*, vol. 5, 2022. [Online]. Available: https://doi.org/10.3389/fdata.2022.913608

[3] M. D. Ekstrand, A. Das, R. Burke, and F. Diaz, "Fairness in information access systems," *Foundations and Trends® in Information Retrieval*, vol. 16, no. 1–2, 2022. [Online]. Available: https://doi.org/10.1561/1500000079

[4] A. Ferraro, X. Serra, and C. Bauer, "Break the loop: Gender imbalance in music recommenders," in *Proceedings of the 2021 Conference on Human Information Interaction and Retrieval*, ser. CHIIR '21. New York, NY, USA: ACM, 2021, pp. 249–254.

[5] C. Bauer, "Allowing for equal opportunities for artists in music recommendation," in *Proceedings of the 1st Workshop on Designing Human-Centric Music Information Research Systems*, ser. wsHCMIR '19, Delft, The Netherlands, 2019, pp. 16–18. [Online]. Available: http://arxiv.org/abs/1911.05395

[6] ——, "Report on the ISMIR 2020 special session: How do we help artists?" *ACM SIGIR Forum*, vol. 54, no. 2, 2020. [Online]. Available: https://doi.org/10.1145/3483382.3483398

[7] N. Sonboli, J. J. Smith, F. Cabral Berenfus, R. Burke, and C. Fiesler, "Fairness and transparency in recommendation: The users' perspective," in *Proceedings of the 29th ACM Conference on User Modeling, Adaptation and Personalization*, ser. UMAP '21. New York, NY, USA: ACM, 2021, pp. 274–279. [Online]. Available: https://doi.org/10.1145/3450613.3456835

[8] K. Robinson, D. Brown, and M. Schedl, "User insights on diversity in music recommendation lists," in *Proceedings of the 21st International Society for Music Information Retrieval Conference*, ser. ISMIR '20. ISMIR, Oct. 2020, pp. 446–453. [Online]. Available: https://doi.org/10.5281/zenodo.4245464

[9] J. H. Lee, L. Pritchard, and C. Hubbles, "Can we listen to it together? Factors influencing reception of music recommendations and post-recommendation behavior," in *Proceedings of the 20th International Society for Music Information Retrieval Conference*, ser. ISMIR '19. ISMIR, Nov. 2019, pp. 663–669. [Online]. Available: https://doi.org/10.5281/zenodo.3527896

[10] A. Ferraro, X. Serra, and C. Bauer, "What is fair? Exploring the artists' perspective on the fairness of music streaming platforms," in *Human-Computer Interaction – INTERACT 2021: 18th IFIP TC 13 International Conference*, ser. INTERACT '21, vol. 12933. Cham, Germany: Springer, 2021, pp. 562–584. [Online]. Available: https://doi.org/10.1007/978-3-030-85616-8_33

[11] I. Siles, A. Ross Arguedas, M. Sancho, and R. Solís-Quesada, "Playing spotify's game: artists' approaches to playlisting in latin america," *Journal of Cultural Economy*, 2022. [Online]. Available: https://doi.org/10.1080/17530350.2022.2058061

[12] B. Friedman, P. H. Kahn, A. Borning, and A. Huldtgren, *Value Sensitive Design and Information Systems*. Dordrecht: Springer Netherlands, 2013, pp. 55–95. [Online]. Available: https://doi.org/10.1007/978-94-007-7844-3_4

[13] K. Dinnissen and C. Bauer, "Amplifying artists' voices: Item provider perspectives on influence and fairness of music streaming platforms," in *Proceedings of the 31st ACM Conference on User Modeling, Adaptation and Personalization*, ser. UMAP '23. New York, NY, USA: ACM, 2023, pp. 238–249. [Online]. Available: https://doi.org/10.1145/3565472.3592960

[14] E. Bugliarello, R. Mehrotra, J. Kirk, and M. Lalmas, "Mostra: A flexible balancing framework to trade-off user, artist and platform objectives for music sequencing," in *Proceedings of the ACM Web Conference 2022*, ser. WWW '22. New York, NY, USA: ACM, 2022, p. 2936–2945. [Online]. Available: https://doi.org/10.1145/3485447.3512014

[15] R. Mehrotra, J. McInerney, H. Bouchard, M. Lalmas, and F. Diaz, "Towards a fair marketplace: Counterfactual evaluation of the trade-off between relevance, fairness & satisfaction in recommendation systems," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, ser. CIKM '18. New York, NY, USA: ACM, 2018, pp. 2243–2251. [Online]. Available: https://doi.org/10.1145/3269206.3272027

[16] S. M. Mousavifar and J. Vassileva, "Investigating the efficacy of persuasive strategies on promoting fair recommendations," in *Persuasive Technology*, ser. PERSUASIVE '22. Cham, Germany: Springer International Publishing, 2022, pp. 120–133. [Online]. Available: https://doi.org/10.1007/978-3-030-98438-0_10

[17] A. Holzapfel, B. Sturm, and M. Coeckelbergh, "Ethical dimensions of music information retrieval

technology," *Transactions of the International Society for Music Information Retrieval*, vol. 1, no. 1, pp. 44–55, Sep 2018. [Online]. Available: https://doi.org/10.5334/tismir.13

[18] 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

[19] P. Tschmuck, *The Economics of Music*, 2nd ed. Newcastle upon Tyne, UK: Agenda Publishing, 2021. [Online]. Available: https://doi.org/10.2307/j.ctv1wgvb9x

[20] C. Bauer and M. Schedl, "On the importance of considering country-specific aspects on the online-market: an example of music recommendation considering country-specific mainstream," in *51st Hawaii International Conference on System Sciences*, ser. HICSS '18, 2018, pp. 3647–3656. [Online]. Available: https://doi.org/10.24251/HICSS.2018.461

[21] A. Epps-Darling, R. Takeo Bouyer, and H. Cramer, "Artist gender representation in music streaming," in *Proceedings of the 21st International Society for Music Information Retrieval Conference*, ser. ISMIR '20. ISMIR, Oct. 2020, pp. 248–254. [Online]. Available: https://doi.org/10.5281/zenodo.4245416

[22] K. Lee and K. Lee, "My head is your tail: Applying link analysis on long-tailed music listening behavior for music recommendation," in *Proceedings of the Fifth ACM Conference on Recommender Systems*, ser. RecSys '11. New York, NY, USA: ACM, 2011, pp. 213–220. [Online]. Available: https://doi.org/10.1145/2043932.2043971

[23] D. Kowald, M. Schedl, and E. Lex, "The unfairness of popularity bias in music recommendation: A reproducibility study," in *Advances in Information Retrieval*, ser. ECIR '20. Cham, Germany: Springer International Publishing, 2020, pp. 35–42. [Online]. Available: https://doi.org/10.1007/978-3-030-45442-5_5

[24] M. Saveski and A. Mantrach, "Item cold-start recommendations: Learning local collective embeddings," in *Proceedings of the 8th ACM Conference on Recommender Systems*, ser. RecSys '14. New York, NY, USA: ACM, 2014, pp. 89–96. [Online]. Available: https://doi.org/10.1145/2645710.2645751

[25] H. Abdollahpouri and S. Essinger, "Multiple stakeholders in music recommender systems," in *1st International Workshop on Value-Aware and Multi-stakeholder Recommendation at RecSys 2017*, ser. VAMS '17, August 2017, pp. 1–3. [Online]. Available: http://arxiv.org/abs/1708.00120

[26] C. Bauer and E. Zangerle, "Leveraging multi-method evaluation for multi-stakeholder settings," in *Proceedings of the 1st Workshop on the Impact of Recommender Systems*, ser. ImpactRS '19, vol. 2462, 2019. [Online]. Available: http://ceur-ws.org/Vol-2462/short3.pdf

[27] P. Knees, A. Ferraro, and M. Hübler, "Bias and feedback loops in music recommendation: Studies on record label impact," in *Proceedings of the 2nd Workshop on Multi-Objective Recommender Systems, co-located with 16th ACM Conference on Recommender Systems (RecSys 2022)*, ser. CEUR Workshop Proceedings, vol. 3268. CEUR-WS.org, 2022. [Online]. Available: https://ceur-ws.org/Vol-3268/paper6.pdf

[28] P. Pu, L. Chen, and R. Hu, "A user-centric evaluation framework for recommender systems," in *Proceedings of the Fifth ACM Conference on Recommender Systems*, ser. RecSys '11. New York, NY, USA: ACM, 2011, p. 157–164. [Online]. Available: https://doi.org/10.1145/2043932.2043962

[29] European Union, "Regulation (eu) 2022/2065 of the european parliament and of the council of 19 october 2022 on a single market for digital services and amending directive 2000/31/ec (Digital Services Act)," 2022. [Online]. Available: https://eur-lex.europa.eu/eli/reg/2022/2065/oj

[30] N. Tintarev and J. Masthoff, "Beyond explaining single item recommendations," in *Recommender Systems Handbook*, F. Ricci, L. Rokach, and B. Shapira, Eds. New York, NY: Springer US, 2022, pp. 711–756. [Online]. Available: https://doi.org/10.1007/978-1-0716-2197-4_19

[31] B. Ferwerda, E. Ingesson, M. Berndl, and M. Schedl, "I don't care how popular you are! investigating popularity bias in music recommendations from a user's perspective," in *Proceedings of the 2023 Conference on Human Information Interaction and Retrieval*, ser. CHIIR '23. New York, NY, USA: ACM, 2023, pp. 357–361. [Online]. Available: https://doi.org/10.1145/3576840.3578287

[32] Y. Jin, N. Tintarev, N. N. Htun, and K. Verbert, "Effects of personal characteristics in control-oriented user interfaces for music recommender systems," *User Modeling and User-Adapted Interaction*, vol. 30, pp. 199–249, 2020. [Online]. Available: https://doi.org/10.1007/s11257-019-09247-2

[33] T. Kulesza, S. Stumpf, M. Burnett, S. Yang, I. Kwan, and W.-K. Wong, "Too much, too little, or just right? ways explanations impact end users' mental models," in *2013 IEEE Symposium on Visual Languages and Human Centric Computing*, ser. VL/HCC '13, 2013, pp. 3–10. [Online]. Available: https://doi.org/10.1109/VLHCC.2013.6645235

[34] P. Kouki, J. Schaffer, J. Pujara, J. O'Donovan, and L. Getoor, "Personalized explanations for hybrid recommender systems," in *Proceedings of the*

24th International Conference on Intelligent User Interfaces, ser. IUI '19. New York, NY, USA: ACM, 2019, pp. 379–390. [Online]. Available: https://doi.org/10.1145/3301275.3302306

[35] G. Zhao, H. Fu, R. Song, T. Sakai, Z. Chen, X. Xie, and X. Qian, "Personalized reason generation for explainable song recommendation," *ACM Transactions on Intelligent Systems and Technology*, vol. 10, no. 4, jul 2019. [Online]. Available: https://doi.org/10.1145/3337967

[36] M. Millecamp, N. N. Htun, C. Conati, and K. Verbert, "To explain or not to explain: The effects of personal characteristics when explaining music recommendations," in *Proceedings of the 24th International Conference on Intelligent User Interfaces*, ser. IUI '19. New York, NY, USA: ACM, 2019, pp. 397–407. [Online]. Available: https://doi.org/10.1145/3301275.3302313

[37] I. Andjelkovic, D. Parra, and J. O'Donovan, "Moodplay: Interactive mood-based music discovery and recommendation," in *Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization*, ser. UMAP '16. New York, NY, USA: ACM, 2016, pp. 275–279. [Online]. Available: https://doi.org/10.1145/2930238.2930280

[38] M. Millecamp, N. N. Htun, Y. Jin, and K. Verbert, "Controlling Spotify recommendations: Effects of personal characteristics on music recommender user interfaces," in *Proceedings of the 26th Conference on User Modeling, Adaptation and Personalization*, ser. UMAP '18. New York, NY, USA: ACM, 2018, pp. 101–109. [Online]. Available: https://doi.org/10.1145/3209219.3209223

[39] Y. Liang and M. C. Willemsen, "Exploring the longitudinal effects of nudging on users' music genre exploration behavior and listening preferences," in *Proceedings of the 16th ACM Conference on Recommender Systems*, ser. RecSys '22. New York, NY, USA: ACM, 2022, pp. 3–13. [Online]. Available: https://doi.org/10.1145/3523227.3546772

[40] F. Sanna Passino, L. Maystre, D. Moor, A. Anderson, and M. Lalmas, "Where to next? a dynamic model of user preferences," in *Proceedings of the Web Conference 2021*, ser. WWW '21. New York, NY, USA: ACM, 2021, pp. 3210–3220. [Online]. Available: https://doi.org/10.1145/3442381.3450028

[41] S. Petridis, N. Daskalova, S. Mennicken, S. F. Way, P. Lamere, and J. Thom, "Tastepaths: Enabling deeper exploration and understanding of personal preferences in recommender systems," in *27th International Conference on Intelligent User Interfaces*, ser. IUI '22. New York, NY, USA: ACM, 2022, p. 120–133. [Online]. Available: https://doi.org/10.1145/3490099.3511156

[42] A. Anderson, L. Maystre, I. Anderson, R. Mehrotra, and M. Lalmas, "Algorithmic effects on the diversity of consumption on Spotify," in *Proceedings of The Web Conference 2020*, ser. WWW '20. New York, NY, USA: ACM, 2020, pp. 2155–2165. [Online]. Available: https://doi.org/10.1145/3366423.3380281

[43] C. Bauer and E. Zangerle, "Information imbalance and responsibility in recommender systems," in *2nd Workshop on Green (Responsible, Ethical and Social) IT and IS—the Corporate Perspective (GRES-IT/IS)*, ser. GRES-IT/IS '18. Vienna, Austria: Department für Informationsverarbeitung und Prozessmanagement, WU Vienna University of Economics and Business, March 2018. [Online]. Available: https://epub.wu.ac.at/7681/

[44] J. W. Morris and D. Powers, "Control, curation and musical experience in streaming music services," *Creative Industries Journal*, vol. 8, no. 2, pp. 106–122, 2015. [Online]. Available: https://doi.org/10.1080/17510694.2015.1090222

[45] G. Guest, A. Bunce, and L. Johnson, "How many interviews are enough?: An experiment with data saturation and variability," *Field Methods*, vol. 18, no. 1, pp. 59–82, 2006. [Online]. Available: https://doi.org/10.1177/1525822X05279903

[46] P. Mayring, *Qualitative content analysis*, 2nd ed. Sage, 2004, vol. 1, pp. 159–176.

[47] K. Dinnissen and C. Bauer, "Questionnaire: Music Industry Professionals' View on Music Streaming Services and Recommender Systems," Jul. 2023. [Online]. Available: https://doi.org/10.5281/zenodo.8121152

[48] "Spotify EQUAL," 2023, accessed: 2023-07-07. [Online]. Available: https://open.spotify.com/genre/equal-page

[49] "Women's Impact," 2023, accessed: 2023-07-07. [Online]. Available: https://www.deezer.com/en/channels/womensvoices

[50] "Diversity & Tradition: New Black Americana," 2023, accessed: 2023-07-07. [Online]. Available: https://tidal.com/browse/playlist/d80fa7b2-a08b-4b49-9de7-65326d5dfe51

[51] M. Bogen, A. Rieke, and S. Ahmed, "Awareness in practice: Tensions in access to sensitive attribute data for antidiscrimination," in *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, ser. FAT* '20. New York, NY, USA: ACM, 2020, pp. 492–500. [Online]. Available: https://doi.org/10.1145/3351095.3372877

[52] C. Hosey, L. Vujović, B. St. Thomas, J. Garcia-Gathright, and J. Thom, "Just give me what i want: How people use and evaluate music search," in *Proceedings of the 2019 CHI Conference on Human*

Factors in Computing Systems, ser. CHI '19. New York, NY, USA: ACM, 2019, p. 1–12. [Online]. Available: https://doi.org/10.1145/3290605.3300529

[53] M. Clark, "YouTube Music will let you make your own custom radio stations," Feb. 2023, accessed: 2023-07-07. [Online]. Available: https://www.theverge.com/2023/2/21/23609228/youtube-music-radio-builder-custom-stations

[54] "Never Miss Home Again With Deezer's Country Selector," Sep. 2020, accessed: 2023-07-07. [Online]. Available: https://www.deezer-blog.com/press/deezer-country-selector/

[55] "Spotify Debuts a New AI DJ, Right in Your Pocket," Feb. 2023, accessed: 2023-07-07. [Online]. Available: https://newsroom.spotify.com/2023-02-22/spotify-debuts-a-new-ai-dj-right-in-your-pocket/

[56] I. Pettersson, C. Fredriksson, R. Dadgar, J. Richardson, L. Shields, and D. McKenzie, "Minimizing change aversion through mixed methods research: A case study of redesigning spotify's your library," in *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems*, ser. CHI EA '23. New York, NY, USA: ACM, 2023. [Online]. Available: https://doi.org/10.1145/3544549.3573875

TOWARDS A NEW INTERFACE FOR MUSIC LISTENING: A USER EXPERIENCE STUDY ON YOUTUBE

Ahyeon Choi Eunsik Shin Haesun Joung Joongseek Lee Kyogu Lee
Department of Intelligence and Information, Seoul National University
{chah0623, esshin, gotjs3841, joonlee8, kglee}@snu.ac.kr

ABSTRACT

In light of the enduring success of music streaming services, it is noteworthy that an increasing number of users are positively gravitating toward YouTube as their preferred platform for listening to music. YouTube differs from typical music streaming services in that they provide a diverse range of music-related videos as well as soundtracks. However, despite the increasing popularity of using YouTube as a platform for music consumption, there is still a lack of comprehensive research on this phenomenon. As independent researchers unaffiliated with YouTube, we conducted semi-structured interviews with 27 users who listen to music through YouTube more than three times a week to investigate its usability and interface satisfaction. Our qualitative analysis found that YouTube has five main meanings for users as a music streaming service: 1) exploring musical diversity, 2) sharing unique playlists, 3) providing visual satisfaction, 4) facilitating user interaction, and 5) allowing free and easy access. We also propose wireframes of a video streaming service for better audio-visual music listening in two stages: search and listening. By these wireframes, we offer practical solutions to enhance user satisfaction with YouTube for music listening. These findings have wider implications beyond YouTube and could inform enhancements in other music streaming services as well.

1. INTRODUCTION

In recent years, the music streaming industry has witnessed a significant surge in popularity, with market leaders such as Spotify, Apple Music, and Amazon Music dominating the market [1]. Alongside this trend, YouTube has solidified its position as a prominent platform for diverse video content, including documentaries, daily vlogs, entertainment shows, and more. As users flocked to YouTube for various types of content, the platform naturally became a hub for music-related videos as well. Users now have easy access to a wide range of music video content on YouTube, contributing to the growing trend of consuming music through video formats [2].

Indeed, YouTube delivers a distinctive multi-sensory experience by showcasing a vast variety of music-related videos such as music videos, live performances, curated playlists with visual artworks, and cover performances, enabling users to enjoy music through a fusion of visual and auditory elements. Despite Spotify's global prominence based on subscribers, YouTube has seen an increasing number of users turning to its platform for music consumption [1, 3]. This trend is evident in regions like South Korea [4, 5] and Latin America [6], where YouTube dominates as a preferred music platform.

Given YouTube's current dominance in music consumption, there's a need for a more comprehensive investigation into this behavior and patterns. Earlier studies have explored YouTube's role as a streaming service [7], compared its usability with Spotify [3], and analyzed music consumption behavior on YouTube [8]. However, the elements contributing to YouTube's rise as a primary music platform and the actual levels of user satisfaction are still not fully understood, indicating a need for further user-focused research.

Thus, this study aims to conduct in-depth interviews with music consumers on YouTube, examining their behavior, comparing the advantages and disadvantages of using YouTube as a music consumption tool with other music streaming services, and reevaluating YouTube's standing as a tool for music consumption. Additionally, we propose a new interface design that enhances the usability of music-related searches and listening. This research was conducted independently by our team, with no financial backing or data provided by YouTube or any associated organization. With our study, we aim to contribute to the ongoing conversation on YouTube's role as a music platform and offer insights into developing an innovative interface that elevates the user's music listening experience.

2. RELATED WORK

In the field of music information retrieval (MIR), research on music streaming services includes studies on improving recommendation algorithms [9–11], understanding user behavior and patterns of use [12–16], and studying user experiences and interfaces [17–21]. These studies aimed to enhance overall user satisfaction and engagement with music streaming services by providing personalized recommendations, improving the user interface, and identify-

© Ahyeon Choi, Eunsik Shin, Haesun Joung, Joongseek Lee, and Kyogu Lee. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Ahyeon Choi, Eunsik Shin, Haesun Joung, Joongseek Lee, and Kyogu Lee, "Towards a New Interface for Music Listening: A User Experience Study on YouTube", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

ing the factors that influenced user behaviors and preferences.

Compared to other music streaming services, research on music consumption through YouTube has only recently gained attention due to the platform's relatively late recognition as a music consumption platform. Early studies on YouTube's music videos have revealed that music is the most consumed content category on YouTube, and researchers have classified the types of YouTube's music content while analyzing their differences [7]. Furthermore, [3] reported that YouTube is used as frequently as Spotify and is perceived as superior to Spotify in terms of its shareability and accessibility.

As YouTube's influence in music consumption grows, recent research has examined three types of online music practices according to the role YouTube plays: default, soundtracking, and complementary platforms [8]. Authors report that one of the main results is that YouTube's music videos are listened to, rather than watched. However, the significance of visual elements in music listening can differ based on the genre or content. Additionally, it is worth mentioning that the participants in the study reported only occasional use of YouTube for music, which may limit the generalizability of the findings to other contexts, such as frequent YouTube users.

Therefore, this study aims to examine the usage behavior of users who use YouTube more than three times a week in everyday situations, report on the characteristics of the subject group, and classify the content used. In addition, we draw out advantages and disadvantages through usability tests to newly consider the role of YouTube as a music-listening tool. Moreover, the study proposes interface improvement measures to fill the research gap on "how to improve the music listening environment through YouTube." Considering the diverse range of devices used to access YouTube, including mobile devices, PCs, tablets, and TVs, we primarily focus on the mobile device, taking into account its widespread usage among participants.

3. METHODS

3.1 Participant

We recruited 27 Seoul National University students (12 males, 15 females) aged 18 or older (mean=23.40, sd=3.13). Our recruitment focused on participants who listen to music on YouTube at least three times a week while excluding those who rely solely on YouTube Music without using YouTube. This approach allowed us to concentrate on the distinct characteristics of consuming music through videos on YouTube, which encompass both visual elements and audio. Participants were compensated with a cash payment of KRW 10,000. Ethics approval was obtained from the Institutional Review Board of SNU.

3.2 Study Design

Informed by previous studies' methodologies and the specific needs of our research, we designed our interview in two stages: a preliminary questionnaire [22, 23], followed by a semi-structured interview [22–24] that includes a brief ice-breaking session [25]. The preliminary questionnaire collects demographic data and music consumption habits of the participants, such as their academic majors, relationship with music, frequency and duration of YouTube use for music, and specific contexts of YouTube music consumption (excluding YouTube Music). Additionally, we also sought information regarding their subscription to YouTube Premium or usage of YouTube Music.

Phase	Requirements
A Verbal Interview	Asking about participants' music listening habits and preferences along with the motivation to use YouTube.
Usability Test	Comparison of YouTube and other music streaming services and feedback on the interface of YouTube for searching and listening to music.
UI proposal	Propose YouTube interface design for music listening freely, and explain yourself.

Table 1. Three steps of semi-structured interview

Following an ice-breaking session, the semi-structured interview proceeded with three main segments (Table 1). First, we explored participants' regular music consumption habits, such as frequency, platform preference, and content preferences. Second, participants were asked to demonstrate the process of searching and listening to music on YouTube, which allowed for a natural exploration of the platform's advantages and disadvantages in comparison to other music streaming services. Third, participants utilized empty interface templates on iPad to design a new interface for music searching and listening, enabling them to customize the screen ratio, functions, buttons, and more. Each interview, lasting roughly 30-40 minutes, was recorded and transcribed using NAVER Clova Note, with participant consent.

3.3 Analysis

We identified the overarching themes and trends of the participants' responses and organized the data accordingly. The data were categorized into the following topics: primary streaming service, weekly listening time, music listening type, preferred music genres or content on YouTube, situations YouTube is used for music listening, reasons for using YouTube as a music consumption tool, music search methods on YouTube, criteria for video selection, advantages and disadvantages of YouTube compared to other services, and a summary of interface proposal sessions.

We generated a list of keywords for the qualitative analysis, which includes the advantages and disadvantages of using YouTube for music listening and user interface proposals from interviewees. To validate our classifications and identify commonalities, we repeated the process of analysis and consensus-building three times among the researchers similar to the analysis process in [22, 26, 27]. Grounded theory [28] and content analysis [29] were also

used as a guide throughout the process of keyword generation. We also referred to previous qualitative studies in the field of MIR [8, 15, 24, 26, 27] to guide our data analysis, as well as to ensure consistency in our reporting and citation practices. Finally, we thoroughly reviewed the keyword lists to extract the main findings of how YouTube is used as a music consumption tool by the participants based on the method of theme analysis [30].

To better understand the participants' interface design proposals, we compared the proposals from the participants and reviewed the summary of the interface designing sessions. From this process, we synthesized useful design implications and arrived at wireframe designs for the music search and listening screens.

Category	Keyword	Freq	Total
Musical Diversity	official soundtrack + α	14	23
	playlist	9	
Convenience	familiarity	4	15
	accessibility	4	
	subscription fee	4	
	Customizing	3	
User Interaction	recommendation	10	13
	comments	3	
Visual Contents	thumbnail	6	10
	video	4	
etc.	etc.	1	1

Table 2. Pros. keywords of usability test

4. RESULT

4.1 Behavior and Characteristics of Music Consumption on YouTube

As the current study investigates interview data from a sample of 27 users, it is important to take into account the unique characteristics of this group. Therefore, information concerning the participants' music consumption behaviors and preferences was gathered through preliminary surveys and interviews. The results showed that participants typically used YouTube about five times per week (mean = 4.89, sd = 1.93), for a total of approximately five hours (mean = 5.35, sd = 3.76), to listen to music while engaging in various activities, such as studying, relaxing, commuting, and exercising. No one specialized in music. The majority of participants used the free version of YouTube and did not subscribe to YouTube Premium. Additionally, some participants supplemented their music listening with other platforms such as YouTube Music, Melon, Spotify, and Genie.

Participants enjoyed a diverse range of music genres on YouTube. The top five genres mentioned most frequently were OST (original soundtrack of movies or dramas, 13 times), pop (12 times), K-pop (11 times), classical music (7 times), and indie music (7 times). Other genres mentioned in order of frequency include J-pop, ballads, old-fashioned music (mid-20th-century Korean pop and ballads), jazz, rock, band music (with live instrumentation and elements of rock, pop, and indie), new age, hip-hop, EDM, and R&B. Music content can be broadly divided into three categories: 1) Official music content such as music videos, 2) Live music content such as performances, concerts, festivals, and 3) User-generated content such as playlists and cover videos. In terms of frequency of mention, the order was 3-2-1 (27 times, 25 times, 8 times) respectively.

4.2 Advantages of using YouTube for music listening

Alongside our anticipation that YouTube serves as an audiovisual music listening tool, we found that YouTube possesses various strengths compared to other streaming services (Table 2). Musical diversity was the most frequently mentioned category, with two main points: the availability of non-official music in addition to official releases, and the diversity of playlist content compared to other streaming services.

With streaming services, I can only listen to official releases, ***but with YouTube, I can listen to not only official releases but covers and other user-generated content.*** (P11)

Unlike other services, ***YouTube's diverse playlists prevent repetitive listening by offering a wide range of songs within similar genres.*** (P26)

Also, convenience was mentioned as an advantage, with familiarity, accessibility, no subscription fee, and user customization.

I use it because I'm used to it. I've used Melon and YouTube Music before, ***but I settled with YouTube because it was more convenient.*** (P12)

Since YouTube is free, there's no need to pay for other services. (P8)

As for user interaction, most users mentioned the recommendation algorithm itself and the ability to view other users' opinions through comments.

The recommendation algorithm is good. *I often find great new songs through it.* (P17)

It's good to be able to see other people's opinions and sympathize by reading comments. (P4)

Lastly, the visual content of thumbnails and videos was mentioned as an advantage.

I can use both sight and sound when listening to music with videos. (P4)

When I play playlists with thumbnails, like at a housewarming party, it adds to the atmosphere, and it's good for interior purposes too. (P6)

While we initially expected the inclusion of visual elements to be a significant advantage of YouTube, the participants' usage patterns proved more diverse. Some appreciated the visual components, while others turned to YouTube strictly for audio during activities like work or sleep (P2, P5, P22, P23, P24). These observations align with prior research [8], showing the varied ways users utilize YouTube for music. Although some mentioned listening to audio with the screen off (P7, P23, P24), we ex-

Category	Keyword	Freq	Total
User Interaction	comments	12	21
	recommendation	9	
Manipulation	button / tap	10	17
	display ratio	7	
Playlist	playlist contents	4	10
	making playlist	4	
	mix playlist	2	
Section Search	timestamp	6	9
	playback bar	3	
Lack of Info.	song information	5	8
	log information	3	
Underutilization	replay	5	8
	volume control	3	
Contents Quality	sound quality	5	6
	video quality	1	
etc.	(video) data size	4	6
	etc.	2	

Table 3. Cons. keywords of usability test

cluded this aspect from our analysis as it's a feature exclusive to YouTube Premium subscribers.

4.3 Disadvantages of using YouTube for music listening

The inconveniences and disadvantages of listening to music on YouTube were categorized into seven major themes (Table 3). The most frequently mentioned inconvenience was related to user interaction, with many complaints about the inconvenience of filtering the desired information while exploring recommended videos and comments.

In other music streaming services, genre separation is clearly done, ***but YouTube recommends based on the videos you watch, so there is a tendency to lean towards a specific genre.*** (P26)

When watching music videos and reading comments, it's hard to find South Korean users' reactions when most comments are in foreign languages. (P11)

The second most frequently mentioned disadvantage was related to screen manipulation, such as fixed thumbnails, the ratio of videos, and accidental button presses.

It would be nice if I could reduce the screen ratio. *I want to watch the small screen when exercising or doing other things.* (P8)

There are cases where I accidentally press the Shorts button and the music stops. (P22)

Regarding playlists, users complained about not having timestamps for individual songs, the content of playlists made by others, the process of creating playlists themselves, and the mixes provided by YouTube.

It's inconvenient to switch to another song ***if there is no timestamp in the playlist.*** (P18)

Since playlists are made by others, there are few cases where all songs suit my taste, *and there are mediocre songs in between.* (P18)

It's inconvenient to save songs one by one in my library. *It feels slow every time I press the save button, and it is a hassle to press the button several times to save.* (P27)

Lastly, some users mentioned the lack of information about album or song information and lyrics, as well as the lack of log information about previously watched videos as a disadvantage.

It's hard to find album or song information, *and it's frustrating not knowing the information of the concert I am watching.* (P7)

When I use the autoplay function, it is hard to find which song I thought I liked. (P17)

Despite the existence of autoplay and volume control features on YouTube, user complaints arose from a lack of information about these functions. Some users viewed them as drawbacks, unaware of their existence or location. Specifically, enabling autoplay requires navigating to the settings, while fine-tuning volume necessitates physical device button use. This complexity may have heightened user frustration and dissatisfaction.

I wish there is a autoplay button. (P12)

I want to make minute adjustments, ***but even if I increase the volume level by just one, the volume suddenly becomes too loud.*** (P16)

The quality of the content is related to the audio or video quality. Some responses showed low reliability in audio quality when used for music listening.

There are cases where the ***sound quality is poor in content uploaded by individual users.*** (P12)

Aside from that, there were four mentions of concerns about mobile data usage due to large video data size (P2, P9, P11, P25), one mention of discomfort with provocative titles (P23) and experiencing an error when randomly playing saved videos (P14). There were nine mentions related to ads or background playback (P2, P5, P6, P8, P9, P12, P17, P19, P25), but these were excluded from the analysis since they can be resolved with a YouTube premium subscription.

4.4 User Feedback for Interface Improvements

We analyzed users' explanations and drawings of the searching and listening screens, categorizing their demands for interface improvement into three categories: addition, modification, and deletion. These categories, along with relevant quotes, provide specific descriptions of users' interface improvement suggestions.

The first is to request the addition of new information or functions that are currently absent on YouTube, such as a new button or tab, new sorting and filtering criteria, or more information about contents and songs.

It would be great if there was a detailed search button under the search bar*, where you could search by year, album, composer, etc.* (P7)

It would be nice if I can choose options between "all videos recommended" and "music-related recommendations" in the recommendation section. (P9)

The second is to modify the existing functions or configuration of YouTube to increase operability and efficiency when searching and listening to music, such as changing the ratio of various spaces on the interface or changing the positions of existing buttons and information.

It would be great ***if the thumbnail (album cover) could be smaller,*** *and the title, artist, etc. could be displayed next to it.* (P12)

It would be nice to adjust the ratio of the comment box and recommended videos *so that you can view them together.* (P4)

Lastly, there were cases where demands were made to remove things from the existing YouTube interface that are not directly related to music searching or listening.

We don't need the buttons for uploading videos on the bottom menu bar. *It would be great if we could freely configure this menu bar.* (P15)

If we could hide the buttons we don't use often and press the detailed button to show them, it would be neat. (P19)

5. FINDINGS AND DICSUSSION

5.1 Role of YouTube as a music streaming service

Users desire an improved interface for YouTube to maximize its potential as a music consumption tool. We have identified five key roles that YouTube plays in music listening, and based on this, we propose design implications to enhance the user experience.

5.1.1 Exploring musical diversity

YouTube offers users a wide variety of musical genres, artists, and songs to discover and explore, including rare or unreleased music not found on other streaming services. Users can also enjoy various versions of the same song through covers or live performances by different artists. **Design Implication:** To improve search efficiency, music content should be categorized by genre, artist, and mood, and album information such as lyrics should be provided to reduce the need to search for information on other platforms.

5.1.2 Sharing unique playlists

YouTube creators can create and share playlists, simplifying the search process and enabling them to select playlists based on keywords like mood or activity (e.g. warm spring day, driving playlist). **Design Implication:** Playlists should provide song and timeline information, and allow users to switch to the next song with a button. Allowing users to customize songs within the playlist, such as adding or removing them, and saving these changes, would enhance the playlist's functionality.

5.1.3 Providing visual satisfaction

By offering sensory satisfaction beyond just videos, YouTube's visual content enhances the music listening experience. Users appreciate being able to observe the musicians' expressions, gestures, and style, and sometimes even watch music videos solely for visual gratification like repetitive animations or thumbnail images paired with the music. **Design Implication:** The screen size and ratio of the video should be customizable based on the content's characteristics and users' listening environment. For example, users would like the option to decrease the video screen size in public settings or enlarge it to focus on a particular idol member or musician's finger movements.

5.1.4 Facilitating user interaction

YouTube's likes, dislikes, subscriptions, and comments features enable users to interact with the platform and foster a sense of community, resulting in a more engaging music listening experience. Additionally, the recommendation algorithm lets users explore new content and see how others react to music, which is a key motivation for users to use YouTube. **Design Implication:** Users should be able to sort and filter recommendations and comments based on various criteria, such as timeline, keyword, lyrics or the most frequently mentioned, to expand YouTube's social function. Pinning specific comments that users like or refer to frequently could also reduce search time.

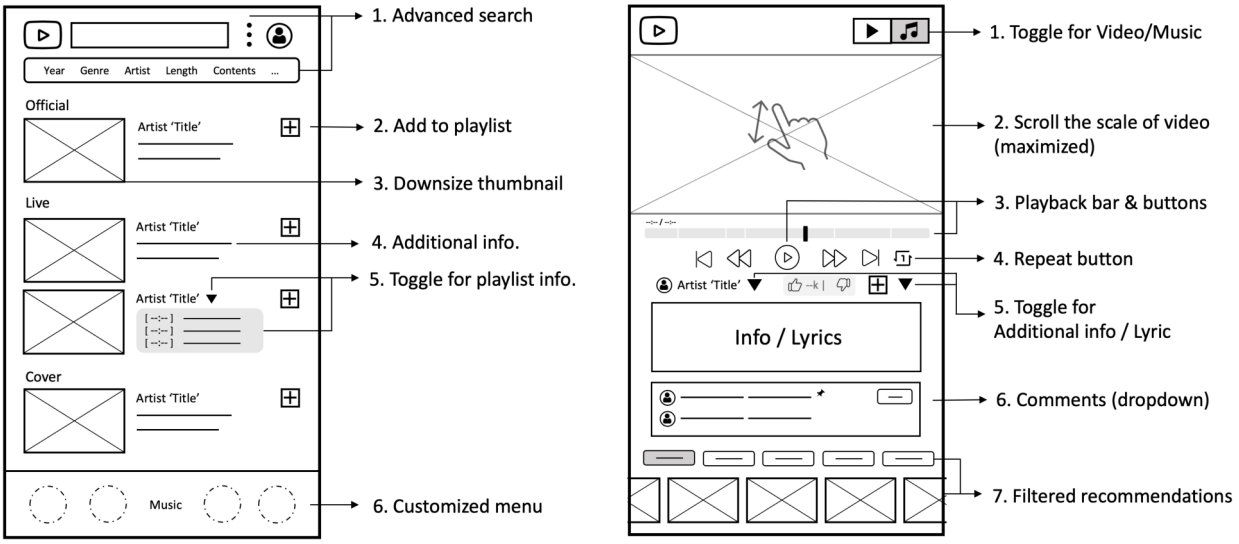
5.1.5 Allowing free and easy access

YouTube's accessibility, cost-effectiveness, and cross-device compatibility make it a convenient option for users to listen to music in various situations. This versatility has led some users to cease subscribing to other streaming services. Primarily, the appeal lies in the free access to a diverse library of music videos, live performances, covers, and user-generated content, resonating with users disinclined to pay for music subscriptions. **Design Implication:** While device-specific interfaces are important, consistent usability is crucial to prevent user confusion or inconvenience.

It is crucial to acknowledge that while some of the proposed features (e.g., album and artist filters, lyrics, smaller screen mode) have already been implemented in the YouTube Music App, users still rely on YouTube to access a diverse range of music videos that are not available on the YouTube Music App. Therefore, our design implications hold the potential to differentiate YouTube from YouTube Music by catering to the experience of video streaming alongside music consumption.

5.2 UI for Audio-Visual Music Streaming Platform

Taking into account the role of YouTube as a music streaming service, the needs of its users, and the interface designs of typical music streaming services, we have developed an ideal wireframe for an audio-visual music listening platform. It consists of two stages: (a) searching and (b) listening screens (Figure 1).



(a) Searching (b) Listening

Figure 1. A Wireframe of youtube UI for music listening

(a) Searching To display diverse music content tailored to users' interests, we added 1) advanced search functionality to the top keyword search bar, allowing users to filter by era, genre, artist, and other details. Additionally, we added 2) a button to easily add multiple videos to a user's playlist, and 3) reduced the thumbnail size to show more videos on one screen. Next to the thumbnail, we included 4) information about the songs in the video, and if the video is a playlist, we added 5) a timeline and information about the included songs. Finally, we made 6) the bottom menu buttons customizable, allowing users to remove buttons when they feel unnecessary and create their own menu.

(b) Listening While maintaining the current structure of the interface, we adjusted the layout and added new features to enhance the music listening experience. 1) Adding a toggle button that allows users to exchange between video watching and music listening. Users can use their fingers to 2) zoom in or out of the video to adjust its size. Previously, users had to click the video to access playback and skip buttons, but we located 3) the playback bar and related functions at the bottom of the video. We also made the 4) repeat button more visible. We added 5) a toggle button to expand or shorten album information or lyrics, and made 6) comments expandable in a similar manner, with a function for users to pin comments they want to keep visible. We added 7)a filtered recommendation feature to suggest reduced-size videos based on specific user-selected filters. This allows for easier exploration of related content through horizontal scrolling.

The findings of this research hold potential for application across a variety of streaming services. Features such as advanced search functions, customizable menus, and enhanced playlist capabilities can improve user engagement and satisfaction. Effective presentation of music-related information enriches the listening experience, while additional functionalities such as video zoom or comment pinning foster a personalized user experience. These findings can significantly benefit YouTube, as well as aid other music streaming platforms like Spotify, Apple Music, and Amazon Music, and video streaming services including music videos, like Bilibili and Vimeo, in optimizing their interfaces according to their unique characteristics and users' needs.

6. CONCLUSION

This study explored the music listening behaviors of YouTube users and analyzed the advantages and disadvantages of YouTube as a music streaming service. We proposed new interface wireframes to improve usability and re-examined YouTube's role as a tool for music listening. Undoubtedly, there are constraints in actualizing the proposed interface fully on YouTube. Nevertheless, some suggestions on improving visual satisfaction, comment exploration, and toggle button to exchange the interface between video watching and music listening mode could be considered in designing the overall interface of video streaming platforms.

Our study has limitations depending on the small sample size of Korean users, and it is essential to consider several important factors. Firstly, our interface design primarily focused on mobile environments, which may limit its direct applicability to other devices like PCs and TVs. Secondly, the relatively narrow age range and educational levels of our participants may affect the generalizability of our findings. Thirdly, the absence of comparative studies on similar video platforms and services hinders our understanding of YouTube's performance as a video streaming service. However, these limitations present opportunities for future research to explore and address the diverse needs of users across different devices, demographics, and video services. Overall, our study provides valuable insights and paves the way for further advancements in user-centered design for music streaming services.

7. REFERENCES

[1] "Business of apps, music app report 2023," https://www.businessofapps.com/data/music-app-report/, accessed: 2023-04-14.

[2] C. Vernallis, *Unruly media: YouTube, music video, and the new digital cinema*. Oxford University Press, 2013.

[3] L. A. Liikkanen and P. Åman, "Shuffling services: Current trends in interacting with digital music," *Interacting with Computers*, vol. 28, no. 3, pp. 352–371, 2016.

[4] "Statista, most frequently used music streaming or download services south korea 2022," https://zrr.kr/1xyb, accessed: 2023-04-14.

[5] "Korea create content agency (kocca), survey of music users 2022," https://zrr.kr/BHhX, accessed: 2023-04-14.

[6] "Statista, top music streaming platforms in selected countries latin america 2021," https://www.statista.com/statistics/1291284/music-streaming-platforms-latin-america/, accessed: 2023-04-14.

[7] L. A. Liikkanen and A. Salovaara, "Music on youtube: User engagement with traditional, user-appropriated and derivative videos," *Computers in human behavior*, vol. 50, pp. 108–124, 2015.

[8] J.-S. Beuscart, S. Coavoux, and J.-B. Garrocq, "Listening to music videos on youtube. digital consumption practices and the environmental impact of streaming," *Journal of Consumer Culture*, p. 14695405221133266, 2022.

[9] S. Freeman, M. Gibbs, and B. Nansen, "'don't mess with my algorithm': Exploring the relationship between listeners and automated curation and recommendation on music streaming services," *First Monday*, vol. 27, no. 1, 2022.

[10] Z. Cheng, J. Shen, L. Zhu, M. S. Kankanhalli, and L. Nie, "Exploiting music play sequence for music recommendation." in *IJCAI*, vol. 17, 2017, pp. 3654–3660.

[11] D. Sánchez-Moreno, A. B. Gil González, M. D. Muñoz Vicente, V. López Batista, and M. N. Moreno-García, "Recommendation of songs in music streaming services: dealing with sparsity and gray sheep problems," in *Trends in Cyber-Physical Multi-Agent Systems. The PAAMS Collection-15th International Conference, PAAMS 2017 15*. Springer, 2018, pp. 206–213.

[12] B. Zhang, G. Kreitz, M. Isaksson, J. Ubillos, G. Urdaneta, J. A. Pouwelse, and D. Epema, "Understanding user behavior in spotify," in *2013 Proceedings IEEE INFOCOM*. IEEE, 2013, pp. 220–224.

[13] J. Fuller, L. Hubener, Y.-S. Kim, and J. H. Lee, "Elucidating user behavior in music services through persona and gender." in *ISMIR*, 2016, pp. 626–632.

[14] N. Montecchio, P. Roy, and F. Pachet, "The skipping behavior of users of music streaming services and its relation to musical structure," *Plos one*, vol. 15, no. 9, p. e0239418, 2020.

[15] J. H. Lee and R. Price, "Understanding users of commercial music services through personas: Design implications." in *ISMIR*, 2015, pp. 476–482.

[16] M. L. Barata and P. S. Coelho, "Music streaming services: understanding the drivers of customer purchase and intention to recommend," *Heliyon*, vol. 7, no. 8, p. e07783, 2021.

[17] B. Ferwerda, E. Yang, M. Schedl, and M. Tkalcic, "Personality and taxonomy preferences, and the influence of category choice on the user experience for music streaming services," *Multimedia tools and applications*, vol. 78, pp. 20 157–20 190, 2019.

[18] M. Mäntymäki and A. Islam, "Gratifications from using freemium music streaming services: Differences between basic and premium users," 2015.

[19] A. N. Hagen, "The playlist experience: Personal playlists in music streaming services," *Popular Music and Society*, vol. 38, no. 5, pp. 625–645, 2015.

[20] J. H. Lee and R. Price, "User experience with commercial music services: An empirical exploration," *Journal of the Association for Information Science and Technology*, vol. 67, no. 4, pp. 800–811, 2016.

[21] D. M. Weigl and C. Guastavino, "User studies in the music information retrieval literature." in *ISMIR*, 2011, pp. 335–340.

[22] J. M. Morse, "Approaches to qualitative-quantitative methodological triangulation," *Nursing research*, vol. 40, no. 2, pp. 120–123, 1991.

[23] S. Bunian, K. Li, C. Jemmali, C. Harteveld, Y. Fu, and M. S. Seif El-Nasr, "Vins: Visual search for mobile user interface design," in *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, 2021, pp. 1–14.

[24] J. H. Lee and A. T. Nguyen, "How music fans shape commercial music services: A case study of bts and army." in *ISMIR*, 2020, pp. 837–845.

[25] L. Kindbom, "How does interface design and recommendation system in video streaming services affect user experience?: A study on netflix ui design and recommendation system and how it shapes the choices young adults between the ages 18 and 26 make." 2022.

[26] J. H. Lee, A. Bhattacharya, R. Antony, N. K. Santero, and A. Le, "â?? finding homeâ?: Understanding how music supports listenersâ?? mental health through a case study of bts." in *ISMIR*, 2021, pp. 358–365.

[27] J. H. Lee, B. Bare, and G. Meek, "How similar is too similar?: Exploring users' perceptions of similarity in playlist evaluation." in *ISMIR*, vol. 11, 2011, pp. 109–114.

[28] B. G. Glaser and A. L. Strauss, *Discovery of grounded theory: Strategies for qualitative research*. Routledge, 2017.

[29] A. Luo, "What is content analysis and how can you use it in your research," *Retrived from https://www. scribbr. com/methodology/content-analysis/on November*, vol. 4, p. 2020, 2019.

[30] V. Clarke, V. Braun, and N. Hayfield, "Thematic analysis," *Qualitative psychology: A practical guide to research methods*, vol. 3, pp. 222–248, 2015.

FILOBASS: A DATASET AND CORPUS BASED STUDY OF JAZZ BASSLINES

Xavier Riley
Queen Mary University of London
j.x.riley@qmul.ac.uk

Simon Dixon
Queen Mary University of London
Centre for Digital Music

ABSTRACT

We present FiloBass: a novel corpus of music scores and annotations which focuses on the important but often overlooked role of the double bass in jazz accompaniment. Inspired by recent work that sheds light on the role of the soloist, we offer a collection of 48 manually verified transcriptions of professional jazz bassists, comprising over 50,000 note events, which are based on the backing tracks used in the FiloSax dataset. For each recording we provide audio stems, scores, performance-aligned MIDI and associated metadata for beats, downbeats, chord symbols and markers for musical form.

We then use FiloBass to enrich our understanding of jazz bass lines, by conducting a corpus-based musical analysis with a contrastive study of existing instructional methods. Together with the original FiloSax dataset, our work represents a significant step toward a fully annotated performance dataset for a jazz quartet setting. By illuminating the critical role of the bass in jazz, this work contributes to a more nuanced and comprehensive understanding of the genre.

1. INTRODUCTION

The role of the double bass (also known as the string bass or upright bass) in jazz is nearly ubiquitous as a time keeper, outliner of harmony and as an occasional soloist. A key function is to play "walking bass", where the harmony of the song is outlined by playing chord tones on strong beats and linking them with arpeggio, scale or chromatic movements on the remaining beats in the bar. This style has emerged as a way to provide a rhythmic and harmonic foundation to support a soloist. We believe that the harmonic techniques that performers use to outline chord changes could provide important information for enhanced understanding of jazz from an MIR perspective, e.g. for generative models. Due to the relatively simple rhythmic vocabulary, this style lends itself to algorithmic approaches which reduce the problem to beatwise pitch predictions, as discussed in Section 2. However, we recognize that this is a simplified view of bass performance, as bass lines also contain rhythmic subtleties and other nuances which serve to increase the interest and texture of the music over time.

The FiloSax dataset [1] addressed a need for high quality annotations [2] to enable downstream tasks like automatic music transcription, score layout and performance analysis. Building on this, we address the need for similarly high quality data relating to the double bass as used in jazz, by turning our attention to the backing tracks used to create that dataset. The backing tracks are taken from the Aebersold series [1] and include performances by professional musicians.

Given the high quality of the bass playing on these tracks, we provide fine-grained annotations to allow for detailed stylistic and harmonic analysis. We believe that this represents the first large scale dataset to include detailed performance timing for jazz bass, which in turn should allow for more realistic generative modelling applications and better results for automatic transcription models. The transcriptions have been carried out using a semi-automatic pipeline which we describe in Section 3. Each note was checked manually and additionally proof-read by a professional jazz bassist. We also publish the extracted audio stems together with the transcriptions using the SoundSlice platform [2] to allow for easy browsing and evaluation [3]. Audio, MIDI and MusicXML artefacts along with the code to produce our analysis are available to download via the same site.

2. RELATED WORK

Despite the important role of bass in the jazz genre, study of this subject has often relied on fully manual transcriptions which are extremely labour intensive to produce (see [3] for an example). To address the need for data on a larger scale, important work was led by Abeßer et al. into automatic transcription of bass lines in a jazz context [4–6]. One of the motivations for their work was the idea that accurate bass transcriptions may be used to derive information about the harmony of a song, which in turn could aid with the task of automatic chord estimation. This resulted in 41 automatic bass transcriptions (with manual verification) as part of the Weimar Jazz Dataset [7] (WJD). These are beat-wise pitch transcriptions, meaning that they are only a partial annotation of the performance, omitting in-

© X. Riley and S. Dixon. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** X. Riley and S. Dixon, "FiloBass: A Dataset and Corpus Based Study of Jazz Basslines", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

[1] http://jazzbooks.com/jazz/JBIO
[2] https://www.soundslice.com/
[3] https://aim-qmul.github.io/FiloBass/

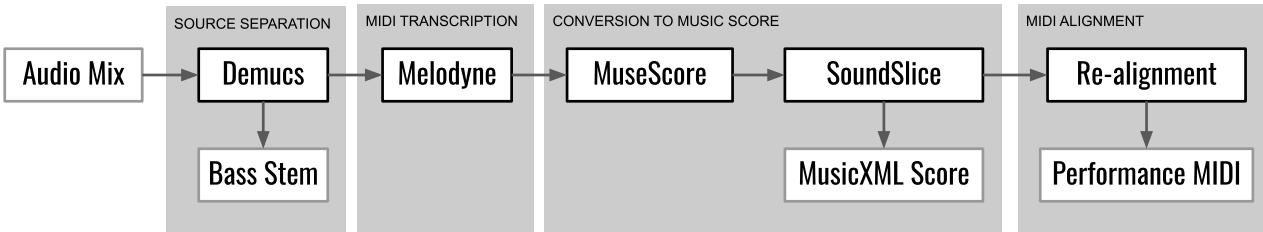


Figure 1. Flow diagram describing the main stages of the proposed method.

formation about rhythmic details, which may limit the use of this dataset in some downstream tasks such as performance analysis or generative modelling. Recent releases of the WJD dataset have included a further 415 fully automatic transcriptions of the bass notes for each beat.

The RWC-Jazz database [8] (a subset of the widely cited RWC dataset) provides audio and aligned MIDI annotations for 5 pieces, which have multiple recordings across a number of different instrument groupings which include bass. Bass is included on 37 recorded tracks which total around 3 hours of audio, however the audio is synthesised from samples of isolated notes and is mixed rather than provided as individual audio stems. This allows for accurate alignment at the expense of some realism in terms of articulation and dynamic range.

Formal research into walking bass has also focused on rule-based generation for modelling bass performances [9]. By incorporating rules described in instructional materials for learning jazz bass, the authors were able to construct a hidden Markov model (HMM) which produced musically relevant results according to subjective listening tests. The authors mention a lack of training data for this task and also note that they were unable to model anything beyond beat-wise pitch estimation.

Outside of the jazz genre, Araz [10] describes a pipeline for transcribing bass lines from electronic music. This approach relies on source separation to extract a bass stem before transcribing it to quantised MIDI. This approach assumes that the music is recorded at a fixed tempo, which is usually the case for electronic genres however this is not usually the case for jazz performances. The MedleyDB [11] dataset provides a large corpus of multitrack audio recordings. Of these, 71 have been and annotated and resynthesised using the process described in [12] to produce the MDB-bass-synth dataset. This dataset is primarily aimed at training and evaluating framewise pitch estimation (f0) methods. We also note the IDMT-SMT-Bass dataset [13] which provides individual recordings of each note on an electric bass with a variety of playing techniques. This may be a good basis for a synthetic dataset to approach similar tasks. A summary of the available datasets is shown in Table 1.

3. METHODOLOGY

We now describe the process used to create the dataset which is summarised in Figure 1. We would like to emphasise that the work was carried out by the main author, a semi-professional bassist, and later checked and verified by another professional jazz bassist. Despite the use of automatic methods, every note was checked manually at least twice as a result. While this process was expensive in terms of time spent, the resulting increase in accuracy will provide a solid foundation for future methods.

3.1 Audio Recordings

All of the 48 backing tracks in this dataset are recorded in a standard format using professional jazz musicians. Details of the performers are shown in Table 2. They feature a jazz trio (piano, bass and drums) with bass panned to the left, drums panned centrally and piano panned to the right. This allows for convenient separation of bass and drums by using a single channel of audio. We are able to further isolate this single channel to obtain a bass stem using the Demucs source separation tool [14]. The producers of these tracks (Aebersold) have a catalogue of over 1300 tracks recorded in a similar fashion, which means that this approach could be applied to additional tracks in the future.

3.2 Transcription

For the initial transcription of performance MIDI, we opted to use the commercial program Melodyne [4], specifically their "Melodic" detection algorithm. This is more typically used for editing vocal performances, however the pitch tracking and note segmentation proved to be broadly accurate for the separated bass stems. The program also offers a convenient interface to edit onsets and pitches manually in cases where the automatic analysis was judged to be incorrect. Each of the 48 scores were loaded into Melodyne and manually corrected where necessary.

To produce a score from the performance MIDI we employed a multi-step process. The first step was to import the existing downbeat annotations from the FiloSax dataset into Melodyne. We then used the "Make tempo constant" feature of Melodyne to produce a new file in which variations in the tempo were removed and the note positions rescaled accordingly. For those without access to Melodyne, we note that a similar result could be achieved using the `adjust_times` function from the PrettyMIDI library [15].

[4] https://www.celemony.com/en/melodyne/what-is-melodyne

Name	Annotation Method	Audio sources	Sync. level	Track count	Duration (s)	Note count	Additional Metadata	Scores
WJD Bass	Automated + Manual	Audio mix	Beat	41	1851	5000	Downbeat, Chord	No
WJD v2.2	Automated	Audio mix	Beat	456	49010	122540	Downbeat, Chord	No
MDB-bass-synth	Automated	Audio mix, Audio stems	Frame	71	14393	N/A	None	No
RWC-Jazz	Manual	Audio mix	Note	37	10878	19183	Downbeat, Chord	No
IDMT-SMT-Bass	N/A	Individual notes	N/A		12960	4300	None	No
FiloBass (ours)	Automated + Manual	Audio mix, Bass stem	Note	48	17880	53646	Downbeat, Chord	Yes

Table 1. Comparison of existing bass datasets

Name	Track count	Note count	Born
Christian Doky	1	1401	1969
Dennis Irwin	1	1321	1951
John Goldsby	3	2564	1958
Lynn Seaton	1	1278	1957
Michael Moore	1	753	1945
Ray Drummond	2	2181	1946
Ron Carter	5	5885	1937
Rufus Reid	14	15280	1944
Steve Gilmore	10	12323	1943
Todd Coolman	3	3952	1954
Tyrone Wheeler	6	5474	1965
Wayne Dockery	1	1050	1941

Table 2. Details for each bassist in the dataset

From this constant tempo version, we export a MIDI file from Melodyne and then import this into MuseScore 3 [5] using their MIDI import procedure. This was found to work better when the tempo was made constant first. This yields a score representation, however the variations in timing can produce non-idiomatic representations in the score which need to be corrected. This was done by exporting MusicXML and performing the final corrections using the SoundSlice platform, which allowed the transcription to be edited with reference to the synchronized audio from the original bass stem. Chord annotations are then copied from the FiloSax metadata and all 48 scores were checked by a professional jazz bassist to ensure accuracy and readability.

Finally, we used the alignment method proposed by Nakamura et al. [16] to realign the final score representation to the original MIDI performance data. This step is necessary to obtain a 1-to-1 correspondence in note annotations between score and performance MIDI. However, after working with these annotations we found that the timing information in the performance MIDI produced by Melodyne was not of sufficiently high quality. This resulted in issues when evaluating automatic transcription methods (see 5). To improve the alignment quality further, we align the MIDI to the model activations of a pre-trained guitar transcription model following the work of Maman and Bermano [17]. The realigned MIDI outputs are included in the final dataset.

[5] https://musescore.org/en

3.3 Repeated Passages

During the construction of the original FiloSax dataset, one of the objectives was to capture a consistent amount of saxophone data for each track. Since the original backing tracks varied in length, the authors edited the original backing tracks to repeat certain sections (usually complete choruses) in order to meet their criteria. This impacts the production of this dataset in that some passages are repeated exactly, however they were transcribed by treating them as a complete performance. This may lead to slight variations in how the rhythmic figures are notated which may be an issue for certain downstream tasks, for example introducing a bias in generative models. We recognise this and will provide instructions on how to remove the repeated sections if desired. Otherwise we provide transcriptions for each track in its entirety to allow for easy alignment with the existing FiloSax data.

3.4 Double Stops, Grace Notes and Ghost Notes

The source material used for this dataset is predominantly monophonic in nature, however the performers do make use of double stops (polyphony) in some places. We have transcribed these in the score and alignments but we also provide a monophonic version of the dataset with a view to ease of use in downstream tasks. The use of effects such as grace notes (extremely short notes) or ghost notes (where the string is partially or fully dampened to produce a percussive sound) is prevalent throughout the dataset and these can be viewed as an important aspect of the style. A guiding principle for producing the score representation is that they are readable by a sufficiently experienced bassist. With this in mind, we have notated ghost notes where these can be clearly heard on the recording however in cases where these effects were judged to be subtle or fleeting we have omitted them. We understand that this approach could be seen as subjective but we did so to prioritise the goal of making a readable and idiomatic score output over a completely consistent yet less readable score.

3.5 "Common Practice" versus Real Performance

The backing tracks used to create this dataset were originally conceived as practice aids for instrumental soloists. As such, the performances on these tracks could be viewed as a sort of "common practice" of jazz accompaniment. The performers focus on outlining chord changes and rhythms clearly to allow the soloist to focus on their role. This aspect of the data makes it a valuable example for studying how these accompaniments are constructed.

However, they may not be entirely representative of performances from live or studio recordings, as musicians may be more inclined to take musical risks in those settings. For this reason, the figures that we derive in our later analysis might not be fully representative of live or studio performance. A comparison is a potential area for exploration in future work.

3.6 Dataset Contents and Distribution

The final dataset comprises 48 tracks with contents as follows: Melodyne project files, audio mixes, isolated bass stems (from source separation software), performance-aligned MIDI with velocity information, and music scores in MusicXML format. We also include metadata which was compiled as part of the FiloSax dataset which includes timings for chords, sections, beats and downbeats.

As discussed in [1], the backing tracks themselves are subject to copyright restrictions so we are unable to release these. However, we provide instructions on how to obtain the files from the original provider. All other assets (including the source-separated stems) will be made freely available to researchers.

4. ANALYSIS

We now present a corpus analysis of the data in which we demonstrate the potential for insights on a musical level. As a starting point, we seek to answer some queries about the harmonic and rhythmic functions of a typical walking bass line as represented in the data. A number of commercial jazz bass methods from different authors are summarised in [3] which we will refer to where appropriate. All analyses which follow were derived from the dataset by converting note-level information to a Pandas [18] dataframe using the Music21 Python library [19]. The queries used to perform the analysis will be released alongside the dataset.

4.1 Chord Degrees Used in Bass Line Construction

As jazz performance is a cultural practice, a strict set of rules for bass line construction has not been established. However, given the size of the proposed dataset we can start to provide a quantitative analysis of the choices made by performers during their improvisations.

Concerning the question of which chord degrees are favoured by the player, we analyse the function of each note in the dataset as it relates to the chord being played underneath it. In Figure 2 we see that bassists will favour the root note of the chord when constructing walking bass lines, as these are used in 32.7% of all notes played. This is rather basic from a musical perspective, but we can now point to data that bolsters existing empirical observations. When we examine the note played at each new chord change event, we see from Figure 2 that the use of chord roots is even more prevalent, with the proportion rising to 67.9% of the total. This reflects the role of the bass in outlining the harmony of the song.

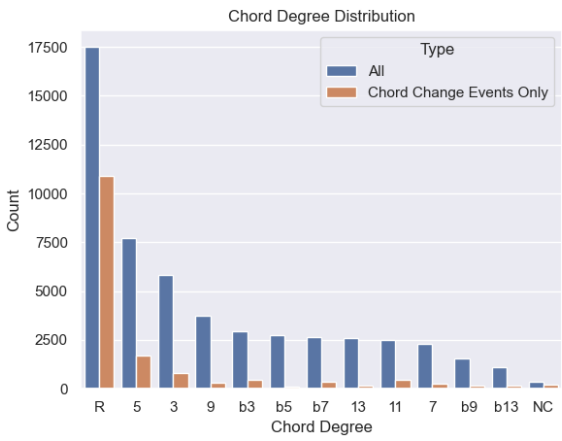


Figure 2. Global distribution of chord degrees

4.2 Use of Rhythmic Fills versus Quarter Note Pulse

In his educational method book, bassist Ron Carter [20] describes the process of adding rhythmic interest, or "fills", to a line. However, he cautions the student "not to overdo" their use before advising that: "personal tastes and judgement will govern this area of your playing". We can make an attempt to quantify this more precisely by examining what percentage of measures in the dataset contain a simple set of 4 quarter notes, and which deviate from this. We find that 62.81% of measures are indeed 4 quarter notes. While this is not a substitute for developing good taste, knowing this percentage might help in guiding a more analytical player.

4.3 Deriving Common Patterns

The annotations in this dataset also allow us to examine sequences of chord degrees that are commonly used in bass line construction. Over the 6400 chord symbols annotated, 3900 distinct patterns of chord degrees over chords are played. The 5 most common patterns for a chord lasting 4 beats are shown in Table 3. From these we can see a preference towards using tones from major and minor triads (i.e. 1, ♭3, 3 and 5). Given that the root movements in jazz are often perfect 4ths apart, we see that a number of the patterns approach the 4th via tones or semitones (i.e. from ♭3, 3, ♭5 or 5). This analysis of patterns only considers the chord degree, however a more detailed examination of patterns including sequential ideas and motifs is a subject of future work.

4.4 Semitone and V-to-I Approaches

In "Creating Jazz Basslines", author Jim Stinnet emphasises the use of semitone approaches. This is where target notes which fall on strong beats or chord changes are preceded by a note which is a semitone above or below the target (described in [3]). In this dataset we observe that this is indeed common, with ascending and descending semitones being the most often used intervals overall as shown in Figure 3. For notes which land on chord changes, semi-

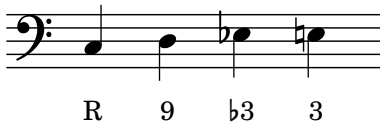
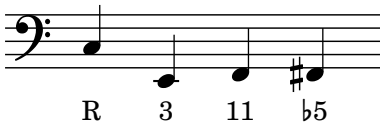
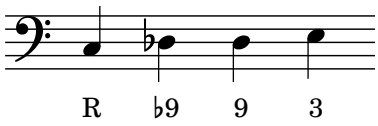
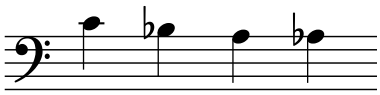
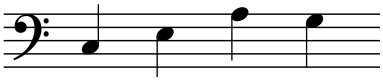
Pattern	Count	% of total
 R 9 ♭3 3	360	4.6%
 R 3 11 ♭5	195	2.5%
 R ♭9 9 3	113	1.43%
 R ♭7 13 ♭13	113	1.43%
 R 3 13 5	111	1.41%

Table 3. The five most common chord degree n-grams for 7898 chord instances lasting 4 or more beats. Examples are notated in C major for illustration.

tone approaches are even more prevalent. We summarise the most common intervals to approach chord changes (target tones) in Table 4.

In the "Walking Bassics" by Fuqua, Zisman, and Sher (described in [3]), the authors advocate the use of V to I movements for students however our data suggests that this is relatively uncommon in practice (9.66% ascending a perfect fourth and 4.30% descending). This is an interesting example of an idea that seems intuitive in theory (V to I is a strong bass movement for walking bass) but is not reflected in practice.

4.5 Step, Leap or Staying Put?

As we have seen in Section 4.1, in the majority of cases the performer will aim to play root notes when a new chord arrives but this leaves the question of how these root notes are typically connected together into a musically pleasing line. From the data, we can examine whether performers tend to use step-wise motion (tones and semitones), larger intervalic leaps (minor thirds or greater) or whether they choose to repeat a note. Looking at Figure 3 we see that there is a slight preference toward using step-wise motion (the largest group at 47.5%). Viewing the interval distribution plot we can also see that intervalic leaps are slightly more likely when the line is ascending especially for the interval of 5 semitones which corresponds to a perfect fourth.

Approach	Interval to target	Count	% of total
D♭ ↘ C	−1	4318	26.75
B♮ ↗ C	+1	3384	20.97
B♭ ↗ C	+2	1921	11.90
G ↗ C	+5	1560	9.66
C → C	0	1172	7.26
G ↘ C	−7	694	4.30
D ↘ C	−2	656	4.06

Table 4. The most common intervals used to approach a chord change (totalling 16141 events). For illustration all approaches are shown relative to a target tone of C.

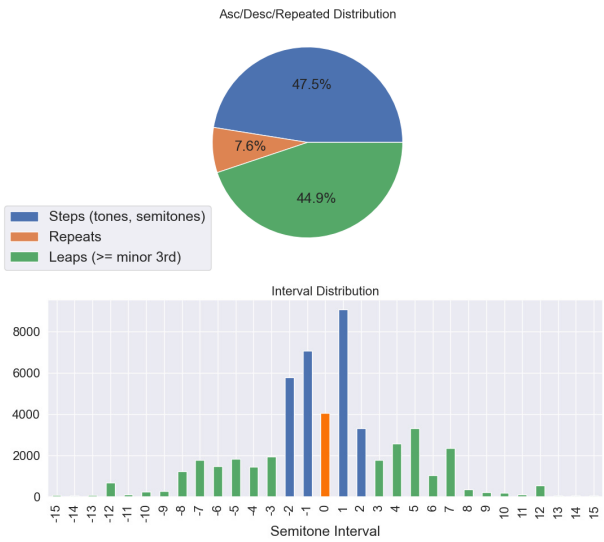


Figure 3. Distribution of intervals, grouped as step-wise movements (2 semitones or less), leaps (3 or more semitones) or repeats (no change from the preceding note).

4.6 Melodic Contour

The performer has a number of parameters available when improvising a bass line, one of which is the direction of the line. Sigi Busch (summarised in [3]) refers to the idea of "voice leading" within a bass line to link important chord tones while maintaining a direction, but none of the other methods summarised in [3] advise on how to choose directions or when to change them. Referring to the data now, we can see in Figure 4 that a high number of changes in direction is preferred, with the mean length of a sequence before a change falling at 2.46 notes. Intriguingly, the distribution of sequence lengths exhibits a power law. This phenomenon has been observed in several cases when analysing symbolic music corpora [21] but to our knowledge this is the first evidence in relation to walking bass lines.

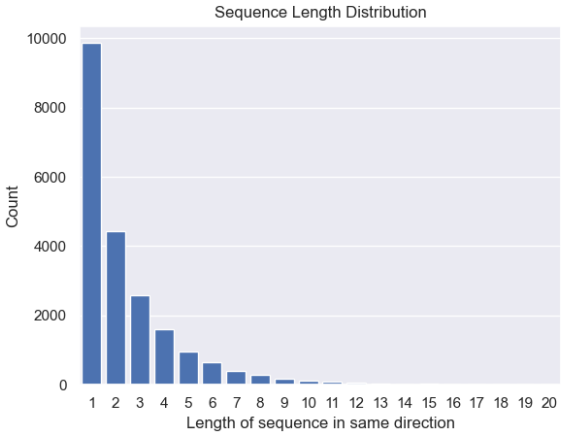


Figure 4. Sequence length (number of intervals) of lines maintaining a constant direction.

	CREPE Notes	Basic Pitch	Melodyne
R_{no}	74.11 ± 12.09	81.28 ± 6.26	79.52 ± 14.77
P_{no}	71.81 ± 13.33	51.40 ± 6.28	78.48 ± 15.41
F_{no}	72.89 ± 12.68	62.73 ± 5.55	78.95 ± 15.02
O	78.77 ± 2.68	65.24 ± 4.51	87.94 ± 3.91

Table 5. Automatic note transcription results for FiloBass, showing mean scores and standard deviation for Recall, Precision, F-measure and Overlap. Only onsets were evaluated and a timing tolerance of 50ms was used.

5. AUTOMATIC TRANSCRIPTION BASELINE

Using the accurate alignment data we have collected, we provide initial results for automatic note transcription — a bass line baseline. An exhaustive appraisal of transcription accuracy is beyond the scope of this work but we hope these results will encourage the use of this dataset in related future work.

We use the `mir_eval` [22] library to calculate precision, recall, F-measure and overlap scores. A default threshold of 50ms was used and only onset timings were considered. This is due to the difficulty of assessing offsets, as described in [23]. Three methods are examined for this task; the "Basic Pitch" package described in [23], the "CREPE Notes" method proposed in [24] and the commercial software Melodyne using the "Melodic" algorithm. The results from Melodyne were not manually corrected for this evaluation. Results for all methods are shown in Table 5. We see from these results that the proprietary commercial software outperforms the best research solutions for this dataset, however a significant amount of work is required to correct the remaining errors. During this work we also appreciated the Melodyne UI for note editing during our manual correction process. We note that similar projects in future may benefit from open source tools that allow a more streamlined note correction workflow.

6. DISCUSSION AND FUTURE WORK

In collating a dataset and performing a corpus analysis with reference to jazz bass methods, we hope to have provided useful insights into the role of the bass in jazz. The analysis provided here is not exhaustive however, and we hope that future research can reveal more about the mental model that performers use when constructing their bass accompaniment. In particular we hope to examine the role of timing, dynamics and use of sequential ideas in further work. We are also interested in pairing the FiloBass data with the FiloSax data for further analysis. The relationships between bass line and melody in a jazz setting could be explored further, with a view to developing more realistic generative models for both bass lines and solos.

We believe that the dataset has a wide number of potential uses beyond musicological analysis. Recent work on automatic music transcription (AMT) has highlighted that performance can be improved as more data is made available [2] and this dataset can help to address this need.

An additional task which we hope to address in future is that of automatic chord estimation (ACE). Following the hypothesis of Abeßer et al. [4], we believe that this data could be used to train a system to estimate chords from the bass line directly. Chord estimation is a particularly challenging task in the jazz setting due to the rich harmonic vocabulary so novel approaches here may be welcome.

The scores which were produced as part of this data should also be valuable to researchers, as they provide a potential source of training data and evaluation for monophonic score processing tasks. In particular, they will be useful for rhythmic parsing (quantisation), automatic score layout and related sub-tasks such as spelling of accidentals.

7. CONCLUSIONS

We present FiloBass: a new dataset for jazz bass lines. Making use of the detailed annotation data, we are able to demonstrate a quantitative approach to reinforce traditional musicological analysis of the role of the bass in jazz performance.

Through examination of this dataset we demonstrate that a number of rules put forward in jazz bass method books are supported by larger scale data. These can be summarised as follows: the root note of the chord is usually played on the first beat of a new chord; this root is approached via a semitone step where possible; the rhythm comprises a quarter note pulse most of the time; a balance is maintained between ascending and descending contours. We are aware though, that any analytical project of this sort cannot be truly comprehensive and can only offer a guide to the performer. The musical context and the taste and experience of the musician will determine when to follow the "default" most likely path and when to choose a different route.

8. ACKNOWLEDGEMENTS

The first author 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].

9. REFERENCES

[1] D. Foster and S. Dixon, "Filosax: A dataset of annotated jazz saxophone recordings," in *Proceedings of the 22nd International Society for Music Information Retrieval Conference*, Online, 2021, pp. 205–212.

[2] 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 (ICLR)*, 2022.

[3] H. Pinheiro, "Jazz bass method books versus actual performance: The case study of Charlie Haden," Master's thesis, University of Louisville, 2018. [Online]. Available: https://ir.library.louisville.edu/etd/2939

[4] J. Abeßer, S. Balke, K. Frieler, M. Pfleiderer, and M. Müller, "Deep learning for jazz walking bass transcription," in *AES International Conference on Semantic Audio*. Audio Engineering Society, 2017.

[5] J. Abeßer and S. Balke, "Improving bass saliency estimation using label propagation and transfer learning," in *Proceedings of the 18th International Society for Music Information Retrieval Conference*, Paris, France, 2018, pp. 306–312.

[6] J. Abeßer and M. Müller, "Jazz bass transcription using a U-Net architecture," *Special Issue on Machine Learning Applied to Music/Audio Signal Processing, Electronics*, vol. 10, no. 6, Jan. 2021.

[7] K. Frieler, F. Höger, M. Pfleiderer, and S. Dixon, "Two web applications for exploring melodic patterns in jazz solos," in *Proceedings of the 18th International Society for Music Information Retrieval Conference*, Paris, France, 2018, pp. 777–783.

[8] M. Goto, "Development of the RWC Music Database," in *Proceedings of the 18th International Congress on Acoustics (ICA)*, vol. 1, 2004, pp. 553–556.

[9] A. Shiga and T. Kitahara, "Generating walking bass lines with HMM," in *Perception, Representations, Image, Sound, Music CMMR*. Springer International Publishing, 2021, pp. 248–256.

[10] O. Araz, "Automatic bassline transcription for electronic music," in *Late Breaking Demo at the 22nd International Society for Music Information Retrieval Conference*, 2021. [Online]. Available: https://archives.ismir.net/ismir2021/latebreaking/000016.pdf

[11] 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 *Proceedings of the 15th International Society for Music Information Retrieval Conference*, Taipei, Taiwan, 2014, pp. 155–160.

[12] J. Salamon, R. M. Bittner, J. Bonada, J. J. Bosch, E. Gómez, and J. P. Bello, "An analysis/synthesis framework for automatic F0 annotation of multitrack datasets," in *Proceedings of the 18th International Society for Music Information Retrieval Conference*, Suzhou, China, 2017, pp. 71–78.

[13] J. Abeßer, H. Lukashevich, and G. Schuller, "Feature-based extraction of plucking and expression styles of the electric bass guitar," in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, Mar. 2010, pp. 2290–2293.

[14] S. Rouard, F. Massa, and A. Défossez, "Hybrid transformers for music source separation," Nov. 2022, arXiv:2211.08553 [cs, eess]. [Online]. Available: http://arxiv.org/abs/2211.08553

[15] C. Raffel and D. P. W. Ellis, "Intuitive analysis, creation and manipulation of MIDI data with pretty_midi," in *Late Breaking Demo at the 22nd International Society for Music Information Retrieval Conference*, 2014. [Online]. Available: https://ismir2014.ismir.net/LBD/LBD29.pdf

[16] E. Nakamura, K. Yoshii, and H. Katayose, "Performance error detection and post-processing for fast and accurate symbolic music alignment," in *Proceedings of the 18th International Society for Music Information Retrieval Conference*, Suzhou, China, 2017, pp. 347–353.

[17] B. Maman and A. H. Bermano, "Unaligned supervision for automatic music transcription in the wild," in *International Conference on Machine Learning*, vol. 162. ICML, 2022, pp. 14 918–14 934.

[18] W. McKinney *et al.*, "Data structures for statistical computing in Python," in *Proceedings of the 9th Python in Science Conference*, vol. 445. Austin, TX, 2010, pp. 51–56.

[19] 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*, Utrecht, Netherlands, 2010, pp. 637–642.

[20] R. Carter, *Building Jazz Bass Lines*, ser. Bass Builders. Hal Leonard, 1998.

[21] D. Rafailidis and Y. Manolopoulos, "The power of music: Searching for power-laws in symbolic musical data," in *12th Panhellenic Conference on Informatics*, Jan. 2008.

[22] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, and D. P. W. Ellis, "A transparent implementation of common MIR metrics," in *15th International Society for Music Information Retrieval Conference*, 2014.

[23] 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 *2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 781–785.

[24] X. Riley and S. Dixon, "CREPE Notes: A new method for segmenting pitch contours into discrete notes," in *Proceedings of the 20th Sound and Music Computing Conference*, Stockholm, Sweden, 2023, pp. 1–5.

COMPARING TEXTURE IN PIANO SCORES

Louis Couturier[1] **Louis Bigo**[2] **Florence Levé**[1,2]
[1] MIS, Université de Picardie Jules Verne, Amiens, France
[2] Univ. Lille, CNRS, Centrale Lille, UMR 9189 CRIStAL, F-59000 Lille, France
{louis.couturier, florence.leve}@u-picardie.fr, louis.bigo@univ-lille.fr

ABSTRACT

In this paper, we propose four different approaches to quantify similarities of compositional texture in symbolically encoded piano music. A melodic contour or harmonic progression can be shaped into a wide variety of different rhythms, densities, or combinations of layers. Instead of describing these textural organizations only locally, using existing formalisms, we question how these parameters may evolve throughout a musical piece, and more specifically how much they change. Hence, we define several distance functions to compare texture between two musical bars, based either on textural labels annotated with a dedicated syntax, or directly on symbolic scores. We propose an evaluation methodology based on textural heterogeneity and contrasts in classical Thema and Variations using the TAVERN dataset. Finally, we illustrate use cases of these tools to analyze long-term structure, and discuss the impact of these results on the understanding of musical texture.

1. INTRODUCTION

The term *texture* is used at various levels of description in the music domain. Initially related to the description of sound features, it is also used in symbolic representations of music to describe musical streams through a variety of concepts characterizing the volume and the organization of basic score elements such as notes and voices, which encompass high-level concepts such as monophony and polyphony [1] [1–4]. Between these two extremes, elements of musical texture include layers, voices, melodic or rhythmic patterns, articulation and instrumentation [2,5]. Huron interestingly summarizes it in three main ideas [3]: (1) the density of musical elements, (2) the diversity or inhomogeneity of elements, and (3) the overall sonic activity. The first two can be included in the notion of *compositional texture*, as opposed to orchestral (or timbral) texture [6]. Compositional texture, which is the object of study of the present work, is mostly embedded in the symbolic score. It is worth noting that some models of texture only focus on a particular musical dimension. Nordgren's categorization for instance deals with the vertical dimension only, with note doubling and spacing [7]. Conversely, other approaches focus on the time dimension, as a *complement* to harmony, as in [8], or [9] for style-transfer. Figure 1 shows multiple versions of the same musical theme, which is shaped into different compositional textures.

We aim at quantifying the differences of compositional texture in piano music. Previous studies provided local descriptions of texture [10–12]. Here, we question how textural dimensions may evolve through a whole piece of music. This objective requires the elaboration of dedicated tools to compare textures, more precisely to assess the distance, or dissimilarity, between two given textural configurations.

A number of Music Information Retrieval (MIR) tasks involve the search of similarities at various scales, from pattern detection [13, 14] to genre classification [15, 16]. In the audio domain, music similarity lies at the center of content-based recommender systems [17, 18]. At the level of the musical score, music similarity has also been extensively studied on specific musical notions such as melody [13–15, 19–22], rhythmic pattern [23, 24], or chord and harmonic progressions [25–27]. Classical approaches for computing similarities include edit-distance on string-based representation, or geometric distance on pianoroll-like representations [28]. New latent embeddings of music also emerged from development of deep neural networks, as well as metric learning methods, like [29]. Although measures of music similarity may ultimately reflect some similarities in the perception of music [30,31], we compare textural information based on symbolic music scores only. In particular, we focus here on studying different representations of texture in order to build interpretable dissimilarity measures.

In this work, we propose distance functions to quantify textural dissimilarity between musical bars from piano scores. We first detail four types of textural distance (Section 2). Then, we introduce estimators of textural heterogeneity and contrast, for longer musical extracts, and propose a dedicated methodology to evaluate our distances, using a dataset of Thema and Variations (Section 3). Finally, we provide use cases of such distances, especially in the context of form or structure analysis (Section 4).

[1] Polyphony, as a type of texture, has a stronger meaning than "the simultaneous presence of possibly more than one note". Here, it implies "two or more lines moving independently" [1]. Similarly, monophonic texture is not restricted to single melodies, but designates the presence of a unique musical line – possibly with note doubling or parallel motions.

© L. Couturier, L. Bigo, and F. Levé. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).
Attribution: L. Couturier, L. Bigo, and F. Levé, "Comparing Texture in Piano Scores", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

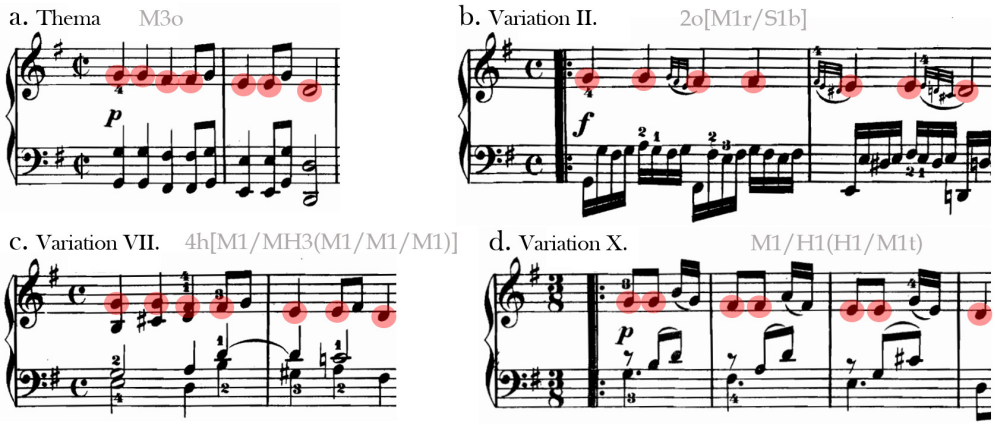


Figure 1: Examples of different textures from *Ten Variations in G on 'Unsere dummer Pöbel meint'* by W. A. Mozart (K. 455, 1784). A textural annotation, following the syntax defined in [10], is provided for each example. The melodic contour (circled in red) is shared among the variations, but the overall compositional texture changes. a. The theme is introduced in monophonic texture: three voices merge into a single musical idea, in parallel octave motions; b. There are now only two notes sounding at the same time: the vertical density decreases. But horizontal density is increased by sixteenth notes; c. A more homophonic texture: three or four threads, mostly synchronous; d. Here, we identify two layers of melody and accompaniment. In these last three bars, the harmony changes, but compositional texture is exactly the same.

2. DEFINING DISTANCES FOR COMPOSITIONAL TEXTURE

The distances that are designed in this paper aim at comparing compositional texture at the scale of individual musical bars. We focus on (polyphonic) piano scores of the Western Classical repertoire, with no voice separation.

2.1 Distances based on textural labels

Textural annotations have been produced in [6] for piano music, on Mozart's sonatas. For each annotated bar, a label enclose two levels of textural information: on the one hand, a set of keywords that indicate the presence of certain properties of the overall textural configuration, or in one of its layer (like parallelism, melodic or harmonic roles...); on the other hand, a vertical structuration of the musical content into main textural layers and sublayers [10]. We propose two distance functions based on this information.

2.1.1 Distance between textural elements

The first distance is based on a set of binary *textural elements* which have been defined in [6, section 3.2]. These indicators express the presence of atomic textural characteristics in a musical bar. They include specific functions of the musical layers: melodic (`M`), harmonic (`H`), or static (`S`), relationships between voices: homorhythmy (`h`), parallel (`p`) or octave (`o`) motions, as well as characteristic musical figures such as sustained (`t`) or repeated (`r`) notes, scale motives (`s`), oscillations (`b`), sparse horizontal density (`_`) and neat changes of texture in the bar (`,`).

A musical bar a is therefore abstractly represented by a vector *texel*(a) which comprised of the 12 textural elements from its label. The distance function d_{texel}returns the Hamming distance between the vectors. It is an integer between zero and 12 that corresponds to the number of textural elements that differ between two bar annotations:

$$d_{texel}(a,b) = \sum_{i=1}^{12} |\text{texel}_i(a) - \text{texel}_i(b)|$$

where a and b are two musical bars, and $\text{texel}_i(\cdot)$ the binary value of the ith textural element of a given bar.

2.1.2 Textural diversity and density

At a higher level of description, textural annotation of piano scores mainly focus on grouping threads [2] of notes into distinct musical layers. Examples 1.a and 1.d both have three threads, but they are organized differently. In 1.a, they merge into a single layer. On the contrary, in 1.d, the threads are divided in 2 main textural layers: its texture is more *diverse* than 1.a without being thicker.

This grouping of threads is formalized in [10] under the terms of *density* (number of threads, of simultaneous sounding notes; the thickness or *density-number* in [2]) and *diversity* (number of distinct layers). These two dimensions allow to embed any textural label in the planar textural space represented in Figure 2 (left).

The *density-Diversity* distance d_{dD} separating two bars is defined as the Euclidean distance between their labels in this space. In previous examples, the first bar of 1.a and 1.d respectively have density-diversity coordinates of (3,1) and (3,2), resulting in a distance of 1.

Note that this distance only takes into account the vertical dimension of compositional texture (as in [7]). A drawback of restricting textural analysis to only two dimensions is that it is less sensitive to small textural fluctuation: as an example, the 1164 labels released by [6] only use 17 distinct combinations of density and diversity values. Nevertheless, this condensed description allows an interpretable

[2] The term 'thread' designates the most atomic elements that can be combined into musical 'layers' [5, p.65].

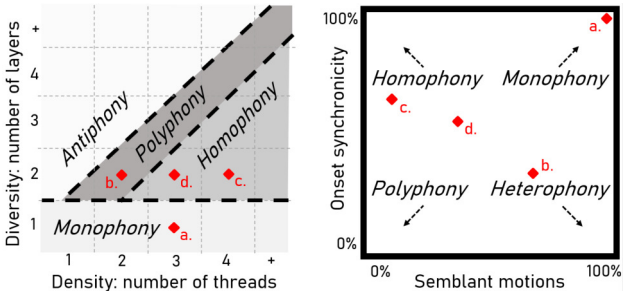


Figure 2: Schematic representations of textural spaces proposed respectively by Couturier et al. [6] (left) and Huron [3] (right). In both case, areas of the space are matched to *main types* of texture [1, 4]. The boundaries are not strict, though. The four examples of Figure 1 are also represented in these spaces.

approach for high-level analysis, as it reflects main textural strategies (see Figure 2).

2.2 Score-based distances

The distances defined in Section 2.1 are based on manually-annotated textural labels. Such textural annotations are however rarely available as their production requires substantial time and expert knowledge. In contrast, this section presents two distance functions that can systematically be computed on encoded musical scores.

2.2.1 Adapting Huron's textural space

Another two-dimensional textural space has been proposed by Huron in [3]. It is used in this article to categorize full musical pieces among *main types* of texture (see also [1]): polyphony, monophony, homophony and heterophony, represented on Figure 2. Instead of analyzing the quantity and grouping of musical threads (see Section 2.1.2), it relies on the relationships between them: the proportions of *onset synchronization* and *semblant motions*.

The original study used a pre-existing separation of voices in the pieces (such as Bach's Inventions and Sinfonias) to compute these features. For both of them, we provide an estimation of the value in the interleaved polyphonic case – i.e. without separation of voices. Note that it is not always possible to find a valid and unique voice separation in piano scores [32]. The details of the implementation, out of the scope of the article, can be found in the dedicated repository [1]. They are:

- The ratio of *onset synchrony* quantifies the degree of homorhythmy of the note onsets. A value of 1 indicates a perfect synchronization of note onsets, which is the case in monophonic (see Figure 1.a) or homophonic (chordal or hymnal) textures. This value decreases if note onsets happens while other notes are sustained. For example, 1.b has a value of 0.25: in this case, only one onset over four is fully synchronous.
- The ratio of *semblant motions* estimates to what extent the directions of pitch motions are similar. These feature has its maximal value in the case of monophony, once again, whereas the presence of multiple concurrent layers with opposite motions will reduce its value.

We use these dimensions to build a new distance d_{huron}(the *Huron* distance) between two bars, which is obtained by summing their differences of *onset synchrony* values and *semblant motions* values, using our implementation in the polyphonic interleaved case. This corresponds to the Manhattan distance between their respective coordinates in this textural space.

2.2.2 Features of density

We present a last set of three distances based on low-level textural features, focusing on vertical and horizontal density. On the one hand, vertical density refers to the thickness of the texture, the number of simultaneous notes – similarly to the density evoked in Section 2.1.2. On the other hand, horizontal density describes the volume of successive notes, and their position in time. For both dimensions, we use a value of volume and a value of dispersion:

- `vert_avg`: average thickness, in number of notes. After slicing the bar into successive pitch sets, we count the number of pitches in each slice, weighted by their duration.
- `vert_std`: standard deviation of the number of pitches in each onset of one or more notes.
- `horiz_avg`: average number of onsets per beat. The duration of one beat is inferred from the bar time signature.
- `horiz_std`: standard deviation of the regularity of onsets, i.e. around the average duration between successive onsets.

We use Manhattan distance to compare two vectors of features, computed on two target bars. We define and test three variants of this distance: based on the two horizontal features only (d_{h}), on the two vertical ones (d_{v}), or on all the four (d_{hv}).

2.3 Implemention details and release

The features are extracted from the musical scores using intermediate Tab-Separated Values (TSV) files, which contain a list of notes (see [33]). The code [1], in Python, includes a converter to this format from both Humdrum **kern [34] and musicXML formats, using music21 Python library [35].

3. EVALUATING TEXTURAL DISTANCES

3.1 Dataset

To evaluate the relevance of the distances proposed in the previous section, we use bars from classical *Thema*

[1] Available at `http://algomus.fr/code`.

and variations. [36] emphasizes the links between musical variations in general, and musical similarity. In the genre of Thema and variations, a theme is reproduced in short sections with various changes of (textural) parameters, but in a way that allow to recognize the original melodic contour and/or harmony; as in Figure 1. This structure has the advantage of providing both dissimilar examples (in distinct variations), and similar examples (in the same variation).

Although no explicit mention of textural homogeneity within variations has been found in musicological literature, authors more often insist on the higher contrast between distinct variations [37, p.570]. The genre of Thema and variations provides "the largest esthetic spectrum" [38], and this variety of content is valuable in our case. We rely on this fundamental assumption for the rest of the paper: *on average, a musical bar is more similar – in texture – to a bar from the same musical phrase, than to any other bar in another variation or piece*.

We use the TAVERN dataset [39], which consists in 27 sets of thema and variations by Mozart (10) and Beethoven (17). The variations are already segmented into structural phrases, totalling 1060 of them in the whole dataset. We take those phrases as structural units in which we use the score-based distances defined in section 2.2. Further annotations of texture would be required to apply label-based distances on this dataset.

Remark. The texture of phrases can vary within the same variation, to a lesser extent – this is generally the case in bipartite or tripartite variations, which is a common structure in this context [37, 38]. Changes of mode (major/minor) often occur, in general at least once per set of variations. This change is not considered as textural, but it is often accompanied by changes of other musical parameters that are in the scope of texture, so it would still add valuable information.

3.2 Heterogeneity and contrast

To evaluate textural dissimilarities on full musical extracts, we introduce two indicators:

- *heterogeneity*: the heterogenity (h_d) within a single set of bars corresponds to *the average distance between pairs of distinct bars from the set*, for a given distance function d.
- *contrast*: the contrast (c_d) between two sets of bars is defined as *the average distance value between pairs of bars from the two extracts*, for a given distance function d.

More formally, we have:

$$h_d(S) = \underset{\forall (m_i,m_j)\in S^2, i\neq j}{avg} d(m_i, m_j)$$

$$c_d(S_1, S_2) = \underset{\forall m_i\in S_1, \forall m_j\in S_2}{avg} d(m_i, m_j)$$

where avg is the arithmetic mean operator, S, S_1 and S_2 are sets of bars, and m_i, m_j denote bars/measures in those sets.

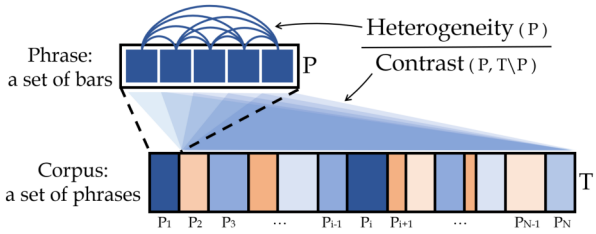


Figure 3: Schematic representation of the computation of heterogeneity of a specific phrase (arcs above) and contrast (links between the bars of phrase P and all the other bars outside P, in the corpus T). Our evaluation metric is the average value of this ratio for all the phrases of the corpus.

The heterogeneity is a measure of dispersion: a lower value means that samples in the extract are more similar between each other (given a distance d). We specifically ignore the comparisons of a bar with itself to reduce the influence of the size of S.

Let us illustrate those two indicators for the descriptor-based distance d_{hv} (Section 2.2.2) using examples Figure 1.a and Figure 1.d. We note $S_a = \{$bars of Figure 1.a$\} = \{a_1, a_2\}$ and $S_d = \{$bars of Figure 1.d$\} = \{d_1, d_2, d_3\}$. In S_a, the heterogeneity is simply equal to the distance between its two bars: small differences occur in horizontal density, but not in vertical density. We obtain a value of 0.6. To compute the *heterogeneity* in S_d, we have three possible unordered pairs of bars to compare ($\{d_1, d_2\}$, $\{d_1, d_3\}$ and $\{d_2, d_3\}$); however, the texture in these bars is precisely the same regarding d_{hv}, resulting a in value of zero of heterogeneity. The inequality $h_d(S_a) > h_d(S_d)$ can be interpreted as "S_a is more texturally heterogeneous than S_d, with regards to distance d". The contrast c_d between S_a and S_d is 1.825. In general, the inequality $h_d(S_a) < c_d(S_a, S_d)$ means that a bar in S_a is, on average, more similar to other bars in S_a than bars in S_d.

3.3 Evaluation methodology

We evaluate how heterogeneous the texture is within each phrases of the TAVERN dataset, compared to the rest of the corpus. Under the assumption exposed in Section 3.1, we assess the quality of a textural distance d by looking for the lowest *average Relative Heterogeneity* on TAVERN phrases (T):

$$\text{aRH}_T(d) = \underset{\forall P_i\in T}{avg} \left(\frac{h_d(P_i)}{c_d(P_i, T\setminus P_i)}\right)$$

where T is the set of all the phrases in TAVERN dataset, P_i is the ith phrase of the dataset, and d is a textural distance function between individual musical bars.

This process is schematized in Figure 3. For a given phrase, if the ratio between intra-phrase heterogeneity and inter-phrases contrast is very low, it means that the extract is rather homogeneous, and that this texture – or whatever the distance d represents – is rather specific to this extract compared to the rest of the corpus. If this ratio is above 1, it means that the bars in this phrase are more similar to other

Distance	d	$\text{aRH}_T(d)$
Horiz. and Vert. density features	d_{hv}	0.51
Horizontal density features	d_{h}	0.39
Vertical density features	d_{v}	0.64
Huron's textural space	d_{huron}	0.72
Comparison: Pitch class content	d_{pc}	0.80

Table 1: Evaluation of textural distances using the *Average Relative Heterogenity* on phrases of the TAVERN dataset ($\text{aRH}_T(d)$), to minimize.

bars outside the phrases than between themselves. Put differently: a value below 1 show that intra-phrase distances (heterogeneity) are smaller than inter-phrases comparison (contrast with the corpus). The value of $\text{aRH}_T(d)$ is the average of this ratio on all the phrases of the corpus.

Remarks. We could directly compute values of *contrast* or *heterogeneity* on reference data, using different textural distance d_i and opt for the most convincing values. However, these values are not directly comparable if they are based on different distances: they are average values of specific distances, and thus follow their respective – and possibly very different – order of magnitude. Also note that the *contrast* is not a distance function (or metric) because the contrast between the same set of bars could be different from zero – if its bars that are not all the same. The functions presented in Section 2 *are* metrics, applied to different representations of texture in a musical bar.

3.4 Results

The results, for all score-based distances, are shown in Table 1. Using the distance based on all density features (d_{hv}), the aRH_T of 0.51 indicates that a musical bar is, on average, a twice more similar to bars in the same phrase than to the rest of the corpus. The use of horizontal density features alone (d_{h}) improves this value (0.39), highlighting the importance of the time dimension to discriminate between textures.

For comparison, we integrate an additional distance (d_{pc}) that describe not textural but harmonic content – computing Euclidean distance between pitch-classes profile. Its aRH_T of 0.80 is still below 1, which means that intra-phrase d_{pc}values (heterogeneity) are smaller than inter-phrase comparison (contrast with the corpus); this is not surprising in tonal music. But most importantly, this evaluation metric value is higher than for all other textural distances. This gap contributes to validate the use of Thema and Variations as a source of empiric ground truth examples of textural similarities.

3.5 Links between distances

In Table 2, we display correlations between all the distances defined in Section 2. They are computed on all distinct pairs of bars among 1160 from Mozart piano sonatas (K. 279, K. 280, K. 283), for which we have both textural annotations [6] and encoded scores [33]. We use Spearman correlation, that depicts similarities of rankings of these values.

	d_{hv}	d_{h}	d_{v}	d_{huron}	d_{dD}	d_{texel}
d_{hv}	1.00	"	"	"	"	"
d_{h}	**0.95**	1.00	"	"	"	"
d_{v}	**0.33**	0.05	1.00	"	"	"
d_{huron}	0.10	0.10	0.05	1.00	"	"
d_{dD}	0.19	0.03	**0.53**	0.03	1.00	
d_{texel}	0.10	0.06	0.14	0.01	0.20	1.00

Table 2: Spearman correlation between textural distances as defined in Section 2, evaluated on all pair of bars in three Mozart piano sonatas (K. 279, K. 280, K. 283).

	d_{hv}	d_{h}	d_{v}	d_{huron}	d_{dD}	d_{texel}
Horizontal density (time dimension)	×	×				×
Vertical density (thickness)	×		×		×	×
Semblant motions, parallelism				×		×
Roles of layers (melody, acc. ...)						×
Main types of texture (see Fig.2)				×	×	
Computed on symbolic scores	×	×	×	×		
Computed on annotated labels					×	×

Table 3: Summary of the distances defined in Section 2, and the different dimensions of compositional texture that they take into account.

Huron-space distance (d_{huron}) and texel distance (d_{texel}) seem independant from other distances. We explain it by the fact that our distances focus on different dimensions of texture, summarized in Table 3. In particular, d_{texel}covers a wide range of abstract concept and is the only function that deals with the roles of layers, which is difficult to approximate using low-level features. Otherwise, we find that using horizontal density features only (d_{h}) gives a very similar behavior than using all four density features (d_{hv}) – with a correlation of 0.95. Although Density-Diversity distance (d_{dD}) and vertical-feature distance (d_{v}) deal with very different level of abstraction, they correlates positively (0.53), as they both focus on the vertical dimension of texture.

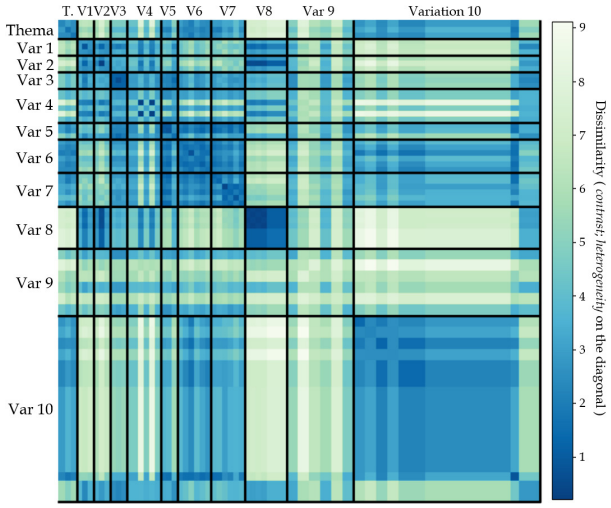


Figure 4: Textural dissimilarities between the phrases of *Ten Variations in G on 'Unsere dummer Pöbel meint'* by W. A. Mozart (K. 455, 1784). Intersections are colored according to *contrast* values using d_{hv}distance (Section 2.2.2), and *heterogeneity* of phrases on the diagonal (Section 3.2). The phrases are scaled according to their size in number of bars (totalling 338 in the whole piece). The most similar extracts are shown in dark blue, whereas light green indicates higher dissimilarity. – We identify blocks of consecutive similar variations, such as (1,2,3) or (5,6,7); inner structure of varations may reveal contrasting segments in the case of Variation 4; Variation 9 is very contrasted due to the alternation between chordal texture and fast melodic lines; the penultimate phrase comes back to the original texture of the Thema.

4. USE CASES FOR STRUCTURE ANALYSES

4.1 Long-term textural dissimilarities

The *contrast* defined in Section 3.2 can be used as a dissimilarity measure between any sets of bars, from individual pieces to entire corpora. It may also emphasize the relationships between sections, or phrases, of a given piece of music. Figure 4 shows an example of self-similarity matrix based on textural contrast between phrases of a piece in Thema and Variations form. Beyond the case of Thema and Variations, the contrast measure gives an overview of the piece macrostructure, and may even link thematic material up to transposition, such as recapitulation parts in sonata form. More generally, the proposed distances can lead to promising and original approaches for automatic structure segmentation.

4.2 Short-term textural changes

In this paper, we assume lower textural heterogeneity within phrases of thema and variations. But in the general case, changes of textures may occur in the middle of a phrase. Following the intuition that in-phrase texture changes mostly occur in openings and endings of phrases in the TAVERN dataset, we evaluate d_{hv}using the same methodology as in Section 3.3, but systematically ignore the last bar of each phrase of the corpus. We find that $\text{aRH}_T(d_{\text{hv}})$ decreases from 0.51 to 0.42. When removing each first bar instead, it drops to 0.34. In comparison, removing the second or third bar of the phrases increases the original value of $\text{aRH}_T(d_{\text{hv}})$ to respectively 0.55 and 0.54. This shows that the 'core' of phrases have slightly more textural homogeneity, and most importantly that openings and endings are less similar to the middle of phrases. Typical examples are transitional melodies and final chords in cadences – which often constrast with the rest of the phrase. We believe that our distance can be used to study more precisely these local changes of texture within short sections.

5. CONCLUSION AND FURTHER WORKS

The textural distances proposed in this paper give promising perspectives for the computation of multi-level similarities in symbolic music. On the one hand, comparing textural labels allows to rely on expert data, which is already known as texturally meaningful. This information already carries a lot of abstraction, but it is costly to produce in practice and can lead to a certain amount of subjectivity [40]. Moreover, the low amount of available annotations hinders our ability to evaluate the quality of these distances. On the other hand, using symbolic features that can be computed automatically is more practical, and also more objective. In further work, we plan to investigate the best features to use at a more global scale, as well as their relative contibution.

Although the proposed distances are drawn at the level of musical bars, we elaborated a more global dissimilarity measure to compare sets of several bars, and highlight textural contrast between and within structural sections of musical pieces. This measure made possible a quantitative evaluation of textural distances on a corpus of Thema and Variations, based on the assumption that texture is more dissimilar between two distinct variations, and more homogeneous within single variations.

Our distances capture different facets of compositional texture, at different levels of abstraction (see Table 3). Focusing on more atomic and independant textural aspects can enhance the precision and the interpretability of our analyses. However, ensuring a proper disentanglement of such dimensions remains a major challenge. Integrating Thema and variations in the evaluation methodology is a step further to link theoratical models of texture to concrete, and somewhat intuitive, examples. It contributes to a better understanding of some models of texture, but also of musical texture itself.

A potential continuation of this work is to broaden the scope of our experiments to other repertoires. We believe that the tools introduced in this paper are easily extendable to other styles of written polyphonic music, or to other instruments. In the meantime, the present experiments on Western classical piano music already offer promising opportunities of quantitative analyses of texture with regards to genre, style, form or harmony.

6. ACKNOWLEDGMENT

We want to thank Jean-Paul Chehab for fruitful discussion at the beginning of the project. We also thank Dinh-Viet-Toan Le, Alexandre D'Hooge and the rest of the Algomus team, as well as the anonymous reviewers for their constructive feedback. This research is partly funded by Région Hauts-de-France and by Agence Nationale de la Recherche (ANR), project TABASCO ANR-22-CE38-0001.

7. REFERENCES

[1] B. Benward and M. Saker, *Music: In Theory and Practice, Vol. I. Eigth Edition.* McGraw-Hill, 2008, ch. 7, p. 145–162.

[2] W. Berry, *Structural functions in music: A Probing Exploration of Conceptual Bases and Techniques for the Analysis of Tonality, Texture, and Rhythm.* Prentice-Hall, 1976.

[3] D. Huron, "Characterizing musical textures," in *International Computer Music Conference (ICMC 1989)*, 1989, pp. 131–134.

[4] J. Dunsby, "Considerations of texture," *Music & Letters*, vol. 70, no. 1, pp. 46–57, 1989.

[5] D. M. de Sousa, "Textural design: A compositional theory for the organization of musical texture," Ph.D. dissertation, Universidade Federal do Rio de Janeiro, 2019.

[6] L. Couturier, L. Bigo, and F. Levé, "A dataset of texture annotations in Mozart piano sonatas," in *International Society for Music Information Retrieval Conference (ISMIR 2022)*, 2022.

[7] Q. R. Nordgren, "A measure of textural patterns and strengths," *Journal of Music Theory*, vol. 4, no. 1, pp. 19–31, 1960.

[8] Z. Wang, D. Wang, Y. Zhang, and G. Xia, "Learning interpretable representation for controllable polyphonic music generation," in *International Society for Music Information Retrieval Conference (ISMIR 2020)*, 2020.

[9] O. Cífka, U. Şimşekli, and G. Richard, "Groove2Groove: one-shot music style transfer with supervision from synthetic data," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2638–2650, 2020.

[10] L. Couturier, L. Bigo, and F. Levé, "Annotating symbolic texture in piano music: a formal syntax," in *Sound and Music Computing Conference (SMC 2022)*, 2022.

[11] M. Giraud, F. Levé, F. Mercier, M. Rigaudière, and D. Thorez, "Towards modeling texture in symbolic data," in *International Society for Music Information Retrieval Conference (ISMIR 2014)*, 2014, pp. 59–64.

[12] D.-V.-T. Le, M. Giraud, F. Levé, and F. Maccarini, "A corpus describing orchestral texture in first movements of classical and early-romantic symphonies," in *Digital Libraries for Musicology (DLfM 2022)*, 2022, pp. 22–35.

[13] D. Conklin, "Pattern in music," *Journal of Mathematics and Music*, vol. 15, no. 2, pp. 95–98, 2021. [Online]. Available: https://doi.org/10.1080/17459737.2021.1947404

[14] B. Janssen, W. B. de Haas, A. Volk, and P. van Kranenburg, "Finding repeated patterns in music: State of knowledge, challenges, perspectives," in *International Symposium on Computer Music and Multidisciplinary Research (CMMR 2013)*, 2013, pp. 277–297.

[15] 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*, 2018, pp. 34–37.

[16] J. Ens and P. Pasquier, "Quantifying musical style: Ranking symbolic music based on similarity to a style," *arXiv preprint arXiv:2003.06226*, 2020.

[17] P. Knees and M. Schedl, *Music similarity and retrieval: an introduction to audio-and web-based strategies*. Springer, 2016, vol. 9.

[18] M. Schedl, P. Knees, B. McFee, and D. Bogdanov, "Music recommendation systems: Techniques, use cases, and challenges," in *Recommender Systems Handbook*. Springer, 2021, pp. 927–971.

[19] T. Collins, J. Thurlow, and R. Laney, "A comparative evaluation of algorithms for discovering translational patterns in baroque keyboard works," in *International Society for Music Information Retrieval Conference (ISMIR 2010)*, 2010, pp. 3–8.

[20] V. Velardo, M. Vallati, and S. Jan, "Symbolic melodic similarity: State of the art and future challenges," *Computer Music Journal*, vol. 40, no. 2, pp. 70–83, 2016.

[21] F. Karsdorp, P. van Kranenburg, and E. Manjavacas, "Learning similarity metrics for melody retrieval." in *ISMIR*, 2019, pp. 478–485.

[22] S. Park, T. Kwon, J. Lee, J. Kim, and J. Nam, "A cross-scape plot representation for visualizing symbolic melodic similarity." in *ISMIR*, 2019, pp. 423–430.

[23] F. Bruford, O. Lartillot, S. McDonald, and M. Sandler, "Multidimensional similarity modelling of complex drum loops using the groovetoolbox," in *International Society for Music Information Retrieval Conference (ISMIR 2020)*, 2020, pp. 207–215.

[24] D. Cocharro, G. Bernardes, G. Bernardo, and C. Lemos, "A review of musical rhythm representation and (dis) similarity in symbolic and audio domains," *Perspectives on Music, Sound and Musicology: Research, Education and Practice*, pp. 189–208, 2021.

[25] T. Rocher, M. Robine, P. Hanna, and M. Desainte-Catherine, "A survey of chord distances with comparison for chord analysis," in *ICMC*, 2010.

[26] W. B. De Haas, M. Robine, P. Hanna, R. C. Veltkamp, and F. Wiering, "Comparing approaches to the similarity of musical chord sequences," in *Exploring Music Contents: 7th International Symposium, CMMR 2010, Málaga, Spain, June 21-24, 2010. Revised Papers 7*. Springer, 2011, pp. 242–258.

[27] D. Bountouridis, H. V. Koops, F. Wiering, and R. C. Veltkamp, "A data-driven approach to chord similarity and chord mutability," in *2016 IEEE Second International Conference on Multimedia Big Data (BigMM)*. IEEE, 2016, pp. 275–278.

[28] K. Lemström and A. Pienimäki, "On comparing edit distance and geometric frameworks in content-based retrieval of symbolically encoded polyphonic music," *Musicae Scientiae*, vol. 11, no. 1_suppl, pp. 135–152, 2007.

[29] J. Lee, N. J. Bryan, J. Salamon, Z. Jin, and J. Nam, "Disentangled multidimensional metric learning for music similarity," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6–10.

[30] S. McAdams, S. Vieillard, O. Houix, and R. Reynolds, "Perception of musical similarity among contemporary thematic materials in two instrumentations," *Music Perception*, vol. 22, no. 2, pp. 207–237, 2004.

[31] E. Cambouropoulos, "How similar is similar?" *Musicae Scientiae*, vol. 13, no. 1_suppl, pp. 7–24, 2009.

[32] C. Finkensiep and M. A. Rohrmeier, "Modeling and inferring proto-voice structure in free polyphony," in *International Society for Music Information Retrieval Conference (ISMIR 2021)*, 2021, pp. 189–196.

[33] 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.

[34] D. Huron, "Music information processing using the humdrum toolkit: Concepts, examples, and lessons," *Computer Music Journal*, vol. 26, no. 2, pp. 11–26, 2002.

[35] M. S. Cuthbert and C. Ariza, "music21: A toolkit for computer-aided musicology and symbolic music data," in *International Society for Music Information Retrieval Conference (ISMIR 2010)*, 2010, pp. 637–642.

[36] A. Volk, W. B. de Haas, and P. Van Kranenburg, "Towards modelling variation in music as foundation for similarity," in *Proceedings of the 12th International Conference on Music Perception and Cognition and the 8th Triennial Conference of the European Society for the Cognitive Sciences of Music*. School of Music Studies, Aristotle University of Thessaloniki, 2012, pp. 1085–1094.

[37] W. E. Caplin, *Analyzing Classical Form: An approach for the classroom*. Oxford University Press, 2013.

[38] E. De Montalembert and C. Abromont, *Guide des genres de la musique occidentales*. Paris: Fayard Henry Lemoine, 2010.

[39] 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 *International Society for Music Information Retrieval Conference (ISMIR 2015)*, 2015.

[40] Y. Ni, M. McVicar, R. Santos-Rodriguez, and T. De Bie, "Understanding effects of subjectivity in measuring chord estimation accuracy," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 12, pp. 2607–2615, 2013.

INTRODUCING DiMCAT FOR PROCESSING AND ANALYZING NOTATED MUSIC ON A VERY LARGE SCALE

Johannes Hentschel[1] **Andrew McLeod**[2] **Yannis Rammos**[1]
Martin Rohrmeier[1]
[1] Digital and Cognitive Musicology Lab, École Polytechnique Fédérale de Lausanne, Switzerland
[2] Fraunhofer IDMT, Ilmenau, Germany
johannes.hentschel@epfl.ch

ABSTRACT

As corpora of digital musical scores continue to grow, the need for research tools capable of manipulating such data efficiently, with an intuitive interface, and support for a diversity of file formats, becomes increasingly pressing. In response, this paper introduces the Digital Musicology Corpus Analysis Toolkit (DiMCAT), a Python library for processing large corpora of digitally encoded musical scores. Equally aimed at music-analytical corpus studies, MIR, and machine-learning research, DiMCAT performs common data transformations and analyses using dataframes. Dataframes reduce the inherent complexity of atomic score contents (e.g., notes), larger score entities (e.g., measures), and abstractions (e.g., chord symbols) into easily manipulable computational structures, whose vectorized operations scale to large quantities of musical material. The design of DiMCAT's API prioritizes computational speed and ease of use, thus aiming to cater to machine-learning practitioners and musicologists alike.

1. INTRODUCTION

Given the proliferation of large corpora of digital scores (e.g., [1–4]), the computational challenges of analyzing symbolically encoded staff notation loom large in Digital Musicology and MIR. In principle, any symbolic music encoding is equally amenable to algorithmic processing, to the extent that it is consistent and comprehensive. In practice, however, the *visual* efficiency of staff notation—which conglomerates tonal, rhythmic, metric, articulatory, and other musical parameters in context-dependent and position-dependent symbols—is inversely related to its *computational* efficiency. Analyzing large collections of digital scores is thus hindered not only by the sheer volume of data involved, but also by the intrinsic complexity of the representations comprising musical structures. [5, 6]

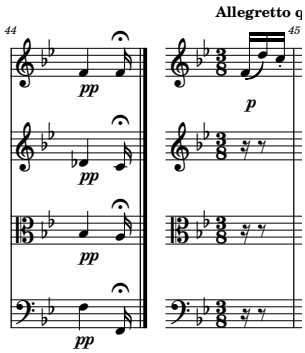
To address such challenges, we present the Digital Musicology Corpus Analysis Toolkit (DiMCAT), which uses dataframes [7–9] to disentangle pertinent score features within tabular representations, providing an interface for processing and analyzing large collections of dataframe-structured score data. DiMCAT supports MusicXML, MEI, Humdrum, and MuseScore (see Section 3.1), among other formats, and provides an expandable range of music analysis functionalities, including feature extraction, similarity analysis, and visualization. Addressed to Digital Musicology and MIR communities alike, its purpose is to provide a user-friendly interface for "distant-reading" staff-notated score corpora, and for utilizing score data in machine-learning pipelines. Efficiency at scale was among our primary design goals, an aspect which is only growing in importance as corpus sizes have continued to grow (e.g., [3]), with scores, rather than MIDI encodings, increasingly used to train large computational models (e.g., [10]).

In this paper, we describe the design and implementation of DiMCAT and argue for its usefulness through trials with various corpora. First, in Section 2, we outline a rationale for the representation of staff notation as dataframes, and for the underlying data-relational mindset. Section 3 presents the library design from a user's perspective, covering such topics as data loading (Section 3.1) and the slice/-group/analysis pipeline (Section 3.2). Evidence of ease-of-use in musicological research is provided in Section 4, and a comparison to extant libraries is made in Section 5.

2. UTILIZING DATAFRAMES TO REPRESENT SCORES

Dataframes were first introduced in 1990 as part of the statistical programming language S [7] and later ported to its descendant R [11]. Since then, they have become ubiquitous in the data science and machine learning communities, with a multitude of supplementary frameworks released across the spectrum of programming languages, often aiming to overcome performance problems associated with large dataframes (e.g., modin [12]). The wide adoption of dataframes can be attributed to their versatility, convenience, and operational principles, which resemble those of relational databases, spreadsheets, and nested arrays [9, 11–13]. DiMCAT encodes all score information within dataframe objects provided by either pandas or modin, with support for additional libraries (which we refer to as "backends") planned in future versions.

© J. Hentschel, A. McLeod, Y. Rammos, and M. Rohrmeier. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** J. Hentschel, A. McLeod, Y. Rammos, and M. Rohrmeier, "Introducing DiMCAT for processing and analyzing notated music on a very large scale", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.



interval	qstamp	mc	mn	mc_onset	mn_onset	duration	duration_q	timesig	staff	voice	name	midi	tpc	octave
	fraction	*int*	*int*	*fraction*	*fraction*	*fraction*	*float*	*str*	*int*	*int*	*str*	*int*	*int*	*int*
[87.0, 87.25)	87	44	44	1/4	1/4	1/16	0.25	2/4	4	1	F2	41	-1	2
[87.0, 87.25)	87	44	44	1/4	1/4	1/16	0.25	2/4	3	1	A3	57	3	3
[87.0, 87.25)	87	44	44	1/4	1/4	1/16	0.25	2/4	2	1	C4	60	0	4
[87.0, 87.25)	87	44	44	1/4	1/4	1/16	0.25	2/4	1	1	F4	65	-1	4
[87.25, 87.5)	349/4	45	44	0	3/16	1/16	0.25	3/8	1	1	F4	65	-1	4
[87.25, 87.5)	349/4	45	44	0	3/16	1/16	0.25	3/8	2	1				
[87.25, 87.5)	349/4	45	44	0	3/16	1/16	0.25	3/8	3	1				
[87.25, 87.5)	349/4	45	44	0	3/16	1/16	0.25	3/8	4	1				
[87.5, 87.75)	175/2	45	44	1/16	1/4	1/16	0.25	3/8	1	1	D5	74	2	5
[87.5, 88.0)	175/2	45	44	1/16	1/4	1/8	0.5	3/8	2	1				
[87.5, 88.0)	175/2	45	44	1/16	1/4	1/8	0.5	3/8	3	1				
[87.5, 88.0)	175/2	45	44	1/16	1/4	1/8	0.5	3/8	4	1				
[87.75, 88.0)	351/4	45	44	1/8	5/16	1/16	0.25	3/8	1	1	C5	72	0	5

Table 1: Ludwig van Beethoven, *String quartet op. 18/6*, 4th movement ("La Malinconia"), measure number (**mn**) 44. The measure contains the section break after the slow introduction and is composed of two incomplete measure units with counts (**mc**) 44 and 45. The new 3/8 time signature (**timesig**) of the latter is introduced by a 3/16 upbeat, mathematically completing the 2/4 meter of the former. The dataframe represents notes and rests from beat 2 onwards. Its index (bold values on the left) comprises left-closed, right-open intervals which express the start and end points of each event on the score's timeline, measured in quarter notes. Each column has a name (**bold**) and a data type (*italic*). The first eight columns contain temporal information (see Section 2.1). The columns **staff** and **voice** determine a notational layer. The last four columns express pitch-related information (**tpc** is tonal pitch class, expressed as the distance from C measured in perfect fifths) and are empty for rows representing rests. Special columns are omitted (e.g., ties, tremolos, or grace notes).

2.1 Representing staff notation as dataframes

Most encoding standards symbolically represent staff notation in hierarchical fashion. This includes most non-XML plaintext formats—at least those capable of encoding multiple staves, such as Lilypond, ABC, or Humdrum's **kern—as well as XML-based standards. Table 2 shows a selection of tags in order of hierarchical nesting, from outermost to innermost, for three common XML-based standards. The table reveals that these standards recognize almost the same types of score elements, albeit located at different levels within the document tree. Among these elements are staves, measures, textural layers ('voices'), chords (understood as groups of notes sharing a stem) and, finally, notes. For certain features, such as tempo and dynamic markings, the choice of hierarchical anchor is to some extent arbitrary: a tempo marking, for instance, might be attached to a specific measure or to a chord within that measure. `DiMCAT`'s approach to the unified modeling of diverse hierarchical representations consists in traversing them and grouping score elements of the same type in the same dataframe. This obviates mapping the particularities of each standard into a common score model, a process which would either inherit a degree of arbitrariness, or resort to error-prone estimations in order to eradicate it. [1]

`DiMCAT` disentangles the underlying score hierarchy by grouping elements in five distinct categories, which we refer to as "facets". These are:

- notes and rests ("events", including ties, tremolos, grace notes, etc.)
- performance details ("control events"; tempo, dynamics, slurs, lyrics, articulation, etc.);
- measures ("flow control"; measure durations, staves, repeat indications, *fine*, etc.);

MuseScore	MEI	musicXML	
<Score>	<music> ···	<score-partwise>	<score-timewise>
<Staff>	(<part> ···)	<part>	<measure>
<Measure>	<measure> ···	<measure>	<part>
<Voice>	<staff> ···	<note>	
<Chord>	<layer> ···	<staff>, <voice>	
<Note>	(<chord> ···)		
	<note>		

Table 2: Synopsis of XML tag hierarchies in three widespread XML-based score models. Models differ mainly in the placement and naming of score elements (<layer> being equivalent to <voice>). In the MEI column, ellipses (···) suggest that any number of hierarchical levels may be nested, and parentheses mark optional layers. MusicXML has two distinct organizational strategies (partwise or scorewise), which converge at the note level.

- analytical annotations ("labels"; chord changes, form labels, algorithmic outputs, etc.);
- metadata. [2]

A facet is the raw, original representation of a category of score elements from which specific, homogeneous "features" can be derived. In its simplest form, a feature is a subset of a facet in terms of rows and/or columns. For example, the `NotesAndRests` facet (shown in Table 1) comprises the `Rests` feature from which all rows and columns about notes have been removed. Other features offer variants requiring simple transformations. For instance, the `Notes` feature may be requested with tied note heads fused into single note events, with metrical weights added, or with pitches expressed as scale degrees (for an example, see Listing 2). Other features require more substantial computational analysis on a set of features or

[1] Such a mapping, employed for example by the `music21` score processing library [14], is rather suitable when a complete model of the score needs to be maintained for further processing.

[2] Note that visual details such as beaming are not loaded by default whenever they are deemed irrelevant for a distant-listening setting.

facets, and necessitate the invocation of an `Analyzer` (see Section 3.2.3).

Our approach thus projects different hierarchical score representations into a paradigm similar to that of relational databases. Structural relations previously expressed by the underlying score hierarchy are now expressed via IDs (for example, the columns 'staff', 'voice', and 'mc' in Table 1). In addition, all objects (except metadata) are unambiguously located on the score's musical timeline by means of timestamps. As Table 1 shows, each facet and feature includes five columns expressing timestamps in three partially redundant ways. Timestamps expressed by means of 'mc' (the strict count of measure-like units from the beginning of the piece, regardless of their actual length or displayed measure number), along with 'mc_onset' (the location within a measure-like unit, represented as a fraction of a whole note) serve a crucial function. Given the actual durations of the measure-like units, 'mc' and 'mc_onset' determine 'qstamp', an object's offset from the beginning of the piece in quarter notes. In addition, `DiMCAT` provides 'mn', the measure numbers actually found in score engravings, which are in principle non-unique and based on longstanding editorial conventions [15]. Analogously to 'mc_onset', these units warrant 'mn_onset' positions, required for computing metrical weights consistent with the respective meter (in the column 'timesig').

2.2 Operations on `DimcatResource` objects

To facilitate the processing and analysis of potentially large collections of notated music, `DiMCAT` aggregates facets (as well as features or analysis results) drawn from multiple pieces in a single dataframe, a `DimcatResource`. This approach enables vectorized operations on entire datasets, thus achieving higher performance in comparison to an equivalent sequence of single-dataframe operations. For additional speed when using very large corpora, `DiMCAT` can delegate dataframe operations to a distributed-computing backend such as `modin`, which allows for automatic partitioning and parallel processing [12]. `DimcatResources` natively serialize into ZIP archives accompanied by a Frictionless descriptor file [16] allowing for type-safe data validation and loading with external tools. Furthermore, the "frictionless" design allows `DiMCAT` to treat a descriptor file with its included metadata (column descriptions, file genesis, versioning information) as if it was the described resource itself, and to load the actual data into memory no earlier than required. The loaders described in the following section have the purpose of pre-processing the data to be analyzed, and storing it in a self-contained format that can also be easily served on the web.

3. LIBRARY DESIGN

This section describes `DiMCAT`'s design and API, [3] which parallel familiar routines of musicological research: constructing a corpus (loading and filtering, Section 3.1), organizing relevant corpus data (slicing and grouping), and running algorithms on this selection (analyzing, and plotting (Section 3.2).

3.1 Loading Data

`DiMCAT` defines loaders which parse and store score data for a variety of symbolic encoding standards with the aid of external libraries. [4] This is typically achieved by discovering the relevant files on disk or (from the web) and producing the homogeneous representation (the dataframes presented in Section 2.1) in parallelized fashion, while also compressing and storing the loaded data on disk for later use. Once data has been pre-processed and stored along with its metadata, `DiMCAT`'s default loader is capable of determining which score features are present, and of "lazily" loading them into memory whenever needed for processing. Apart from decreasing the memory footprint, this principle makes it possible to verify, before proceeding, whether the features required for a processing pipeline are actually available (see below). The extracted `Facet` objects remain by design as faithful as possible to the original data in terms of presence and naming of detected elements. Names and types of facet fields are standardized only in relation to the above-mentioned timeline columns, which are necessary for alignment. `Feature` objects, on the other hand, are comprehensively standardized upon extraction to guarantee type safety. Less specialized analyzers such as `Counters` also allow for the processing of `Facets`, and users who frequently work with custom `Features` drawn from nonstandard elements may contribute appropriate extensions to the codebase in the spirit of community-driven development.

Once loaded, the data is represented internally by the `Dataset` class and its various subclasses (see the complete documentation for details). `Dataset` objects are `DiMCAT`'s main drivers and the object type users interact with the most. They grant centralized access to all available dataframes (`Facets`, `Features`, and `Results`), depending on the current stage of computation. These objects in turn enable type-specific transformations and visualizations.

3.2 The Analysis Pipeline

Conceptually, every action performed by `DiMCAT` is a sequence of `PipelineStep` objects which, having been chained together, accept a `Dataset` object, perform a transformation or analysis on it, and return a new data object. In practice, this chaining is not entirely arbitrary, since some computations require data in specific formats (for example, the `CrossEntropy` analyzer requires equal-shaped probability vectors). All pipeline steps can be expressed as, and instantiated from, associative arrays of type `DimcatConfig` which are stored together with a `Dataset`'s descriptor to make the pipeline generation reproducible.

[3] The complete API and documentation can be found at `https://github.com/DCMLab/dimcat`. In addition to pydocs, we provide Jupyter Notebook–based interactive tutorials.

[4] These currently include `ms3` [17] and `music21` [14], with `Verovio` [18] planned to be added in the future.

Each step in a `Pipeline` is fundamentally an instance of one of the following three classes [5]: `Slicers` accept data and partition it into chunks of various sizes—for example, sections between repeat signs, segments under the same guitar chord, or 8th-note-long slices. Slices never cross the boundaries of a piece of music. `Groupers` accept data and group it in formally specified categories—for example, segments with the same chord label, or pieces composed in the same decade. Unlike slices, groups often contain information from across the corpus. `Analyzers` are the heart of the library, performing the actual computation once the data has been sliced and grouped.

Many questions in music corpus studies involve comparisons between groups with a degree of commonality, e.g., between groups of pieces by the same composer, or of segments such as sonata development sections [19]. By combining slicers, groupers and analyzers, music researchers will find in `DiMCAT` an intuitive language for addressing pressing questions in the field.

3.2.1 Slicers

`Slicer` objects invoke one particular feature to compute segmentation boundaries. For example, a `NoteSlicer` invokes the `Notes` feature and stores its timestamps (e.g., note onset positions) as slice markers within the resulting `SlicedDataset`, interpreting them as time intervals. The newly returned dataset will slice any facet or feature subsequently requested, inserting an additional index level or column for slice boundaries (any element spanning over a slice boundary will be split or duplicated). In principle, any feature (e.g., double bar lines, dynamic indications, or the results of a key finder) can serve as the slicing criterion. The properties of the feature used as criterion can then be used for grouping the resulting slices (e.g., slicing a dataset using the results of a key-finding algorithm enables the subsequent grouping of slices by mode; see the following section).

3.2.2 Groupers

Applying a `Grouper` to a dataset is tantamount to binning pieces or slices based on a membership criterion. As a result, any facet or feature requested from a `GroupedDataset` is provided with a prepended index level of group identifiers. This enables both choosing a larger unit of analysis (by analyzing entire groups rather than each contained piece or slice, see the following section) and comparing groups of analysis results (for an example, see Section 4.2).

Frequently used groupers include the `CorpusGrouper`, the `StaffGrouper`, and the `ModeGrouper` (grouping pieces, events, and key slices, respectively). Groupers may also use metadata as criterion: for instance, the `YearGrouper` groups pieces based on their composition dates. Grouping is a computationally cheap operation because it is performed using dataframe indices.

[5] That is without considering auxiliary pipeline steps such as `Writers` which never result in a different dataset type.

3.2.3 Analyzers

`Analyzers` are at the heart of the `DiMCAT` library. Based on its configuration, an analyzer will take one particular or all available features from the `Dataset`, perform the analysis on the minimal unit provided by the `Dataset` (slice or piece), and return an `AnalyzedDataset`. Results can be `Feature` objects (with timestamps) or `Result` objects (without timestamps), both of which are `DimcatResources` (see Section 2.2) and provide suitable methods for retrieving, displaying, transforming, and plotting analysis results. In addition, they allow the combination of piece or piece-slice analysis results into those corresponding to higher units of analysis, e.g., piece results into group results. This makes it possible, by applying several groupers to the same `AnalyzedDataset`, to regroup and recombine individual results. `DiMCAT` currently uses the `plotly` library for creating interactive plots and provides reasonable (but non-binding) defaults for combining grouped results in one figure. The main types of analyzers are `Counters`, `Comparisons` and `ClusterAnalyzers`, `Transformations`, and `RangeAnalyzers`.

The base `Analyzer` class is designed to be easily extensible; additional analyzers can be created by the community without knowledge of the deeper layers of the code. Contributors only need to understand the structure of the features that the new analyzer accepts as input, and select or implement the appropriate result type. Thereafter they implement the new object's serialization `Schema` [6] and one of the methods that performs the actual analysis on a slice-, piece-, or group-specific dataframe. The method `combine()`, used for aggregating two result objects into one—for example, by adding result vectors—only needs to be implemented if no superclass is available to inherit it from. Optional methods include `check()` (for rejecting a dataset or feature if it doesn't fulfill certain criteria), `pre_process()` (for performing analyzer-specific feature transformations), and `post_process()` (for cleaning up the results object, for example by filling in missing values). A new analyzer constructed in this fashion is guaranteed to work with `DiMCAT`'s pipeline architecture.

The basic `Counter` counts the number of rows of any facet or feature (e.g., notes or chord labels). More versatile counters aggregate counts or durations based on the values contained in a given column (e.g., pitch classes), value combinations between several columns (e.g., pitch class–duration pairs), or n-grams (e.g., pairs of successive dynamic indications). Results can be transformed (e.g., by normalizing), various properties can be calculated and returned (e.g., the distributions' entropies), and plots can be generated.

`Comparison` analyzers perform pairwise comparisons on the slices, pieces, or groups represented by a given feature or result, such as the sliding-window autocorrelation of a feature's inter-onset-intervals, the Jaccard similarity between chord vocabularies, or the cross entropy between key profiles. They typically store their result as a

[6] See details in the documentation.

confusion matrix, plotted by default as a heatmap. The results of comparisons lend themselves to subsequent application of `ClusterAnalyzers`, which use common algorithms such as k-means to compute groups that can reveal relations between the features under comparison [20, 21].

`Transformations` apply a function or fit a model in order to translate a feature into a different representation. Examples include analyzers that fit a Gaussian mixture model to a distribution, tokenize pitch events for use in a neural network, or transform pitch-class profiles into Fourier coefficients (as demonstrated in Section 4.1).

`RangeAnalyzers` are useful in cases where only minima and maxima (or the range) of numerical features are relevant. Examples include the line-of-fifths segment covered by a pitch class distribution [22] or the historical timespan covered by a dataset based on composition dates.

Finally, there are many specific analyzers, such as the `PitchClassVectors` analyzer featured in Section 4.1; they perform an analysis or transformation (here, aggregating durations) on one particular feature (here, `Notes`) under a range of specific configuration values (here, for example, type and format of pitch classes). A full list of analyzers is available in the documentation.

4. EXAMPLES

In this section, we present a few examples of musicological questions that can be easily answered using `DiMCAT` (provided the requisite data is available).

4.1 Fourier analysis of pitch class vectors

```
D = Dataset.load("debussy_piano")
D_analyzed = Pipeline(
    [WindowSlicer(quarters_per_slice=1.0),
     PitchClassVectors(),
     DiscreteFourierTransform()]
).process(D)
df = D_analyzed.get_result()
df.sample(5)
```

Listing 1: Pipeline for slicing a dataset by quarter-note windows, computing pitch class vectors and applying the Discrete Fourier Transform.

The Discrete Fourier Transform has seen frequent applications to musical structures, in particular pitch-class sets [23–28]. It belongs in a broader class of techniques which require, in our terms, a "slicing" of the score, such as a chordal reduction or a fixed-window segmentation. For example, as part of a corpus study using the Discrete Fourier Transform [29], `DiMCAT` was used to create enharmonic pitch class vectors for all 2-hand piano compositions by Claude Debussy. Listing 1 demonstrates the simplicity with which this analysis can be expressed as a `DiMCAT` pipeline. In the first line, the data is loaded from a local directory. Then the pipeline is created and processed. In the pipeline, a slicer first slices all pieces at every quarter note, then an analyser creates vectors of aggregated pitch class durations for each slice. Finally, the DFT analyser is run on each vector and the result obtained. A sample of the results, with coefficients 0 through 6 given as complex numbers, is shown in Table 3.

4.2 Evaluating key segments

```
D = Dataset.load("dcml_corpora.datapackage.json")
D_sliced = Pipeline(
    [KeySlicer(),
     ModeGrouper()]
).process(D)
F = DimcatConfig("Notes", format=SCALE_DEGREES)
D_sliced.get_feature(F).plot_grouped() # plot 1
D_grouped = CorpusGrouper().process(D_sliced)
D_grouped.plot_grouped() # plot 2
```

Listing 2: Plotting common dataset transformations (plotting parameters omitted). Plots shown in Figure 1.

Given a score dataset with local-key annotations created by human analysts or an automatic key finder, a researcher might wonder how key segments in the major and minor mode are distributed over the corpora contained in the dataset, and how the tonal pitch-class profiles compare between the two modes. This second example relies on a dataset that includes key annotations [7] and demonstrates the power and ease-of-use of `DiMCAT` pipelines, even without using any analyzers. The `KeySlicer` used in Listing 2 is set up by default to slice the dataset by annotated modulations, and warns the user about pieces for which no local key information is available. The pipeline proceeds by applying the `ModeGrouper` to create one group per mode which, in the present dataset, amounts to a minor-key and a major-key group. Requesting the `Notes` feature from a `Dataset` processed in this fashion, we may retrieve and plot a representation that reflects the grouping, as shown in the upper bar chart in Figure 1. The lower plot demonstrates that the slicing criterion itself may also provide meaningful insights into a dataset. It can be produced by applying a `CorpusGrouper` to the processed `Dataset` and plotting the groups. (Without the corpus grouper, the plot would show the major-minor ratio for the entire dataset—that is, the result of the mode grouper).

5. COMPARISON WITH OTHER LIBRARIES

Other analysis libraries also lend themselves to the analysis of datasets of symbolic music encodings. In this section we compare `DiMCAT` to other open-source libraries which maintain note names (pitch spelling) and support multiple staffs and analytical annotations. Several among them likewise utilize dataframes.

The Humdrum Toolkit was one of the first frameworks for computer-aided music analysis, and is still used for the analysis of Humdrum and kern files across the range of programming languages to which it has been ported. The R package `humdrumR` [30] ports the Humdrum Toolkit into R. While it provides support for computationally efficient dataframes, and includes R's inherent plotting capabilities, like Humdrum itself it cannot import more modern, and arguably more common, symbolic-encoding formats such as musicXML without the use of error-prone converters.

[7] https://github.com/DCMLab/dcml_corpora

corpus	fname	1.0q_slice	0	1	2	3	4	5	6
debussy_other_piano_pieces	**l068_reverie**	**[351.0, 352.0)**	2.00+0.00j	0.32-0.18j	-0.50-0.87j	-1.50+0.50j	-1.00+0.00j	1.18+0.68j	1.00+0.00j
debussy_childrens_corner	**l113-03_childrens_serenade**	**[27.0, 28.0)**	2.25+0.00j	0.92-0.47j	0.37-0.22j	0.75-0.50j	-1.12-0.65j	-1.67-0.03j	-0.75+0.00j
debussy_preludes	**l123-12_preludes_feux**	**[176.0, 177.0)**	6.62+0.00j	0.53-1.57j	-1.94-0.11j	0.00+4.12j	2.69-2.27j	2.47-3.30j	-2.12+0.00j
debussy_etudes	**l136-04_etudes_sixtes**	**[53.5, 54.5)**	4.00+0.00j	-0.12+0.37j	-0.75+0.00j	-0.75-0.25j	-1.25+0.87j	1.62-1.37j	1.50+0.00j
debussy_deux_arabesques	**l066-02_arabesques_deuxieme**	**[137.0, 138.0)**	4.00+0.00j	0.25-0.30j	-0.25-1.30j	-2.00-1.00j	1.75+1.30j	0.25+2.30j	2.00+0.00j

Table 3: Sample rows from a dataframe containing the seven first DFT coefficients gained from quarter-note-window pitch class vectors. The first three columns represent a multi-index indicating corpus, file name, and slice interval (expressed as `qstamp`, see Table 1); they make it possible to trace the result back to the score.

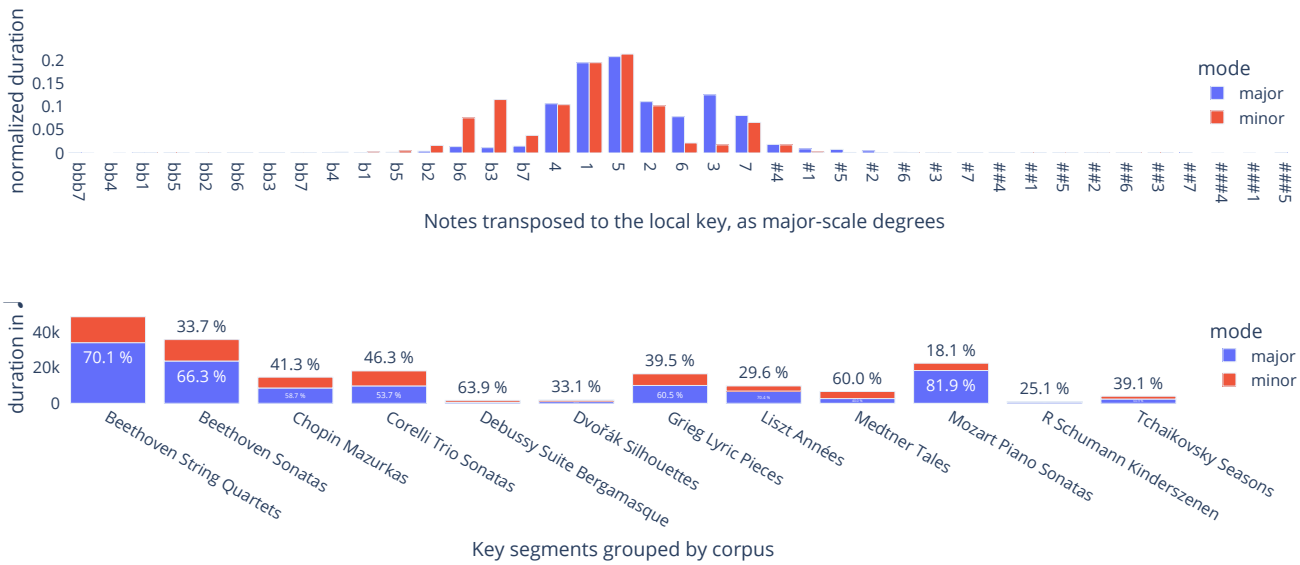


Figure 1: The two plots produced by the code shown in Listing 2.

`Music21` [14] is a large Python library capable of importing all relevant music formats, transforming them into a comprehensive hierarchical model of the score. Relying on an elaborate object model, it provides methods for creating and manipulating the elements of a music score. However, its design renders it computationally demanding [30, 31] for large corpora, and it provides only few methods designed specifically for corpus analysis.

Several Python libraries follow a similar approach to DiMCAT's, analyzing and making available score information in the form of dataframes. These include the `VIS-framework` [32] and `CRIM intervals` [33] (both focusing on intervallic successions and sonorities), `CAMAT` [31] (basic pitch statistics), and `musif` [34] (with a focus on global features of entire scores). Among them, only [31] introduces its own score parser (for MusicXML), with the remaining ones invoking `music21`. [34] also includes the MuseScore parser `ms3` [17] and therefore exposes an architecture that is as easily extensible as ours.

Although `DiMCAT` can, in principle, provide any algorithm that operates on successions or sets of pitch events, its focus on "distant listening" makes it less suited for close-reading studies than some of the aforementioned alternatives. Indeed, its distinguishing feature is the newly introduced *Slice-Group-Analyze* paradigm, designed for inquiries in which the corpus, rather than the individual score, is the primary research object. To this end, `DiMCAT` provides mechanisms for studying the statistical properties of potentially very large amounts of musical material by iteratively applying sequences of segmentation and grouping algorithms with a high degree of combinatorial freedom. From this point of view, the "slice" serves as an additional operational level between "note" [31–33] and "piece" [34].

6. CONCLUSION

In this paper we have introduced `DiMCAT`, a Python library capable of parsing, transforming, and analyzing annotated music score data in a range of symbolic-encoding formats, and to do so efficiently at scale. The library stores data as dataframes, a ubiquitous structure in the fields of digital humanities and data science. `DiMCAT` emphasizes traceability (results can reliably lead to the original score elements) and reproducibility (version identifiers are systematically applied to code and data). Thanks to an interface that masks its inner workings, the functionality of the library is usable and extensible by musicologists with limited programming experience.

`DiMCAT` is released under the GPL-3.0-or-later License, and we intend to continue adding further music analyzers, inviting feedback, requests, and contributions from the community.

7. ACKNOWLEDGEMENTS

This research was supported by the Swiss National Science Foundation within the project "Distant Listening – The Development of Harmony over Three Centuries (1700–2000)" (Grant no. 182811). This project is being conducted at the Latour Chair in Digital and Cognitive Musicology, generously funded by Mr. Claude Latour.

8. REFERENCES

[1] C. S. Sapp, "Online database of scores in the humdrum file format," in *Proceedings of the Proceedings of the 9th International Society for Music Information Retrieval Conference (ISMIR)*, Philadelphia, 2008.

[2] M. Gotham, P. Jonas, B. Bower, W. Bosworth, D. Rootham, and L. VanHandel, "Scores of Scores: An OpenScore project to encode and share sheet music," in *Proceedings of the 5th International Conference on Digital Libraries for Musicology - DLfM '18*. Paris, France: ACM Press, 2018, pp. 87–95.

[3] F. Foscarin, A. McLeod, P. Rigaux, F. Jacquemard, and M. Sakai, "ASAP: A dataset of aligned scores and perfor mances for piano transcription," in *Proceedings of the 21st International Society for Music Information Retrieval Conference (ISMIR)*, 2020.

[4] J. Hentschel, Y. Rammos, M. Neuwirth, F. C. Moss, and M. Rohrmeier, "An annotated corpus of tonal piano music from the long 19th century," *Empirical Musicology Review*, forthcoming.

[5] J. Devaney, "Using note-level music encodings to facilitate interdisciplinary research on human engagement with music," *Transactions of the International Society for Music Information Retrieval*, vol. 3, no. 1, pp. 205–217, Oct. 2020.

[6] F. Foscarin, P. Rigaux, and V. Thion, "Data quality assessment in digital score libraries: The GioQoso Project," *International Journal on Digital Libraries*, vol. 22, no. 2, pp. 159–173, Jun. 2021.

[7] J. Chambers, T. Hastie, and D. Pregibon, "Statistical models in S," in *Proceedings in Computational Statistics*, K. Momirović and V. Mildner, Eds. Heidelberg: Physica-Verlag HD, 1990, pp. 317–321.

[8] W. McKinney, "Data structures for statistical computing in python," in *Proceedings of the 9th Python in Science Conference*, Austin, Texas, 2010, pp. 56–61.

[9] D. Petersohn, "Dataframe systems: Theory, architecture, and implementation," Ph.D. dissertation, University of California, Berkeley, 2021.

[10] A. Mcleod and M. A. Rohrmeier, "A modular system for the harmonic analysis of musical scores using a large vocabulary," in *Proceedings of the 22nd International Society for Music Information Retrieval Conference (ISMIR)*. Online: ISMIR, Nov. 2021, pp. 435–442.

[11] D. Petersohn, S. Macke, D. Xin, W. Ma, D. Lee, X. Mo, J. E. Gonzalez, J. M. Hellerstein, A. D. Joseph, and A. Parameswaran, "Towards scalable dataframe systems," *Proceedings of the VLDB Endowment*, vol. 13, no. 12, pp. 2033–2046, Aug. 2020.

[12] D. Petersohn, D. Tang, R. Durrani, A. Melik-Adamyan, J. E. Gonzalez, A. D. Joseph, and A. G. Parameswaran, "Flexible rule-based decomposition and metadata independence in modin: A parallel dataframe system," *Proceedings of the VLDB Endowment*, vol. 15, no. 3, pp. 739–751, Nov. 2021.

[13] Y. Wu, "Is a dataframe just a table?" in *10th Workshop on Evaluation and Usability of Programming Languages and Tools (PLATEAU 2019)*, ser. OpenAccess Series in Informatics (OASIcs), S. Chasins, E. L. Glassman, and J. Sunshine, Eds., vol. 76. Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2020, pp. 6:1–6:10.

[14] M. S. Cuthbert, "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, pp. 637–642.

[15] E. Gould, *Behind Bars: The Definitive Guide to Music Notation*. London: Faber Music, 2011.

[16] D. Fowler, J. Barratt, and P. Walsh, "Frictionless data: Making research data quality visible," *International Journal of Digital Curation*, vol. 12, no. 2, pp. 274–285, May 2018.

[17] J. Hentschel and M. Rohrmeier, "Ms3: A parser for MuseScore files, serving as data factory for annotated music corpora," *Journal of Open Source Software*, 2023.

[18] L. Pugin, R. Zitellini, and P. Roland, "Verovio: A library for engraving MEI music notation into SVG," in *PugiProceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR)*, 2014, pp. 107–112.

[19] C. White, *The Music in the Data: Corpus Analysis, Music Analysis, and Tonal Traditions*, 1st ed. New York: Routledge, Nov. 2022.

[20] R. Cilibrasi, P. Vitanyi, and R. de Wolf, "Algorithmic clustering of music," *arXiv:cs/0303025*, Mar. 2003.

[21] E. Anzuoni, S. Ayhan, F. Dutto, A. McLeod, F. C. Moss, and M. Rohrmeier, "A historical analysis of harmonic progressions using chord embeddings," in *Sound and Music Computing Conference (SMC)*, 2021, pp. 284–291.

[22] F. C. Moss, M. Neuwirth, and M. Rohrmeier, "The line of fifths and the co-evolution of tonal pitch-classes," *Journal of Mathematics and Music*, pp. 1–25, Mar. 2022.

[23] D. Lewin, "Re: Intervallic relations between two collections of notes," *Journal of Music Theory*, vol. 3, no. 2, pp. 298–301, Nov. 1959.

[24] I. Quinn, "A unified theory of chord quality in equal temperaments," Ph.D. dissertation, Eastman School of Music, Rochester, New York, 2004.

[25] D. Tymoczko, "Set-class similarity, voice eading, and the Fourier transform," *Journal of Music Theory*, vol. 52, no. 2, pp. 251–272, Sep. 2008.

[26] J. Yust, "Applications of DFT to the theory of twentieth-century harmony," in *Mathematics and Computation in Music*, T. Collins, D. Meredith, and A. Volk, Eds. Cham: Springer International Publishing, 2015, vol. 9110, pp. 207–218.

[27] E. Amiot, *Music through Fourier Space*, ser. Computational Music Science. Cham: Springer International Publishing, 2016.

[28] J. D. Harding, "Applications of the Discrete Fourier Transform to music analysis," Ph.D. dissertation, Florida State University, 2021.

[29] S. Laneve, L. Schaerf, G. Cecchetti, J. Hentschel, and M. Rohrmeier, "The diachronic development of Debussy's musical style: A corpus study with Discrete Fourier Transform," *Humanities and Social Sciences Communications*, vol. 10, no. 1, p. 289, Jun. 2023.

[30] N. Condit-Schultz and C. Arthur, "humdrumR: A new take on an old approach to computational musicology," in *Proceedings of the 20th International Society for Music Information Retrieval Conference (ISMIR)*, Delft, 2019.

[31] E. Poliakov and C. R. Nadar, "CAMAT: Computer Assisted Music Analysis Toolkit," in *Proceedings of the Digital Music Research Network One-day Workshop (DMRN+16)*, 2021, p. 12.

[32] C. Antila and J. Cumming, "The VIS framework. Analyzing counterpoint in carge datasets," in *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR)*, 2014.

[33] R. Freedman, A. Morgan, Gould, O. Shostak, T. Dang, A. Janco, D. Russo-Batterham, E. Leon, and H. West, "CRIM intervals," 2023. [Online]. Available: https://github.com/HCDigitalScholarship/intervals

[34] A. Llorens, F. Simonetta, M. Serrano, and Á. Torrente, "Musif: A Python package for symbolic music feature extraction," in *Sound and Music Computing Conference (SMC)*. arXiv, Jul. 2023.

SEQUENCE-TO-SEQUENCE NETWORK TRAINING METHODS FOR AUTOMATIC GUITAR TRANSCRIPTION WITH TOKENIZED OUTPUTS

Sehun Kim
Nagoya University
kim.sehun@g.sp.m.is.nagoya-u.ac.jp

Kazuya Takeda
Nagoya University
kazuya.takeda@nagoya-u.jp

Tomoki Toda
Nagoya University
tomoki@icts.nagoya-u.ac.jp

ABSTRACT

We propose multiple methods for effectively training a sequence-to-sequence automatic guitar transcription model that uses tokenized music representation as an output. Our proposed method mainly consists of 1) a hybrid CTC-Attention model for sequence-to-sequence automatic guitar transcription that uses tokenized music representation, and 2) two data augmentation methods for training the model. Our proposed model is a generic encoder-decoder Transformer model but adopts multi-task learning with CTC from the encoder to speed up learning alignments between the output tokens and acoustic features. Our proposed data augmentation methods scale up the amount of training data by 1) creating bar overlap when splitting an excerpt to be used for network input, and 2) by utilizing MIDI-only data to synthetically create audio-MIDI pair data. We confirmed that 1) the proposed data augmentation methods were highly effective for training generic Transformer models that generate tokenized outputs, 2) our proposed hybrid CTC-Attention model outperforms conventional methods that transcribe guitar performance with tokens, and 3) the addition of multi-task learning with CTC in our proposed model is especially effective when there is an insufficient amount of training data.

1. INTRODUCTION

Automatic guitar transcription is a challenging task that has gained significant attention in the field of music information retrieval due to its potential applications in music analysis, performance evaluation, and transcription of music compositions. Despite recent advancements in the field, there are still several challenges that need to be addressed. One of the major challenges in automatic guitar transcription is the difficulty in extracting relevant features from the audio signal. The variations in timbre, pitch, and playing style make it difficult to distinguish individual notes accurately [1, 2].

Multiple methods exist for representing musical notation suitable for employment in a DNN framework. Two of the popular methods are pianoroll and tokenized musical representation. Figure 1 shows a visualization of the differences between pianoroll and tokenized music representations. Note that each can be converted from one to another. Pianoroll is a visual representation of music that uses a grid-like structure to display the timing, pitch, and duration of notes in a piece. Tokenized music representation is a type of symbolic music representation that breaks down a music signal into small, discrete tokens, which can be analyzed and processed as a sequence of tokens to extract relevant musical features such as pitch, duration, and timing. Recent studies have shown that using tokenized music representation over pianoroll can better learn the temporal dependency between different musical events [3]. Using tokenized music representation along with sequence-to-sequence models is particularly effective and have the potential to improve the performance of automatic music transcription model [1,4]. However, these models also face several challenges such as the lack of training data [1].

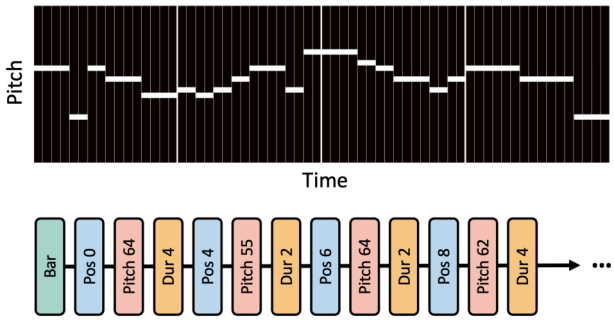


Figure 1. Examples of a pianoroll (upper) and tokenized music representation (lower).

The field of automatic speech recognition (ASR) has provided inspiration for improving automatic music transcription models, as both face similar challenges, including the need to extract relevant features and handle complex temporal and frequency relationships [5]. In the field of ASR, various techniques and models, such as connectionist temporal classification (CTC) [6], Transformer [7] models, data augmentation, transfer learning, and multi-task learning, have shown promising results [8, 9] and can potentially aid the performance of an automatic guitar transcription system.

CTC and attention are two popular techniques used in sequence-to-sequence models for various tasks. CTC is

© S. Kim, K. Takeda, and T. Toda. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** S. Kim, K. Takeda, and T. Toda, "Sequence-to-Sequence Network Training Methods for Automatic Guitar Transcription with Tokenized Outputs", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

an efficient method for training models with an unknown alignment between input and output sequences. It can handle variable-length input and output sequences. However, it does not explicitly model dependencies between input and output sequences [10]. Attention, on the other hand, allows the model to selectively focus on different parts of the input sequence, improving the accuracy of the model on tasks that require complex dependencies. However, attention mechanism is too flexible in the sense that it allows extremely non-sequential alignments, making it relatively difficult to train [8].

In an attempt to improve the performance of an ASR system, researchers have also explored hybrid models that combine CTC and attention mechanisms [8]. The authors reported that the addition of CTC solves the misalignment issues and improves robustness and achieves fast convergence.

Although models such as Conformer-Transformer have been successful in ASR tasks [11], these were not used in guitar transcription mainly due to a lack of training data available. To resolve this issue, we propose data augmentation techniques and a hybrid CTC-Attention model suited for a guitar transcription system [1] that utilizes tokenized music representation and show the effectiveness of the proposed methods. Our contributions are summarized as follows.

- We propose two data augmentation methods for training a sequence-to-sequence model that utilizes tokenized music representation.
- We propose a hybrid CTC-Attention model for automatic guitar transcription.
- We conduct experimental evaluations to confirm the effectiveness of our proposed methods and prove that both the data augmentation techniques and the proposed model enhance guitar transcription performance.

2. RELATED WORK

2.1 Automatic guitar transcription

There have been many successful automatic guitar transcription systems [1, 12–15]. Some of them are based on audio signal processing to estimate the tablature score from a guitar sound signal [12]. Also, approaches that employ probabilistic models have been proposed in some automatic guitar transcription tasks. In [14, 15], a two-step method was employed, where the first step involves determining the pitch of each played note, and the second step involves computing the optimal finger positioning by combining the estimated pitch with physical limitations of feasible fingerings. Since this approach processes information in a sequential manner, information cannot flow from downstream components to upstream ones, making it difficult to be jointly optimized [16].

[1] **Source code available**:
https://github.com/KimSehun725/seq2seqGuitarTranscription

Most of the recent state-of-the-art systems were mainly based on end-to-end deep neural network (DNN) models since end-to-end DNN models have the advantage of the ability to jointly optimize the whole model, showing better results compared to multi-step approaches [13]. Wiggins et al. proposed a convolutional neural network (CNN)-based model architecture [13] that estimates the frame-wise fingering position of a guitar performance. In our previous work [17], a self-attention mechanism was introduced along with CNN to better capture long-term relations and estimate the fingering position in both frame-level and note-level. We proved the effectiveness of the self-attention mechanism used in the guitar transcription model. However, since the proposed systems in [13] and [17] do not detect onset, the output can not be interpreted into reproducible forms such as music score or MIDI.

2.2 Automatic music transcription using tokens

Recently, the use of generic encoder-decoder Transformer architecture has shown its potential in automatic music transcription tasks. Howthorne et al. proposed a generic Transformer architecture for an automatic piano transcription [4]. The proposed method takes mel-spectrogram of audio and autoregressively generates a token sequence. The tokenization method used in this work is similar to how MIDI file stores its note sequences. The vocabulary consists of `note`, `velocity`, and `time` tokens, with the addition of an end-of-sentence (`EOS`) token for ending the sequence. In this tokenization method, the timing of each note is represented with absolute time location within the segment, quantized into 10 ms bins. This kind of tokenization method which represents the time of a note in location as opposed to the time shift from the previous note was reported to work better [18].

Chen et al. proposed a multi-objective generic Transformer model that not only predicts token sequences but also frame-level onsets, offsets, and pitch activation [1]. In the original paper of [1], the authors report that although the proposed model is a generic Transformer model that generates a sequence of tokens, introducing multi-task learning with frame-level labels, i.e., the pianoroll representation, lowers the performance of the token-wise prediction from the decoder, but improves frame-level estimation performance compared to frame-level guitar transcription model proposed in [19]. The authors also mentioned that the lower performance of the model without multi-task learning (that only predicts token sequence) could be attributed to the insufficient size of the training set to learn a dependable language model for the decoder.

3. PROPOSED METHOD

3.1 Hybrid CTC-Attention model for tokenized guitar transcription

3.1.1 Tokenization

As for the tokenization method, we use a slightly modified version of revamped MIDI-derived events (REMI) [20].

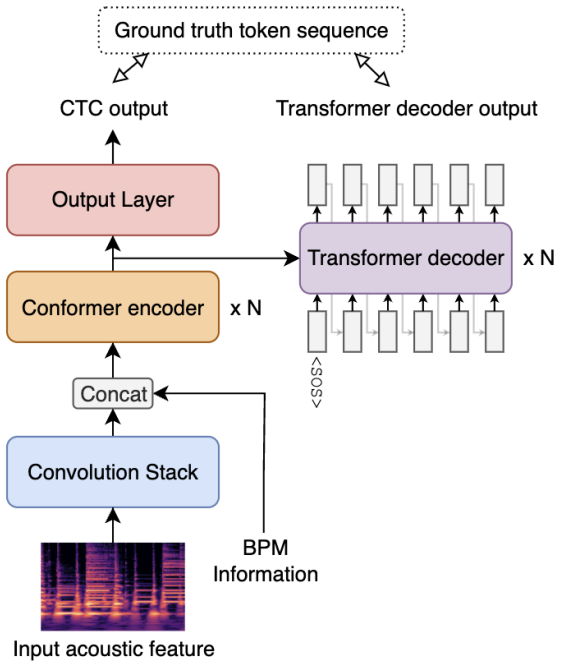


Figure 2. Overview of our proposed model architecture.

Since our main interest is to accurately transcribe a performance with pitch and timing information, we excluded `velocity` tokens and `chord` tokens from the original REMI. Excluding these tokens makes the total sequence shorter, making it easier to pass the restriction of CTC (token sequence length must be shorter than input sequence length), and easier to train. The tokenization method we used in our proposed approach includes the following vocabulary:

Blank [1 category] Used to represent the `blank` token when using CTC. This token gets dropped when decoding the final prediction when applying CTC.

Position [16 categories] Indicates the location in a bar quantized into 16th note. This token is placed to indicate the start position of a note.

Pitch [45 categories] Each class represents the pitch ranging from `E1` to `C5`. This token is placed after the `Position` token to indicate the pitch of a note.

Duration [16 categories] Represents the duration ranging from the 16th note to a bar in 16th note increments. This token is placed after the `Pitch` token to indicate the duration of a note.

Bar [1 category] Token used to denote the start of a bar.

SOS, EOS [2 categories] Used to represent the start and end of a token sequence. These tokens are used for training and inferencing with the Transformer decoder.

3.1.2 Encoder

Figure 2 shows the overview of the proposed model architecture. We will refer to the left side of the figure as encoder and the right side as decoder from here on out.

The structure of the encoder of our proposed model is largely inspired by our previous automatic guitar transcription model proposed in [17], with some modifications for generating a token sequence and conditioning with BPM information. The encoder structure can be divided into three main parts: a convolution stack, a Conformer encoder, and an output layer for generating a token sequence to which CTC can be applied later.

The convolution stack has 2D convolution, max pooling, dropout layers, and a linear layer. Input features go through two convolution blocks with 2D convolution, batch normalization, and an activation function. Latent features are then subsampled by max pooling and refined by another convolution block and max pooling layer. Lastly, a linear layer is added to reduce dimension. Three dropout layers prevent overfitting after max pooling and the final linear layer.

The Conformer encoder closely follows the Conformer block architecture proposed in [21]. The Conformer encoder mainly consists of self-attention modules, convolution modules, and feed-forward modules. For the input to the Conformer encoder, first, we concatenate the output from the convolution stack and the given BPM information of the input segment. Then, the concatenated feature goes through a linear transformation layer, which is omitted from Figure 2 for simplicity. We concatenate BPM information to the output of the convolution stack because the problem formulation of our method is estimating a sequence of tokens based on both acoustic features and BPM information.

Finally, the output layer is a simple linear transformation layer with softmax function at the end for generating CTC token outputs. Unlike the model that predicts frame-level activation probability (pianoroll) from the encoder [1], the encoder of our proposed model generates the probability of token sequence, which we can directly calculate the loss between the output and the ground truth token sequence by calculating CTC loss [6].

During inference, we first apply `argmax` to the encoder outputs, then repeating tokens get merged. Then, the `blank` token gets removed to obtain the final output.

3.1.3 Decoder

The structure of the decoder in our proposed model is roughly the same as the ones used in ASR tasks or other symbolic music transcription tasks such as in [1, 4]. The decoder consists of multiple Transformer decoder blocks stacked in serial. The Transformer decoder block consists of a masked self-attention module, a cross-attention module, and a feed-forward module.

The decoder is trained and validated with a teacher forcing scheme where the token is predicted one step ahead without self-attention looking ahead by masking the self-attention with a diagonal mask in a non-autoregressive manner. During inference, we only give a start-of-sentence (`SOS`) token at first, and autoregressively generate the following tokens by selecting the most probable tokens at each timestep. The generation stops when the decoder gen-

erates an end-of-sentence (EOS) token.

3.1.4 Multi-task learning

Our proposed method is a multi-objective model with both the CTC output from the encoder and the output from the decoder. Therefore, we define a custom loss function for backpropagation.

First, we define the CTC loss $\mathcal{L}_{\mathrm{CTC}}$ as

$$\mathcal{L}_{\mathrm{CTC}} = -\sum_{\mathbf{y}\in\mathcal{B}} \log p(\mathbf{y}|\mathbf{x}), \tag{1}$$

where $\mathcal{B}$ is the set of all possible output sequences including blank symbols, $\mathbf{x}$ is the input sequence, and $p(\mathbf{y}|\mathbf{x})$ is the probability of generating the output sequence $\mathbf{y}$ from the encoder given the input sequence $\mathbf{x}$. The CTC loss is calculated by summing the negative log probabilities of all possible output sequences $\mathbf{y}$ that can be generated from the ground truth sequence. The loss encourages the model to learn to generate the correct output sequence while accounting for possible alignment errors between the input and output sequences.

Next, we define the cross-entropy loss as

$$\mathcal{L}_{\mathrm{CE}} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} \mathbf{y}_{ij} \log\left(\hat{\mathbf{y}}_{ij}\right), \tag{2}$$

where N is the number of samples, C is the number of classes, $\mathbf{y}_{ij}$ is a binary indicator of whether sample i belongs to class j, and $\hat{\mathbf{y}}_{ij}$ is the predicted probability from the decoder of sample i belonging to class j.

The total loss function of our system $\mathcal{L}_{\mathrm{total}}$ can be expressed as

$$\mathcal{L}_{\mathrm{total}} = \alpha\mathcal{L}_{\mathrm{CTC}} + \mathcal{L}_{\mathrm{CE}}, \tag{3}$$

where α is the hyperparameter for controlling the weight of $\mathcal{L}_{\mathrm{CTC}}$.

3.2 Data augmentation

When using a tokenization method whose time resolution is in units of musical lengths, training a model to generate a sequence of tokens requires a large amount of data to properly model the language structure of the tokenization method and the concept of musical length. However, unlike musical instruments such as the piano, which have a significant amount of publicly available training data, the guitar lacks sufficient data for training. The goal of the proposed data augmentation methods is to scale up the amount of data used in the training process.

3.2.1 Bar overlap

In the field of automatic music transcription, splitting a musical excerpt into multiple segments is common, especially with the models that use the attention mechanism since the attention mechanism has a space complexity of $O(n^2)$ with respect to sequence length n. There have been many attempts to train a network by cutting musical pieces into exact lengths in seconds. However, there have been only a few attempts to handle music pieces by cutting them into the same musical length, e.g., 4 bars [17]. When cutting a musical piece into segments whose length is in units of bars, the most naive way of cutting it would be to cut without overlap so that the timing of the start of a segment is the end of the previous one. This results in obtaining $l_{excerpt}/l_{segment}$ segments, assuming that $l_{excerpt}$ is devidable by $l_{segment}$, where $l_{excerpt}$ denotes the bar length of a musical excerpt, and $l_{segment}$ denotes the bar length of a segment. This method was done in our previous work [17].

We propose a method that creates more segments when cutting musical excerpts, by overlapping segments, i.e. sliding a window with a hop length shorter than the window length. This results in obtaining $l_{excerpt}/l_{overlap} - (l_{segment} - 1)$ where $l_{overlap}$ denotes the bar length of the overlap.

3.2.2 Synthetic audio-MIDI pair

With the goal of training a model to properly learn how to generate the token sequence by reliably training the language model for the decoder with a large amount of data, we propose a method that can synthetically create an audio-MIDI pair dataset from MIDI-only data. We generate synthetic audio data by utilizing an oscillator such as a sinusoidal oscillator or a square wave oscillator. This results in obtaining an audio-MIDI pair with its audio being perfectly aligned with the matching MIDI, yet with unnatural timbre. With the synthetically generated audio-MIDI pair dataset, we pretrain the model before training with real-world data. This results in the decoder being trained as a reliable language model for tokenization method, and the model having preliminary knowledge regarding extracting pitch and timing information.

It is possible to use the method to produce an endless amount of data theoretically, by applying it to either a publicly available MIDI dataset or automatically generated MIDI from an automatic symbolic music generation model such as [20,22,23] since the method generates audio-MIDI pair data solely from MIDI.

3.3 Implementation details

Regarding the network settings for the encoder and decoder of our proposed model, we set the number of attention heads and the number of sequential Conformer blocks to 8 and 6 respectively. For the Transformer decoder, the number of attention heads and the number of sequential blocks are set to 4 and 4 respectively. The dimension of both the Conformer encoder and the Transformer decoder is set to 128. We implement the Conformer encoder using the ESPNet2 framework [11]. We use the leaky ReLU activation function [24] throughout the encoder, except for the activation functions in the Conformer encoder and the final output layer.

For the implementation of tokenization, we use MidiTok [25], but slightly modify the original implementation as mentioned in Section 3.1.1. As for the learning rate, we use a cyclic learning rate scheme [26]. The base learning rate is set to 1e-5, the max learning rate is set to 0.001, the number of training iterations in the increasing half of a cy-

Method	Encoder output		Decoder output	
	F1	**TER**	**F1**	**TER**
No data augmentation	0.363±0.159	0.469±0.185	0.526±0.154	0.713±0.219
Bar overlap (BO)	0.555±0.125	0.388±0.090	0.630±0.196	0.497±0.176
Pretrain (PT)	0.512±0.043	0.365±0.029	0.699±0.017	0.441±0.011
Proposed (BO+PT)	**0.666**±0.047	**0.307**±0.025	**0.804**±0.015	**0.336**±0.021

Table 1. Estimation metrics for evaluating the proposed data augmentation methods. For all metrics, we report the mean and standard deviation over the entire dataset. All the experiments were done with the proposed model.

Model	Encoder output		Decoder output	
	F1	**TER**	**F1**	**TER**
Baseline [1]	**0.767**±0.026	-	0.603±0.026	0.589±0.017
Attention only	-	-	0.784±0.014	0.345±0.021
Proposed	0.666±0.047	0.307±0.025	**0.804**±0.015	**0.336**±0.021

Table 2. Estimation metrics for our proposed model, compared with a baseline model and a model without multi-task learning with CTC. For all metrics, we report the mean and standard deviation over the entire dataset. All the experiments were done by applying both of the proposed data augmentation methods. Note that the baseline model does not have TER from the encoder output because the encoder output is pianoroll-like frame-level annotation, not tokens.

Model	Decoder output	
	F1	**TER**
Attention only + BO	0.114±0.046	1.520±0.378
Proposed + BO	**0.630**±0.196	**0.497**±0.176

Table 3. Estimation metrics for simulating the situation where only a small amount of training data is available by not pretraining with synthetic audio-MIDI pair data. We only show the results of the output from the decoder for simplicity.

cle is set to 4 epochs, and the same for the decreasing half. We made the peak learning rate decreases by a rate of 0.9 after each cycle. For the optimizer, we employ Rectified Adam (RAdam) [27]. Finally, we set the weight α which is used in the loss function, to 0.2.

4. EXPERIMENTAL EVALUATION

4.1 Experimental conditions

As for the MIDI data used in the pretraining phase, we used data from Classical Guitar MIDI Archives [28]. We filtered out the data that did not have the properties that we want such as having a time signature other than 4/4 beat or changing tempo over time. As a result, we obtained a total of 1232 minutes of data. We used square wave oscillator to make the synthetic audio, and split the dataset for training/validation/testing with a ratio of 0.9:0.05:0.05.

For the data used in the finetuning phase, we used the GuitarSet [29]. GuitarSet is a dataset for guitar transcription research containing 360 audio recordings, totaling approximately 3 hours. Since the dataset was recorded by six players, we left the recordings of one player for testing and used the rest of the data for training and validation. We rotated the test player to evaluate the methods with a six-fold cross-validation method. For both the pretraining data and finetuning data, we cut the tracks into segments with 4 bars, with 1 bar hop length.

Regarding the input of the network, first, we resampled the audio to 22050 Hz, then we converted the audio to a constant-Q transform (CQT) [30] with a hop length of 256 points, 24 bins per octave spanning over 8 octaves, resulting in a total of 192 frequency bins.

We evaluated the proposed methods in three points: 1) the effectiveness of data augmentation methods, 2) the performance of the model, 3) the effectiveness of introducing CTC in the proposed model when there is only a small amount of training data available. For the experiment measuring the effectiveness of the data augmentation techniques, we compared the metrics of 1) using GuitarSet only and not using bar overlap, 2) only applying bar overlap, 3) pretraining the model with synthetic audio-MIDI pair dataset without bar overlapping, and 4) using both bar overlapping and pretraining.

For the model comparison, we compared our proposed model with 1) the model proposed in [1] which refer to as baseline hereafter, 2) our proposed model without the use of CTC ($\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{CE}}$) which we refer to as attention only model from here on out, and 3) our proposed model.

Regarding the network settings of the baseline model [1], we followed the settings described in the original paper, except we input a 4-bar-long acoustic feature to the network and use the same tokenization method as our proposed method.

4.2 Evaluation metrics

Since the output of our proposed model is a sequence of tokens that can be converted into pianoroll, we used different evaluation metrics for pianoroll domain and token domain.

In the pianoroll domain, we used precision, recall, and F1 score. If the output is a sequence of tokens, we decode the token sequence to pianoroll to calculate precision, recall, and F1 score. In the token domain, we used token er-

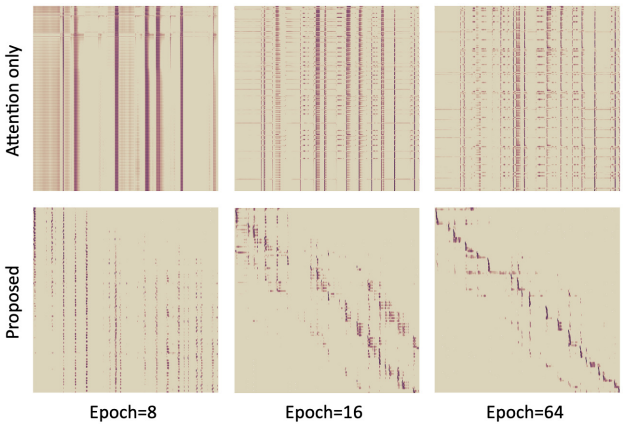


Figure 3. Comparison of the speed in learning alignments between acoustic features (horizontal axis) and tokens (vertical axis). The training was done using only the GuitarSet.

ror rate (TER), which is calculated similarly to word error rate (WER) which is a widely used metric in the research field of ASR and natural language processing (NLP). The only difference is that every element is a token index instead of a word in WER. The TER can be calculated as

$$\mathrm{TER} = \frac{S + D + I}{S + D + C}, \tag{4}$$

where S is the number of substitutions, D is the number of deletions, I is the number of insertions, and C is the number of correct tokens.

The reason why we introduce TER as an evaluation metric is that we want to compensate for the mismatch between the metrics used in pianoroll domain and human perception. For instance, if we only evaluate the system in the pianoroll domain, the outputs with the notes that are shifted for a consistent amount of time will have very low scores. However in the same case, if we use TER as an evaluation metric, only the `position` tokens will lower the metric, not penalizing for the shifted notes as much.

4.3 Results

The result of the experiment comparing the effectiveness of the data augmentation methods is shown in Table 1. The result shows that both data augmentation methods are effective in raising the estimation performance of the proposed model. By comparing the effectiveness of bar overlapping and pretraining, pretraining shows slightly better results in the output from the decoder but worse in the output of the encoder.

The result of the experiment comparing the performance of the models mentioned in Section 4.1 is shown in Table 2. The result shows that our proposed model outperformed the baseline model and the proposed model without utilizing CTC (attention only) in both F1 score and TER. Upon comparing the estimation results of the attention-only model with the proposed model, it is evident that the latter produced superior results, thereby indicating that the inclusion of CTC considerably improves transcription performance.

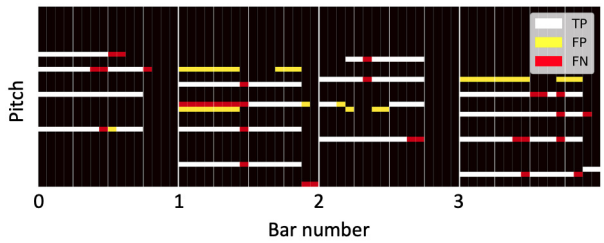


Figure 4. A sample of transcription result from our proposed model. TP, FP, and FN denote true positive, false positive, and false negative respectively.

The result of the experiment simulating a situation with a small amount of training data available is shown in Table 3. We simulated this situation by only using the GuitarSet for training. The result shows that the model with attention only performed very poorly when there is only a small amount of data. However, our proposed model which utilizes multitask learning with CTC outputs from the encoder performs better compared to attention only model. This indicates that employing multi-task learning with CTC is highly effective when there is an insufficient amount of data. Figure 3 shows the attention alignments between acoustic features and tokens. We observed that despite being trained for 64 epochs, the attention only model failed to acquire a reasonable alignment, whereas the suggested model achieved to acquire the desired alignments early on in the training process. The performance difference between the attention only model and the proposed model is likely to be attributed to the difference in the difficulty of learning the correct alignments.

While we did not include the outcomes of using solely synthetic audio-MIDI pair data for both training and testing in any of the tables, it is worth stating that during the pretraining phase of the experiment utilizing the proposed model and data augmentation methods, the output of the decoder with the test data yielded an F1 score of 0.959 and TER of 0.029. This indicates that the amount of data used in the pretraining was sufficient enough to train a reliable language model for the decoder.

5. CONCLUSION

In this paper, we proposed two data augmentation methods for training sequence-to-sequence networks that used tokenized music representation as output, and a hybrid CTC-Attention model for automatic guitar transcription. We confirmed that 1) both of the data augmentation methods are highly effective in training the sequence-to-sequence models when there is an insufficient amount of data, 2) our proposed hybrid CTC-Attention model outperforms conventional methods that transcribe guitar performance with tokens, and 3) the addition of multi-task learning with CTC in our proposed model is especially effective when there is an insufficient amount of training data.

6. ACKNOWLEDGMENT

This work was partly supported by JST CREST Grant Number JPMJCR19A3 and JSPS KAKENHI Grant Number 21H04892, Japan.

7. REFERENCES

[1] Y.-H. Chen, W.-Y. Hsiao, T.-K. Hsieh, J.-S. R. Jang, and Y.-H. Yang, "Towards automatic transcription of polyphonic electric guitar music: A new dataset and a multi-loss transformer model," in *2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 786–790.

[2] X. Fiss and A. Kwasinski, "Automatic real-time electric guitar audio transcription," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011, pp. 373–376.

[3] C. Zhang, Y. Ren, K. Zhang, and S. Yan, "Sdmuse: Stochastic differential music editing and generation via hybrid representation," *arXiv preprint arXiv:2211.00222*, 2022.

[4] 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 (ISMIR)*, 2021.

[5] 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.

[6] 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 (ICML)*, 01 2006, pp. 369–376.

[7] 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 (NIPS)*, vol. 30, pp. 5998–6008, 2017.

[8] S. Watanabe, T. Hori, S. Kim, J. R. Hershey, and T. Hayashi, "Hybrid ctc/attention architecture for end-to-end speech recognition," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1240–1253, 2017.

[9] H. Kheddar, Y. Himeur, S. Al-Maadeed, A. Amira, and F. Bensaali, "Deep transfer learning for automatic speech recognition: Towards better generalization," *arXiv preprint arXiv:2304.14535*, 2023.

[10] A. Graves, "Sequence transduction with recurrent neural networks," *arXiv preprint arXiv:1211.3711*, 2012.

[11] P. Guo, F. Boyer, X. Chang, T. Hayashi, Y. Higuchi, H. Inaguma, N. Kamo, C. Li, D. Garcia-Romero, J. Shi, J. Shi, S. Watanabe, K. Wei, W. Zhang, and Y. Zhang, "Recent developments on espnet toolkit boosted by conformer," in *2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 5874–5878.

[12] 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 *Proc. 17th International Conference on Digital Audio Effects (DAFx-14)*, 2014.

[13] A. Wiggins and Y. E. Kim, "Guitar tablature estimation with a convolutional neural network," in *Proc. 20th International Society for Music Information Retrieval Conference (ISMIR)*, 2019, pp. 284–291.

[14] 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.

[15] G. Hori, H. Kameoka, and S. Sagayama, "Input-output HMM applied to automatic arrangement for guitars," *Journal of Information Processing*, vol. 21, no. 2, pp. 264–271, 2013.

[16] G. W. Lee and H. K. Kim, "Two-step joint optimization with auxiliary loss function for noise-robust speech recognition," *Sensors*, vol. 22, no. 14, 2022.

[17] S. Kim, T. Hayashi, and T. Toda, "Note-level automatic guitar transcription using attention mechanism," in *2022 30th European Signal Processing Conference (EUSIPCO)*, 2022, pp. 229–233.

[18] 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, pp. 955–967, 2020.

[19] 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)*. Paris, France: ISMIR, Sep. 2018, pp. 50–57.

[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 28th ACM International Conference on Multimedia*, 2020, pp. 1180–1188.

[21] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, "Conformer: Convolution-augmented Transformer for speech recognition," in *Proc. Interspeech*, 2020, pp. 5036–5040.

[22] 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 (ISMIR)*, 2021.

[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, AAAI*, 2021.

[24] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," *arXiv preprint arXiv:1505.00853*, 2015.

[25] 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 (ISMIR)*, 2021.

[26] L. N. Smith, "Cyclical learning rates for training neural networks," in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2017, pp. 464–472.

[27] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and J. Han, "On the variance of the adaptive learning rate and beyond," in *International Conference on Learning Representations (ICLR)*, 2020.

[28] "Classical guitar midi archives," accessed on April 1, 2023. [Online]. Available: https://www.classicalguitarmidi.com/

[29] Q. Xi, R. Bittner, J. Pauwels, X. Ye, and J. P. Bello, "Guitarset: A dataset for guitar transcription," in *Proc. 19th International Society for Music Information Retrieval Conference (ISMIR)*, Paris, France, Sep. 2018, pp. 453–460.

[30] J. Youngberg and S. Boll, "Constant-Q signal analysis and synthesis," in *1978 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 3, 1978, pp. 375–378.

Papers – Session V

PESTO: PITCH ESTIMATION WITH SELF-SUPERVISED TRANSPOSITION-EQUIVARIANT OBJECTIVE

Alain Riou[1,2] **Stefan Lattner**[2] **Gaëtan Hadjeres**[3] **Geoffroy Peeters**[1]
[1] LTCI, Télécom-Paris, Institut Polytechnique de Paris, France
[2] Sony Computer Science Laboratories - Paris, France
[3] Sony AI
alain.riou@sony.com

ABSTRACT

In this paper, we address the problem of pitch estimation using Self Supervised Learning (SSL). The SSL paradigm we use is equivariance to pitch transposition, which enables our model to accurately perform pitch estimation on monophonic audio after being trained only on a small unlabeled dataset. We use a lightweight ($<$ 30k parameters) Siamese neural network that takes as inputs two different pitch-shifted versions of the same audio represented by its Constant-Q Transform. To prevent the model from collapsing in an encoder-only setting, we propose a novel class-based transposition-equivariant objective which captures pitch information. Furthermore, we design the architecture of our network to be transposition-preserving by introducing learnable Toeplitz matrices.

We evaluate our model for the two tasks of singing voice and musical instrument pitch estimation and show that our model is able to generalize across tasks and datasets while being lightweight, hence remaining compatible with low-resource devices and suitable for real-time applications. In particular, our results surpass self-supervised baselines and narrow the performance gap between self-supervised and supervised methods for pitch estimation.

1. INTRODUCTION

Pitch estimation is a fundamental task in audio analysis, with numerous applications, e.g. in Music Information Retrieval (MIR) and speech processing. It involves estimating the fundamental frequency of a sound, which allows to estimate its perceived pitch. Over the years, various techniques have been developed for pitch estimation, ranging from classical methods (based on signal processing) [1–4] to machine learning approaches [5, 6].

In recent years, deep learning has emerged as a powerful tool for a wide range of applications, outperforming classical methods in many domains. This is notably true in MIR, where deep learning has led to significant advances in tasks such as music transcription [7–9], genre classification [10–12], and instrument recognition [13–15]. Pitch estimation has also benefited greatly from deep learning techniques [16, 17]. However, these deep learning models often require a large amount of labelled data to be trained, and can be computationally expensive, hindering their practical applications in devices with limited computing power and memory capabilities. Additionally, these models are often task-specific and may not generalize well to different datasets or tasks [18]. Therefore, there is a need for a lightweight and generic model that does not require labelled data to be trained. We address this here.

We take inspiration from the equivariant pitch estimation [19] and the equivariant tempo estimation [20] algorithms which we describe in part 2. As those, we use a SSL paradigm based on Siamese networks and equivariance to pitch transpositions (comparing two versions of the same sound that have been transposed by a random but known pitch shift). We introduce a new equivariance loss that enforces the model to capture pitch information specifically.

This work has the following **contributions**:

- we formulate pitch estimation as a multi-class problem (part 3.1); while [19, 20] model pitch/tempo estimation as a regression problem,
- we propose a novel class-based equivariance loss (part 3.1) which prevents collapse; while [19] necessitates a decoder,
- the architecture of our model is lightweight and transposition-equivariant by design. For this, we introduce Toeplitz fully-connected layers (part 3.4).

We evaluate our method on several datasets and show that it outperforms self-supervised baselines on single pitch estimation (part 4.4.1). We demonstrate the robustness of our method to domain-shift and background music, highlighting its potential for real-world applications (part 4.4.2).

Our proposed method requires minimal computation resources and is thus accessible to a wide range of users for both research and musical applications. In consideration of accessibility and reproducibility, we make our code and pretrained models publicly available [1] .

[1] https://github.com/SonyCSLParis/pesto

© A. Riou, S. Lattner, G. Hadjeres and G. Peeters. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** A. Riou, S. Lattner, G. Hadjeres and G. Peeters, "PESTO: Pitch Estimation with Self-supervised Transposition-equivariant Objective", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

2. RELATED WORKS

2.1 SSL to learn invariant representations.

Siamese networks. Most common techniques for SSL representation involve Siamese networks [21]. The underlying idea is to generate two views of an input, feed them to a neural network, and train the network by applying a criterion between the output embeddings. Various techniques have been developed for generating views [2].

Collapse. However, a major issue with these methods is "collapse", when all inputs are mapped to the same embedding. To address this, various techniques have been proposed. One of the most common is SimCLR [22] which also uses negative samples to ensure that embeddings are far apart through a contrastive loss. Additionally, several regularization techniques have been developed that minimize a loss over the whole batch. Barlow Twins [23] force the cross-correlation between embeddings to be identity, while VICReg [24] add loss terms on the statistics of a batch to ensure that dimensions of the embeddings have high enough variance while remaining independent of each other. On the other hand, [25] explicitly minimize a loss over the hypersphere to distribute embeddings uniformly. Furthermore, incorporating asymmetry between inputs has been shown to improve performance. [26, 27] uses a momentum encoder, while [28] and [29] add a projection head and a stop-gradient operator on top of the network, with [28] also using a teacher network. Finally, [30] incorporates asymmetry to contrastive- and clustering-based representation learning.

Application to audio. While originally proposed for computer vision, these methods have been successfully adapted to audio and music as well. For example, [31], [32], and [33] respectively adapted [22], [23], and [28] to the audio domain. By training their large models on AudioSet [34], they aim at learning general audio representations that are suited for many downstream tasks. More specifically, [35] successfully adapts contrastive learning to the task of music tagging by proposing more musically-relevant data augmentations.

2.2 SSL to learn equivariant representations.

The purpose of the methods described above is to learn a mapping $f : \mathcal{X} \rightarrow \mathcal{Y}$ that is *invariant* to a set of transforms $\mathcal{T}_{\mathcal{X}}$, i.e. so that for any input $\mathbf{x} \in \mathcal{X}$ and transform $t \in \mathcal{T}_{\mathcal{X}}$

$$f(t(\mathbf{x})) \approx f(\mathbf{x}) \tag{1}$$

However, recent approaches [36–38] try instead to learn a mapping f that is *equivariant* to $\mathcal{T}_{\mathcal{X}}$, i.e. that satisfies

$$f(t(\mathbf{x})) \approx t'(f(\mathbf{x})) \tag{2}$$

where $t' \in \mathcal{T}_{\mathcal{Y}}$ with $\mathcal{T}_{\mathcal{Y}}$ a set of transforms that acts on the output space $\mathcal{Y}$. In other words, if the input is transformed, the output should be transformed accordingly. Representation collapse is hence prevented by design.

[2] The most common technique involves randomly applying data augmentations to inputs to create pairs of inputs that share semantic content.

Equivariant representation learning has mostly been applied to computer vision and usually combines an invariance and an equivariance criterion. E-SSL [36] trains two projection heads on top of an encoder, one to return projections invariant to data augmentations while the other predicts the parameters of the applied data augmentations. [37] predicts separately a semantic representation and a rotation angle of a given input and optimizes the network with a reconstruction loss applied to the decoded content representation rotated by the predicted angle. Finally, SIE [38] creates a pair of inputs by augmenting an input and learns equivariant representations by training a hypernetwork conditioned on the parameters of the augmentation to predict one embedding of the pair from the other.

Application to audio. Finally, a few successful examples of equivariant learning for solving MIR tasks recently emerged [19,20]. In particular, [20] introduces a simple yet effective equivariance criterion for tempo estimation while preventing collapse without any decoder or regularization: pairs are created by time-stretching an input with two different ratios, then the output embeddings are linearly projected onto scalars and the network is optimized to make the ratio of the scalar projections match the time-stretching ratio within a pair.

2.3 Pitch estimation.

Monophonic pitch estimation has been a subject of interest for over fifty years [39]. The earlier methods typically obtain a pitch curve by processing a candidate-generating function such as cepstrum [39], autocorrelation function (ACF) [40], and average magnitude difference function (AMDF) [41]. Other functions, such as the normalized cross-correlation function (NCCF) [1, 2] and the cumulative mean normalized difference function [3,42], have also been proposed. On the other hand, [4] performs pitch estimation by predicting the pitch of the sawtooth waveform whose spectrum best matches the one of the input signal.

Recently, methods involving machine learning techniques have been proposed [5, 6]. In particular, CREPE [16] is a deep convolutional network trained on a large corpus to predict pitch from raw audio waveforms. SPICE [19] is a self-supervised method that takes as inputs individual Constant-Q Transform (CQT) frames of pitch-shifted inputs and learns the transposition between these inputs. It achieves quite decent results thanks to a decoder that takes as input the predicted pitch and tries to reconstruct the original CQT frame from it.

Finally, some works [43, 44] aim at disentangling the pitch and timbre of an input audio, thus predicting pitch as a side effect. In particular, DDSP-inv [45] is a DDSP-based approach [46] that relies on inverse synthesis to infer pitch in a self-supervised way.

3. SELF-SUPERVISED PITCH ESTIMATION

3.1 Transposition-equivariant objective

We focus on the problem of monophonic pitch estimation and model it as a classification task. Our model is com-

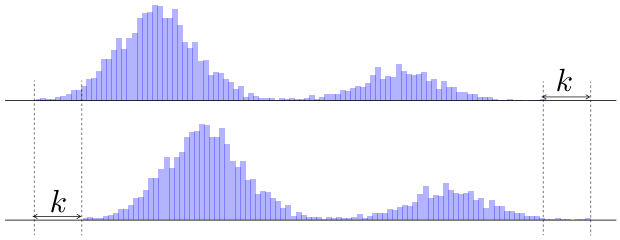


Figure 1. Example of k-transpositions. Visually, $\mathbf{y}$ and $\mathbf{y}'$ are just translated versions of each other. The sign of k and its absolute value respectively indicate the direction and the distance of the translation.

posed of a neural network f_θ that takes as input an audio signal $\mathbf{x}$ and returns a vector $\mathbf{y} = (y_0, \ldots, y_i, \ldots, y_{d-1}) \in [0,1]^d$, which represents the probability distribution of each pitch i. y_i represents the probability that i is the pitch of $\mathbf{x}$. We propose here to train f_θ in a SSL way. For this, similarly to [22, 24, 26, 28, 29], we use data augmentations and Siamese networks.

Given $\mathbf{x}$, we first generate $\mathbf{x}^{(k)}$ by pitch-shifting $\mathbf{x}$ by a *known* number k of semitones. Then, both $\mathbf{x}$ and $\mathbf{x}^{(k)}$ are fed to f_θ which is trained to minimize a loss function between $\mathbf{y} = f_\theta(\mathbf{x})$ and $\mathbf{y}^{(k)} = f_\theta(\mathbf{x}^{(k)})$.

Definition. *For two vectors* $\mathbf{y}, \mathbf{y}' \in \mathbb{R}^d$ *and* $0 \leq k < d$, $\mathbf{y}'$ *is a* k*-transposition of* $\mathbf{y}$ *if and only if for all* $0 \leq i < d$

$$\begin{cases} y'_{i+k} = y_i & \text{when } 0 \leq i < d-k \\ y'_i = 0 & \text{when } i < k \\ y_i = 0 & \text{when } i \geq d-k-1 \end{cases} \tag{3}$$

Similarly, for $-d < k \leq 0$, $\mathbf{y}'$ *is a* k*-transposition of* $\mathbf{y}$ *if and only if* $\mathbf{y}$ *is a* $-k$*-transposition of* $\mathbf{y}'$.

The concept of k-transposition is illustrated in Figure 1. Note also that for a vector $\mathbf{y} \in \mathbb{R}^d$, exists at most one vector $\mathbf{y}' \in \mathbb{R}^d$ that is a k-transposition of $\mathbf{y}$. We can therefore refer to $\mathbf{y}'$ as *the* k-transposition of this vector $\mathbf{y}$.

Equivariance loss. We then design our criterion based on the following assumption: the probability of $\mathbf{x}$ to have pitch i is equal to the probability of $\mathbf{x}^{(k)}$ to have pitch $i+k$, i.e. y_i should be equal to $y^{(k)}_{i+k}$ [3]. In other words, if $\mathbf{x}^{(k)}$ is a pitch-shifted version of $\mathbf{x}$, their respective pitch probability distributions should be shifted accordingly, i.e. $\mathbf{y}^{(k)}$ should be the k-transposition of $\mathbf{y}$.

We take inspiration from [20] to design our equivariance loss. However, in our case, the output of our network f_θ is not a generic representation but a probability distribution. We therefore adapt our criterion by replacing the learnable linear projection head from [20] by the following *deterministic* linear form:

$$\begin{array}{rccc} \phi & : & \mathbb{R}^d & \rightarrow & \mathbb{R} \\ & & \mathbf{y} & \mapsto & (\alpha, \alpha^2, \ldots, \alpha^d)\mathbf{y} \end{array} \tag{4}$$

where α is a fixed hyperparameter [4].

[3] For example, if $k = 2$ semitones, the probability of $\mathbf{x}$ to be C4 is exactly the probability of $\mathbf{x}^{(k)}$ to be a D4, and the same holds for any pitch independently of the actual pitch of $\mathbf{x}$.

[4] We found $\alpha = 2^{1/36}$ to work well in practice.

Indeed, with this formulation, for any k if $\mathbf{y}'$ is a k-transposition of $\mathbf{y}$ then $\phi(\mathbf{y}') = \alpha^k \phi(\mathbf{y})$. Hence we define our loss as

$$\mathcal{L}_{\text{equiv}}(\mathbf{y}, \mathbf{y}^{(k)}, k) = h_\tau \left(\frac{\phi(\mathbf{y}^{(k)})}{\phi(\mathbf{y})} - \alpha^k \right) \tag{5}$$

where h_τ is the Huber loss function [47], defined by

$$h_\tau(x) = \begin{cases} \frac{x^2}{2} & \text{if } |x| \leq \tau \\ \frac{\tau^2}{2} + \tau(|x| - \tau) & \text{otherwise} \end{cases} \tag{6}$$

Regularization loss. Note that if $\mathbf{y}^{(k)}$ is the k-transposition of $\mathbf{y}$ then $\mathcal{L}_{\text{equiv}}(\mathbf{y}, \mathbf{y}^{(k)}, k)$ is minimal. However, the converse is not always true. In order to actually enforce pitch-shifted pairs of inputs to lead to k-transpositions, we further add a regularization term which is simply the shifted cross-entropy (SCE) between $\mathbf{y}$ and $\mathbf{y}^{(k)}$, i.e. the cross-entropy between the k-transposition of $\mathbf{y}$ and $\mathbf{y}^{(k)}$:

$$\mathcal{L}_{\text{SCE}}(\mathbf{y}, \mathbf{y}^{(k)}, k) = \sum_{i=0}^{d-1} y_i \log \left(y^{(k)}_{i+k} \right) \tag{7}$$

with the out-of-bounds indices replaced by 0. The respective contribution of $\mathcal{L}_{\text{equiv}}$ and $\mathcal{L}_{\text{SCE}}$ is studied in part 4.4.3.

Invariance loss. $\mathcal{L}_{\text{equiv}}$ and $\mathcal{L}_{\text{SCE}}$ allow our model to learn relative transpositions between different inputs and learn to output probability distributions $\mathbf{y}$ and $\mathbf{y}^{(k)}$ that satisfy the equivariance constraints. However, these distributions may still depend on the timbre of the signal. This is because our model actually never observed at the same time two different samples with the same pitch.

To circumvent this, we rely on a set $\mathcal{T}$ of data augmentations that preserve pitch (such as gain or additive white noise). We create augmented views $\tilde{\mathbf{x}} = t(\mathbf{x})$ of our inputs $\mathbf{x}$ by applying random transforms $t \sim \mathcal{T}$.

Similarly to [35], we then train our model to be invariant to those transforms by minimizing the cross-entropy between $\mathbf{y} = f_\theta(\mathbf{x})$ and $\tilde{\mathbf{y}} = f_\theta(\tilde{\mathbf{x}})$.

$$\mathcal{L}_{\text{inv}}(\mathbf{y}, \tilde{\mathbf{y}}) = \text{CrossEntropy}(\mathbf{y}, \tilde{\mathbf{y}}) \tag{8}$$

Combining the losses. For a given input sample $\mathbf{x}$ and a given set of augmentations $\mathcal{T}$,

- we first compute $\mathbf{x}^{(k)}$ by pitch-shifting $\mathbf{x}$ by a random number of bins k (the precise procedure is described in section 3.2);
- we then generate two augmented views $\tilde{\mathbf{x}} = t_1(\mathbf{x})$ and $\tilde{\mathbf{x}}^{(k)} = t_2(\mathbf{x}^{(k)})$, where $t_1, t_2 \sim \mathcal{T}$;
- we compute $\mathbf{y} = f_\theta(\mathbf{x})$, $\tilde{\mathbf{y}} = f_\theta(\tilde{\mathbf{x}})$ and $\tilde{\mathbf{y}}^{(k)} = f_\theta(\tilde{\mathbf{x}}^{(k)})$.

Our final objective loss is then:

$$\begin{aligned} \mathcal{L}(\mathbf{y}, \tilde{\mathbf{y}}, \tilde{\mathbf{y}}^{(k)}, k) = \lambda_{\text{inv}} \ & \mathcal{L}_{\text{inv}}(\mathbf{y}, \tilde{\mathbf{y}}) \\ + \lambda_{\text{equiv}} \ & \mathcal{L}_{\text{equiv}}(\tilde{\mathbf{y}}, \tilde{\mathbf{y}}^{(k)}, k) \\ + \lambda_{\text{SCE}} \ & \mathcal{L}_{\text{SCE}}(\tilde{\mathbf{y}}, \tilde{\mathbf{y}}^{(k)}, k) \end{aligned} \tag{9}$$

We illustrate this in Figure 2. To set the weights λ_* we use the gradient-based method proposed by [48–50].

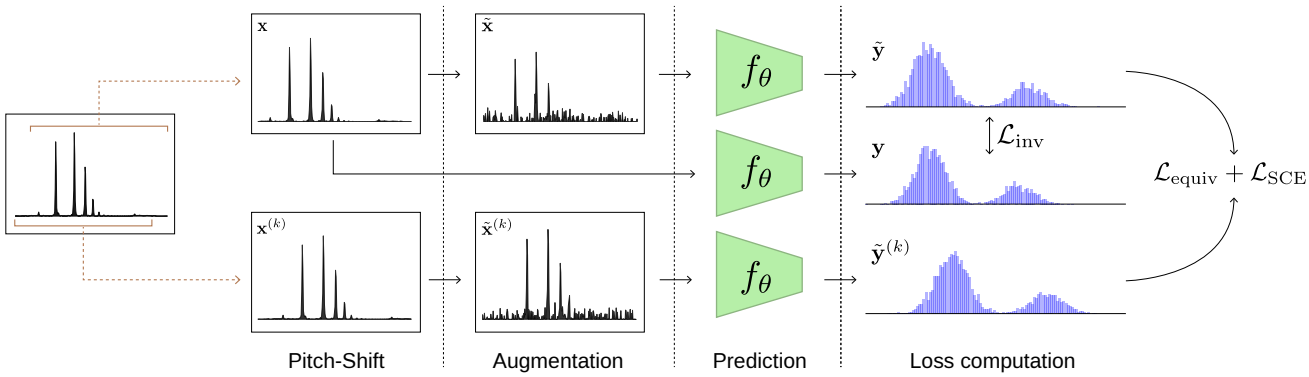


Figure 2. Overview of the PESTO method. The input CQT frame (log-frequencies) is first cropped to produce a pair of pitch-shifted inputs $(\mathbf{x}, \mathbf{x}^{(k)})$. Then we compute $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{x}}^{(k)}$ by randomly applying pitch-preserving transforms to the pair. We finally pass $\mathbf{x}$, $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{x}}^{(k)}$ through the network f_θ and optimize the loss between the predicted probability distributions.

3.2 Audio-frontend

The inputs $\mathbf{x}$ are the individual frames of the CQT. We have chosen the CQT as input since its logarithmic frequency scale, in which bins of the CQT exactly correspond to a fixed fraction b of pitch semitones, naturally leads to pitch-shifting by translation. CQT is also a common choice made for pitch estimation [17, 19, 51].

To compute the CQT, we use the implementation provided in the nnAudio library [52] since it supports parallel GPU computation. We choose $f_{\text{min}} = 27.5$ Hz, which is the frequency of A0 the lowest key of the piano and select a resolution of $b = 3$ bins per semitone. Our CQT has in total $K = 99b$ log-frequency bins, which corresponds to the maximal number of bins for a 16kHz signal.

3.3 Simulating translations.

To avoid any boundary effects, we perform pitch-shift by cropping shifted slices of the original CQT input frame as in [19] [5] . From a computational point of view, it is indeed significantly faster than applying classical pitch shift algorithms based on phase vocoder and resampling.

3.4 Transpostion-preserving architecture

The architecture of f_θ is illustrated in Figure 3. It is inspired by [17]. Each input CQT frame is processed independently: first layer-normed [53] then preprocessed by two 1D-Conv (convolution in the log-frequency dimension) with skip-connections [54], followed by four 1D-Conv layers. As in [17], we apply a non-linear leaky-ReLU (slope 0.3) [55] and dropout (rate 0.2) [56] between each convolutional layer. Importantly, the kernel size and padding of each of these layers are chosen so that the frequency resolution is never reduced. We found in practice that it helps the model to distinguish close but different

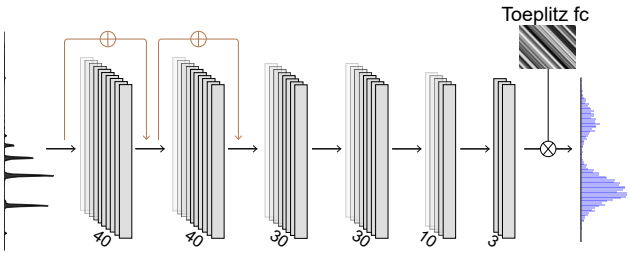


Figure 3. Architecture of our network f_θ. The number of channels varies between the intermediate layers, however the frequency resolution remains unchanged until the final Toeplitz fully-connected layer.

pitches. The output is then flattened, fed to a final fully-connected layer and normalized by a softmax layer to become a probability distribution of the desired shape.

Note that all layers (convolutions [6] , elementwise non-linearities, layer-norm and softmax), except the last final fully-connected layer, preserve transpositions. To make the final fully-connected layer also transposition-equivariant, we propose to use **Toeplitz fully-connected layers**. It simply consists of a standard linear layer without bias but whose weights matrix A is a Toeplitz matrix, i.e. each of its diagonals is constant.

$$A = \begin{pmatrix} a_0 & a_{-1} & a_{-2} & \cdots & a_{-n+2} & a_{-n+1} \\ a_1 & a_0 & a_{-1} & \ddots & \ddots & a_{-n+2} \\ a_2 & a_1 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ a_{m-1} & \cdots & \cdots & \cdots & \cdots & a_{m-n} \end{pmatrix} \tag{10}$$

Contrary to arbitrary fully-connected layers, Toeplitz matrices are transposition-preserving operations and only have $m + n - 1$ parameters instead of mn. Furthermore, they are mathematically equivalent to convolutions, making them straightforward to implement.

[5] Specifically, we sample an integer k uniformly from the range $\{-k_{max}, \ldots, k_{\max}\}$, then generate two CQT outputs, denoted as x and $x^{(k)}$, where x is obtained by cropping the input CQT at indices $[k_{\max}, K - k_{\max} - 1]$, and $x^{(k)}$ is obtained by cropping the input CQT at indices $[k_{\max} - k, K - k_{\max} + k - 1]$, with K the total number of bins of the original CQT frame and $k_{\max}$ = 16 in practice (see Figure 2).

[6] Convolutions roughly preserve transpositions since the kernels are applied locally, meaning that if two transposed inputs are convolved by the same kernel, then the output results will be almost transpositions of each other as well

Model	# params	Trained on	Raw Pitch Accuracy	
			MIR-1K	*MDB-stem-synth*
SPICE [19]	2.38M	private data	90.6%	89.1%
DDSP-inv [45]	-	*MIR-1K / MDB-stem-synth*	91.8%	88.5%
PESTO (ours)	28.9k	*MIR-1K*	**96.1%**	94.6%
PESTO (ours)	28.9k	*MDB-stem-synth*	93.5%	**95.5%**
CREPE [16]	22.2M	many (supervised)	**97.8%**	**96.7%**

Table 1. Evaluation results of PESTO compared to supervised and self-supervised baselines. CREPE has been trained in a supervised way on a huge dataset containing in particular *MIR-1K* and *MDB-stem-synth*. It is grayed out as a reference. For DDSP-inv, we report the results when training and evaluating on the same dataset.

3.5 Absolute pitch inference from $\mathbf{y}$

Our encoder f_θ returns a probability distribution over (quantized) pitches. From an input CQT frame $\mathbf{x}$, we first compute the probability distribution $f_\theta(\mathbf{x})$, then we infer the absolute pitch $\hat{p}$ by applying the affine mapping:

$$\hat{p}(\mathbf{x}) = \frac{1}{b}\left(\arg\max f_\theta(\mathbf{x}) + p_0\right) \tag{11}$$

where $b = 3$ is the number of bins per semitones in the CQT and p_0 is a fixed integer shift that only depends on f_θ. As in [19], we set the integer shift p_0 by relying on a set of synthetic data [7] with known pitch.

4. EXPERIMENTS

4.1 Datasets

To evaluate the performance of our approach, we consider the two following datasets:

1. *MIR-1K* [57] contains 1000 tracks (about two hours) of people singing Chinese pop songs, with separate vocal and background music tracks provided.
2. *MDB-stem-synth* [58] contains re-synthesized monophonic music played by various instruments.

The pitch range of the *MDB-stem-synth* dataset is wider than the one of *MIR-1K*. The two datasets have different sampling rates and granularity for the annotations.

We conduct separate model training and evaluation on both datasets to measure overfitting and generalization performance. In fact, given that our model is lightweight and does not require labelled data, overfitting performance is particularly relevant for real-world scenarios, as it is easy for someone to train on their own dataset, e.g. their own voice. However, we also examine generalization performance through cross-evaluation to ensure that the model truly captures the underlying concept of pitch and does not merely memorize the training data.

4.2 Training details

From an input CQT (see part 3.2), we first compute the pitch-shifted CQT (see part 3.3). Then two random data augmentations $t_1, t_2 \sim \mathcal{T}$ are applied with a probability of 0.7. We used white noise with a random standard deviation between 0.1 and 2, and gain with a random value picked uniformly between -6 and 3 dB. The overall architecture of f_θ (see part 3.4) is implemented in PyTorch [59]. For training, we use a batch size of 256 and the Adam optimizer [60] with a learning rate of 10^{-4} and default parameters. The model is trained for 50 epochs using a cosine annealing learning rate scheduler. Our architecture being extremely lightweight, training requires only 545MB of GPU memory and can be performed on a single GTX 1080Ti.

4.3 Performance metrics

We measure the performances using the following metrics.

1. *Raw Pitch Accuracy* (RPA): corresponds to the percentage of voiced frames whose pitch error [8] is less than 0.5 semitone [61].
2. *Raw Chroma Accuracy* (RCA): same as RPA but considering the mapping to Chroma (hence allowing octave errors) [61].

RCA is only used in our ablation studies.

4.4 Results and discussions

4.4.1 Clean signals

We compare our results with three baselines: CREPE [16], SPICE [19] and DDSP-inv [45]. CREPE is fully-supervised while SPICE and DDSP-inv are two SSL approaches. To measure the influence of the training set, we train PESTO on the two datasets (*MIR-1K* and *MDB-stem-synth*) and also evaluate on the two. This allows to test model generalization.

We indicate the results in Table 1. We see that PESTO significantly outperforms the two SSL baselines (SPICE and DDSP-inv) even in the cross-dataset scenario (93.5% and 94.6%). Moreover, it is competitive with CREPE (-1.7% and -1.2%) which has 750 times more parameters and is trained in a supervised way on the same datasets.

4.4.2 Robustness to background music

Background noise and music can severely impact pitch estimation algorithms, making it imperative to develop robust methods that can handle real-world scenarios where background noise is often unavoidable.

We therefore test the robustness of PESTO to background music. For this, we use the *MIR-1K* dataset, which contains separated vocals and background tracks

[7] synthetic harmonic signals with random amplitudes and pitch

[8] i.e. distance between the predicted pitch and the actual one

	Raw Pitch Accuracy (*MIR-1K*)			
Model	clean	20 dB	10 dB	0 dB
SPICE [19]	91.4%	91.2%	90.0%	81.6%
PESTO				
$\beta = 0$	**94.8%**	90.7%	79.2%	50.0%
$\beta = 1$	94.5%	94.2%	92.9%	**83.1%**
$\beta \sim \mathcal{U}(0, 1)$	94.7%	94.4%	92.9%	81.7%
$\beta \sim \mathcal{N}(0, 1)$	**94.8%**	**94.5%**	**93.0%**	82.6%
$\beta \sim \mathcal{N}(0, \frac{1}{2})$	**94.8%**	**94.5%**	92.9%	81.0%
CREPE [16]	**97.8%**	**97.3%**	**95.3%**	**84.8%**

Table 2. Robustness of PESTO and other baselines to background music with various Signal-to-Noise ratios. Adding background music to training samples significantly improves the robustness of PESTO (see section 4.4.2).

and allows testing various signal-to-noise (here vocal-to-background) ratios (SNRs).

We indicate the results in Table 2. As foreseen, the performance of PESTO when trained on clean vocals (row $\beta = 0$) and applied to vocal-with-background considerably drop: from 94.8% (clean) to 50.0% (SNR = 0 dB) [9].

To improve the robustness to background music, we slightly modify our method to train our model on mixed sources. Instead of using gain and white noise as data augmentations, we create an augmented view of our original vocals signal $\mathbf{x}_{\text{vocals}}$ by mixing it (in the complex-CQT domain) with its corresponding background track $\mathbf{x}_{\text{background}}$:

$$\mathbf{x} = \mathbf{x}_{\text{vocals}} + \beta \mathbf{x}_{\text{background}} \tag{12}$$

Then, thanks to $\mathcal{L}_{\text{inv}}$, the model is trained to ignore the background music for making its predictions.

The background level β is randomly sampled for each CQT frame. The influence of the distribution we sample β from is depicted in Table 2. This method significantly limits the drop in performances observed previously and also makes PESTO outperform SPICE in noisy conditions.

4.4.3 Ablation study

Table 3 depicts the influence of our different design choices. First, we observe that the equivariance loss $\mathcal{L}_{\text{equiv}}$ and the final Toeplitz fully-connected layer (eq.(10)) are absolutely essential for our model not to collapse. Moreover, data augmentations seem to have a negligible influence on out-of-domain RPA (-0.2%) but slightly help when training and evaluating on the same dataset (+1.2%).

On the other hand, it appears that both $\mathcal{L}_{\text{inv}}$ and $\mathcal{L}_{\text{SCE}}$ do not improve in-domain performances but help the model to generalize better. This is especially true for $\mathcal{L}_{\text{SCE}}$, whose addition enables to improve RPA from 86.9% to 94.6% on *MDB-stem-synth*.

Finally, according to the drop of performances in RPA and RCA when removing $\mathcal{L}_{\text{inv}}$, it seems that the invariance loss prevents octave errors on the out-of-domain dataset.

[9] It should be noted that the difference between the 96.1% of Table 1 and the 94.8% of Table 2 is due to the fact that we do not apply any data augmentation (gain or additive white noise) when $\beta = 0$.

	MIR-1K		MDB	
	RPA	RCA	RPA	RCA
PESTO baseline	96.1%	96.4%	94.6%	95.0%
Loss ablations				
w/o $\mathcal{L}_{\text{equiv}}$	5.8%	8.6%	1.3%	6.1%
w/o $\mathcal{L}_{\text{inv}}$	96.1%	96.4%	92.5%	94.5%
w/o $\mathcal{L}_{\text{SCE}}$	96.1%	96.5%	86.9%	93.8%
Miscellaneous				
no augmentations	94.8%	95.4%	94.8%	95.2%
non-Toeplitz fc	5.7%	8.7%	1.2%	6.1%

Table 3. Respective contribution of various design choices of PESTO for a model trained on *MIR-1K*.

5. CONCLUSION

In this paper, we presented a novel self-supervised learning method for pitch estimation that leverages equivariance to musical transpositions. We propose a class-based equivariant objective that enables Siamese networks to capture pitch information from pairs of transposed inputs accurately. We also introduce a Toeplitz fully-connected layer to the architecture of our model to facilitate the optimization of this objective. Our method is evaluated on two standard benchmarks, and the results show that it outperforms self-supervised baselines and is robust to background music and domain shift.

From a musical perspective, our lightweight model is well-suited for real-world scenarios, as it can run on resource-limited devices without sacrificing performance. Moreover, its SSL training procedure makes it convenient to fine-tune on a small unlabeled dataset, such as a specific voice or instrument. Additionally, the resolution of the model is a sixth of a tone but could eventually be increased by changing the resolution of the CQT. Moreover, despite modelling pitch estimation as a classification problem, we make no assumption about scale or temperament.

These features make our method still a viable solution, e.g. for instruments that use quartertones and/or for which no annotated dataset exists. We therefore believe that it has many applications even beyond the limitations of Western music.

Overall, the idea of using equivariance to solve a classification problem is a novel and promising approach that enables the direct return of a probability distribution over the classes with a single, potentially synthetic, labelled element. While our paper applies this approach to pitch estimation, there are other applications where this technique could be useful, such as tempo estimation.

Moreover, modelling a regression task as a classification problem can offer greater interpretability as the output of the network is not a single scalar but a whole probability distribution. Finally, it can generalize better to multi-label scenarios.

Our proposed method hence demonstrates the potential of using equivariance to solve problems that are beyond the scope of our current work. In particular, it paves the way towards self-supervised multi-pitch estimation.

6. ACKNOWLEDGEMENTS

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 the reviewers and meta-reviewer for their valuable and insightful comments.

7. REFERENCES

[1] D. Talkin, "A robust algorithm for pitch tracking (RAPT)," pp. 495–518, 1995.

[2] P. Boersma, "Acurate short-term analysis of the fundamental frequency and the harmonics-to-noise ratio of a sampled sound," *IFA Proceedings 17*, vol. 17, pp. 97–110, 1993. [Online]. Available: http://www.fon.hum.uva.nl/paul/papers/Proceedings_1993.pdf

[3] 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)*, vol. 1, no. 1, 2014, pp. 659–663.

[4] A. Camacho and J. G. Harris, "A sawtooth waveform inspired pitch estimator for speech and music," *The Journal of the Acoustical Society of America*, vol. 124, no. 3, pp. 1638–1652, 2008.

[5] B. S. Lee and D. P. W. Ellis, "Noise robust pitch tracking by subband autocorrelation classification," in *Proc. Interspeech 2012*, 2012, pp. 707–710.

[6] K. Han and D. Wang, "Neural Network Based Pitch Tracking in Very Noisy Speech," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 12, pp. 2158–2168, 2014.

[7] 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*. International Society for Music Information Retrieval, oct 2018, pp. 50–57. [Online]. Available: https://archives.ismir.net/ismir2018/paper/000019.pdf

[8] J. W. Kim and J. P. Bello, "Adversarial learning for improved onsets and frames music transcription," in *Proceedings of the 20th International Society for Music Information Retrieval Conference, ISMIR 2019*. International Society for Music Information Retrieval, jun 2019, pp. 670–677. [Online]. Available: https://archives.ismir.net/ismir2019/paper/000081.pdf

[9] 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, oct 2021. [Online]. Available: https://arxiv.org/abs/2010.01815v3

[10] G. Song, Z. Wang, F. Han, and S. Ding, "Transfer learning for music genre classification," *IFIP Advances in Information and Communication Technology*, vol. 510, pp. 183–190, 2017.

[11] 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. 2018; 1 (1): 4-21.*, 2018.

[12] 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.

[13] V. Lostanlen and C. E. Cella, "Deep convolutional networks on the pitch spiral for music instrument recognition," in *Proceedings of the 17th International Society for Music Information Retrieval Conference, ISMIR 2016*. International Society for Music Information Retrieval, may 2016, pp. 612–618. [Online]. Available: https://archives.ismir.net/ismir2016/paper/000093.pdf

[14] Y. Han, J. Kim, and K. Lee, "Deep Convolutional Neural Networks for Predominant Instrument Recognition in Polyphonic Music," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 1, pp. 208–221, 2017.

[15] A. Solanki and S. Pandey, "Music instrument recognition using deep convolutional neural networks," *International Journal of Information Technology*, vol. 14, no. 3, pp. 1659–1668, 2022. [Online]. Available: https://doi.org/10.1007/s41870-019-00285-y

[16] J. W. Kim, J. Salamon, P. Li, and J. P. Bello, "Crepe: A Convolutional Representation for Pitch Estimation," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, vol. 2018-April, feb 2018, pp. 161–165. [Online]. Available: http://arxiv.org/abs/1802.06182

[17] C. Weiß and G. Peeters, "Deep-Learning Architectures for Multi-Pitch Estimation: Towards Reliable Evaluation," feb 2022. [Online]. Available: http://arxiv.org/abs/2202.09198

[18] R. Geirhos, J. H. Jacobsen, C. Michaelis, R. Zemel, W. Brendel, M. Bethge, and F. A. Wichmann, "Shortcut learning in deep neural networks," *Nature Machine Intelligence*, vol. 2, no. 11, pp. 665–673, apr 2020. [Online]. Available: https://www.nature.com/articles/s42256-020-00257-z

[19] 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, pp. 1118–1128, oct 2020. [Online]. Available: https://dl.acm.org/doi/pdf/10.1109/TASLP.2020.2982285

[20] E. Quinton, "Equivariant Self-Supervision for Musical Tempo Estimation," *Proceedings of the 23rd International Society for Music Information Retrieval Conference, ISMIR 2022*, sep 2022. [Online]. Available: https://archives.ismir.net/ismir2022/paper/000009.pdf

[21] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 1735–1742, 2006.

[22] 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. [Online]. Available: http://proceedings.mlr.press/v119/chen20j/chen20j.pdf

[23] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny, "Barlow Twins: Self-Supervised Learning via Redundancy Reduction," *38th International Conference on Machine Learning, ICML 2021*, mar 2021. [Online]. Available: http://proceedings.mlr.press/v139/zbontar21a/zbontar21a.pdf

[24] 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, 2022. [Online]. Available: https://openreview.net/forum?id=xm6YD62D1Ub

[25] 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. [Online]. Available: https://proceedings.mlr.press/v119/wang20k/wang20k.pdf

[26] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum Contrast for Unsupervised Visual Representation Learning," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, nov 2020, pp. 9726–9735. [Online]. Available: https://openaccess.thecvf.com/content_CVPR_2020/papers/He_Momentum_Contrast_for_Unsupervised_Visual_Representation_Learning_CVPR_2020_paper.pdf

[27] X. Chen, H. Fan, R. Girshick, and K. He, "Improved Baselines with Momentum Contrastive Learning," mar 2020. [Online]. Available: http://arxiv.org/abs/2003.04297

[28] J. B. Grill, F. Strub, F. Altché, 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. [Online]. Available: https://papers.nips.cc/paper/2020/file/f3ada80d5c4ee70142b17b8192b2958e-Paper.pdf

[29] 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. [Online]. Available: https://openaccess.thecvf.com/content/CVPR2021/papers/Chen_Exploring_Simple_Siamese_Representation_Learning_CVPR_2021_paper.pdf

[30] Y. Dubois, T. Hashimoto, S. Ermon, and P. Liang, "Improving Self-Supervised Learning by Characterizing Idealized Representations," sep 2022. [Online]. Available: https://openreview.net/pdf?id=agQGDz6gPOo

[31] 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. [Online]. Available: https://arxiv.org/abs/2010.10915v1

[32] J. Anton, H. Coppock, P. Shukla, and B. W. Schuller, "Audio Barlow Twins: Self-Supervised Audio Representation Learning," sep 2022. [Online]. Available: http://arxiv.org/abs/2209.14345

[33] 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. [Online]. Available: https://dl.acm.org/doi/pdf/10.1109/TASLP.2022.3221007

[34] 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," *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pp. 776–780, 2017.

[35] 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. [Online]. Available: https://archives.ismir.net/ismir2021/paper/000084.pdf

[36] R. Dangovski, L. Jing, C. Loh, S. Han, A. Srivastava, B. Cheung, P. Agrawal, and M. Soljačić, "Equivariant Contrastive Learning," *ICLR 2022 - 10th International Conference on Learning Representations*, oct 2022. [Online]. Available: https://openreview.net/pdf?id=gKLAAfiytI

[37] R. Winter, M. Bertolini, T. Le, F. Noé, and D.-A. Clevert, "Unsupervised Learning of Group Invariant and Equivariant Representations," feb 2022. [Online]. Available: https://openreview.net/pdf?id=47lpv23LDPr

[38] Q. Garrido, L. Najman, and Y. Lecun, "Self-supervised learning of Split Invariant Equivariant representations," feb 2023. [Online]. Available: https://openreview.net/pdf?id=2sIVxJ9Hp0

[39] A. M. Noll, "Cepstrum pitch determination." *The Journal of the Acoustical Society of America*, vol. 41 2, pp. 293–309, 1967.

[40] J. J. Dubnowski, R. W. Schafer, and L. R. Rabiner, "Real-Time Digital Hardware Pitch Detector," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 24, no. 1, pp. 2–8, 1976.

[41] M. J. Ross, H. L. Shaffer, A. Cohen, R. L. Freudberg, and H. Manley, "Average magnitude difference function pitch extractor," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 22, pp. 353–362, 1974.

[42] A. de Cheveigné and H. Kawahara, "YIN, a fundamental frequency estimator for speech and music," *The Journal of the Acoustical Society of America*, vol. 111, no. 4, pp. 1917–1930, 2002.

[43] Y.-J. Luo, K. W. Cheuk, T. Nakano, M. Goto, and D. Herremans, "Unsupervised disentanglement of pitch and timbre for isolated musical instrument sounds," in *International Society for Music Information Retrieval Conference*, 2020.

[44] K. Tanaka, Y. Bando, K. Yoshii, and S. Morishima, "Unsupervised disentanglement of timbral, pitch, and variation features from musical instrument sounds with random perturbation," in *2022 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2022, pp. 709–716.

[45] J. Engel, R. Swavely, A. Roberts, L. . Hanoi, . Hantrakul, and C. Hawthorne, "Self-Supervised Pitch Detection by Inverse Audio Synthesis," *Workshop on Self-Supervision in Audio and Speech at the 37th International Conference on Machine Learning (ICML 2020)*, pp. 1–9, 2020. [Online]. Available: https://goo.gl/magenta/ddsp-inv

[46] J. Engel, L. Hantrakul, C. Gu, and A. Roberts, "DDSP: Differentiable Digital Signal Processing," *The Eighth International Conference on Learning Representations, ICLR 2020*, jan 2020. [Online]. Available: https://openreview.net/pdf?id=B1x1ma4tDr

[47] P. J. Huber, "Robust Estimation of a Location Parameter," *The Annals of Mathematical Statistics*, vol. 35, no. 1, pp. 73–101, 1964. [Online]. Available: http://www.jstor.org/stable/2238020

[48] Z. Chen, V. Badrinarayanan, C. Y. Lee, and A. Rabinovich, "GradNorm: Gradient normalization for adaptive loss balancing in deep multitask networks," in *35th International Conference on Machine Learning, ICML 2018*, vol. 2. International Machine Learning Society (IMLS), nov 2018, pp. 1240–1251. [Online]. Available: http://proceedings.mlr.press/v80/chen18a/chen18a.pdf

[49] P. Esser, R. Rombach, and B. Ommer, "Taming transformers for high-resolution image synthesis," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, dec 2021, pp. 12 868–12 878. [Online]. Available: https://openaccess.thecvf.com/content/CVPR2021/papers/Esser_Taming_Transformers_for_High-Resolution_Image_Synthesis_CVPR_2021_paper.pdf

[50] J. MacGlashan, E. Archer, A. Devlic, T. Seno, C. Sherstan, P. R. Wurman, and P. Stone, "Value Function Decomposition for Iterative Design of Reinforcement Learning Agents," jun 2022. [Online]. Available: https://openreview.net/pdf?id=pNEisJqGuei

[51] 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, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, vol. 2022-May. Institute of Electrical and Electronics Engineers Inc., mar 2022, pp. 781–785. [Online]. Available: https://arxiv.org/abs/2203.09893v2

[52] 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.

[53] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer Normalization," jul 2016. [Online]. Available: http://arxiv.org/abs/1607.06450

[54] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-Decem, dec 2016, pp. 770–778. [Online]. Available: https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/He_Deep_Residual_Learning_CVPR_2016_paper.pdf

[55] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical Evaluation of Rectified Activations in Convolutional Network," may 2015. [Online]. Available: http://arxiv.org/abs/1505.00853

[56] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of*

Machine Learning Research, vol. 15, pp. 1929–1958, 2014.

[57] 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.

[58] J. Salamon, R. M. Bittner, J. Bonada, J. J. Bosch, E. Gómez, and J. P. Bello, "An analysis/synthesis framework for automatic f0 annotation of multitrack datasets," *Proceedings of the 18th International Society for Music Information Retrieval Conference, ISMIR 2017*, pp. 71–78, 2017.

[59] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, vol. 32. Neural information processing systems foundation, dec 2019. [Online]. Available: https://pytorch.org/

[60] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, dec 2015. [Online]. Available: http://arxiv.org/abs/1412.6980

[61] G. E. Poliner, D. P. Ellis, A. F. Ehmann, E. Gómez, S. Streich, and B. Ong, "Melody transcription from music audio: Approaches and evaluation," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 4, pp. 1247–1256, 2007.

THE GAMES WE PLAY: EXPLORING THE IMPACT OF ISMIR ON MUSICOLOGY

Vanessa Nina Borsan **Mathieu Giraud**
Univ. Lille, CNRS, Centrale Lille
UMR 9189 CRIStAL, F-59000 Lille, France
{vanessa,mathieu}@algomus.fr

Richard Groult
Univ Rouen Normandie, INSA Rouen Normandie,
Université Le Havre Normandie, Normandie Univ
LITIS UR 4108, F-76000 Rouen, France
richard.groult@univ-rouen.fr

ABSTRACT

Throughout history, a consistent temporal and spatial gap has persisted between the inception of novel knowledge and technology and their subsequent adoption for extensive practical utilization. The article explores the dynamic interaction and exchange of methodologies between musicology and computational music research. It focuses on an analysis of ten years' worth of papers from the International Society for Music Information Retrieval (ISMIR) from 2012 to 2021. Over 1000 citations of ISMIR papers were reviewed, and out of these, 51 later works published in musicological venues drew from the findings of 28 ISMIR papers. Final results reveal that most contributions from ISMIR rarely make their way to musicology or humanities. Nevertheless, the paper highlights four examples of successful knowledge transfers between the fields and discusses best practices for collaborations while addressing potential causes for such disparities. In the epilogue, we address the interlaced origins of the problem as stemming from the language of new media, institutional restrictions, and the inability to engage in multidisciplinary communication.

1. INTRODUCTION

In 2005, Cook [1] critically addressed the prospects and difficulties of collaborations between Music Information Retrieval (MIR) and musicology, many of which were revisited by Downie in 2009, further examining their implications and potential advancements [2]. With the emergence of empirical research methods and advancements in technology, music research has encompassed multiple academic fields, leading to a transformation in the structures of these disciplines, including Music Information Retrieval (MIR) and contemporary musicology. Given their multidisciplinary nature, the categorization of either is becoming increasingly arbitrary. However, for the purpose of the clarity of further arguments in this paper, we classify "traditional" and humanities-centred music research fields (musicology, music theory, ethnomusicology, etc.) under the umbrella term "musicology." Conversely, we use the term "MIR" to encompass all fields that engage in natural-sciences-based (typically computational) research related to music, such as acoustics, informatics, physics, mathematics, engineering, and more [1].

Despite the significant impact of both fields in broadening our understanding of music, unresolved issues highlighted by Cook continue to hinder their collaboration to this day [1]. In recent years, a growing number of musicologists, along with humanities researchers in general, have shown a preference for working with digital materials rather than physical ones [3], but the application of computation to research can be approached at various levels. There are *general-purpose software*, such as word processors or spreadsheet editors, and *music-oriented software*, such as Sibelius, Finale, and Audacity; there are *programming music/MIR platforms and libraries*, such as Humdrum [4], music21 [5], Librosa [6] and Essentia [7] and then there are *methods and algorithms* as developed by the MIR community, for example [8–11] and others (see [12] for a detailed review). While computer usage is prevalent among many researchers there are fewer musicologists who adopt or contribute to similar methodologies. However, through new media and computational advancements, music and our relationship to it are changing [13]. Given the expansion of what is deemed significant in the "realm of music," it raises the question of whether familiarity with computational languages is becoming a prerequisite for its exploration.

Computational methods assist researchers in handling larger and more varied datasets, but, would musicologists agree that "working with [these] datasets [have] open[ed] up new areas of musicology?" [1] Or, has this shift evoked new areas of research, which are (almost) independent from the musicological domain? The goal of this paper is to ask *to what extent the MIR contributions (in the frames of ISMIR) resonate throughout the musicological community*. The very results of these particular analyses may also

© VN. Borsan, M. Giraud, and R. Groult. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).
Attribution: VN. Borsan, M. Giraud, and R. Groult, "The Games We Play: Exploring The Impact of ISMIR on Musicology", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

[1] Our labels are arbitrary categories for rough orientation, considering disciplines like music cognition that fit into both/neither category.

highlight some of the core issues of miscommunication between the two domains.

We start by outlining the "ready-made" arguments of collaboratory issues of the fields (Section 1), followed by a methodology introduction for a bibliographical study of ten years of ISMIR papers (2010–2021) and their citations, where we present some empirical results (Sections 2 and 3). As an example of "good collaboratory practice", we detail four examples, where ISMIR findings were later used by/for a musicological audience through datasets, methodologies, and tool and/or code (Section 4). In conclusion, we discuss the results, examine potential reasons for the outcomes, and draw on theories of play and media studies to support our findings (Section 5).

2. THE WEB OF ARGUMENTS

Numerous authors have explored the advantages and/or drawbacks of interdisciplinary research in the realm of music. We acknowledge the tensions within musicology without delving into the detailed evolution of historic musicology, ethnomusicology, and systematic (empirical) musicology, as these topics have already been extensively covered([14]). We focus on the development of (pro and con) arguments, generally raised in the *2000s* and *2010s*.

2.1 Years 2000–2010: Enticement Versus Restraint

The critical discussions began with the emergence of more empirically-centred approaches, mostly labelled as systematic musicology. Following the iconic question "Who stole systematic musicology?", Leman [15] observed, that even systematic musicology had no longer belonged to "itself." Conversely, transdisciplinary musicology gained traction among engineering departments (MIR, sound processing), as and neuroscientists and psychologists, who developed a growing interest in the study of *music*.

Amidst the rapid growth of music-related technology production, papers in the early 2000s addressed contemporary musicology, its redefinition, and future methodologies and goals. For some, technologies were viewed as a natural extension for quantitative, big-data, and empirical music analyses [1], while others thought of music research as an interdisciplinary ground of "somewhat equal" subdisciplines, including musicology and MIR. Addressing the benefits of these collaborations, [16, 17], many authors highlighted the benefits of multidisciplinary projects in expanding the boundaries of isolated disciplines for more comprehensive outcomes. In contrast, others warned that in "an era in which interdisciplinarity has become a kind of mantra, verbally subscribed to by nearly everyone, disciplines continue to police their own boundaries [18]." A similar opinion was shared by Parncutt [14], and Leman, who stressed that, even though they like talking about interdisciplinary projects, "it was very rare that researchers went beyond the boundaries of their own disciplines [15]." Additionally, knowledge transfers are anything but fluid among computational scientists and musicologists, thus the ideas expand poorly, if at all [19], hence, they must be improved [12]. The scepticism towards unconditionally welcoming the emerging collaboratory changes thus remained. In 2009, [2] reflected on interdisciplinary dynamics during the first 10 years of ISMIR, highlighting its shortcomings, such as the inability to communicate the produced tools to the user (performer, musicologists, ...), favouring low-level over high-level features and audio over other symbolic music representations, and so forth.

2.2 Years 2010–Today: The Quest for Consensus

The scepticism and critiques were not far-fetched nor properly addressed, as Urberg later noticed that the methodological visions of "fundamentally-renewed" music research, had "not [yet] taken over the majority of musicological scholarship [20]." Nonetheless, he imposed that the methodology of research has already shifted, as there is an ascending trend of new research tools and digitized (music) data representations, a lot of them consciously used by musicologists. So what seems to be the problem?

Finding balance in methodology, data collection and interpretation. Still in the second decade of the 21st century, when the introduced arguments began to overlap, Inskip et. al. [25] conducted a survey in order to answer this question. The study suggests that "[...] efforts should be made into supporting the development of their digital skills and providing usable, useful and reliable software created with a 'musicology-centred' design approach." Otherwise, the "data richness will lead to information overload [26]." As Dahling expressed in 2012, there are many tools for music collection and analysis, of which many "suffer from various shortcomings, such as specificity to a certain repertoire or approach, lack of robustness and flexibility, flawed user interfaces, or output is difficult to interpret [26]." A similar concern has been expressed by others, such as [27] and [28], or, for textual analysis [29]. All of them advocate not only for a more *accessible and flexible computational methods*, but also express the need understand *what these methods do and how*. Alongside epistemological confusion and other (methodological) drawbacks, a similar problem was stressed by Aucouturier and Bigand. Their dialogue-style paper revealed the flaws and prospects for collaborations between MIR and music research (specifically music cognition) [30]. In Drucker's words, "the humanities are not a mere afterthought, simply studying and critiquing the effects of computational methods. [Their theory] can provide ways of thinking differently [31]." In a different light, the latter was also implied by [32].

Cyclical collaboration vs discontinuity. Following Downie's call for improvements [2], some authors discussed *refined measurements* that need to be considered regarding data collection and interpretation, for "obtaining or accessing *high-quality datasets* remains a serious hurdle, especially on a large scale [33]." These hurdles limit the (digital) quality of music research, but not only that. All

Claim/link to musicology	ISMIR papers	Examples of claim
None, "musicolog" is only present in one of the references	147 (42.7%)	
Application of musicological concepts, by only explaining citation, or apply musicological concepts, or hinting towards the possibility of musicological application.	81 (23.6%)	[21] "In order to select relevant low-level features, we refer to *musicology* papers such as [...] which suggest that arousal is related to features including rhythm density, note density, key, dynamic, tempo, etc." [22] "We assume that the music tradition is known, and that the rhythm class (tāḷa) of the piece is from a set of known (from *musicological* literature) tāḷas."
Some claim of musicological utility.	114 (33.7%)	[23] "[...] retaining the rest of the presented framework, e.g. for an analytical ontology of musicological terms supporting the use of digital score annotations to illustrate points in scholarly musicological arguments." (see Section 4) [24] "These features can serve as inputs to machine learning algorithms, or they can be analyzed statistically to derive musicological insights." (see Section 4)

Table 1. Links and/or claims regarding musicology in 342 ISMIR papers from 2012 to 2021 where "musicolog" occurs.

music cannot be collected and/or represented in the same manner, and it is not feasible to investigate and discuss it within identical methodological frameworks [28,34]. They believe that this perspective should be considered not only by musicologists but should also be of equal importance for the field of MIR. Schüler and Huron argued that mutual *theoretical awareness* is essential for musicologists and MIR researchers [19,35]. Methodological tools should not be confused with philosophical worldviews [35], and due to the importance of theory *and* "practice", there must exist a *cyclical collaboration* between the disciplines [27]. Humanities scholars express concern about detached interpretation and the prioritization of "facts" and algorithmic success in studies [28, 29]. Thus, the algorithms must be transparent enough for the scholars to actively participate in the building blocks of their framework and methods. "[I]n the long run, the most 'useful' computational analyses will be the ones which are interactive, confronting a human user with the results of computational analysis and allowing that user to modify or intervene in the procedure to arrive at an acceptable or interesting result [28]."

From a more critical standpoint, Becker asks whether "our failure [is] due to our own shortcomings in not becoming thoroughly versed in the protocols and expectations of another discipline? Or, was the failure due to too stringent protocols and expectations for publication in a [...] journal?", concluding that some disciplinary barriers may be unbreachable due to rigid institutional formations [18]. Leman, conversely, sees the "failure" of collaboration in the notion of the absence of "concrete planned goal at long term, except some vague idea of what all these research activities are up to [15]." Although no firm solutions have been introduced, some humanities authors [29, 36, 37] offered partial theoretical frameworks. Our methodology, inspired by the latter (e.g., Moretti's *Distant Reading*), will be introduced in the following section.

3. METHODOLOGY AND RESULTS

In this section, we discuss the filtering process of ISMIR 2012–21 to examine *whether* and *how* such papers were used in musicological studies. We also provide statistics and information on data availability.

3.1 Article Selection and Filtration: *Which papers claim to have some musicological utility?*

We downloaded all 1055 ISMIR papers [2] from the past 10 years (2012–2021) [3] and converted the .pdf files to .txt files. We retrieved 342 articles which included the root "musicolog", meaning the article contained words like "musicological," "musicology", and "ethnomusicologist" [4]. Next, we reviewed these 342 papers to determine their musicological implications, categorizing them into 3 categories (see Table 1 for examples and details). Subsequently, we focused on the 114 ISMIR papers that claimed some musicological relevance and the citations, if any.

3.2 Citations Analysis: *Were the papers later used "in musicology"?*

To study how and if these 114 papers may have had an impact on musicology, we identified 907 citations of them through Google Scholar. The median of all citations per cited paper is 16. The most cited paper was cited 208 times, while 10 were never cited. We retrieved almost all of these citations [5] and sorted the citing papers by these two (slightly ambivalent) categories.

① *Is any "citator" a musicologist?* As "musicologists", we classified researchers with a Master's or PhD degree in a "musicological" research field or most of their activity was mostly conducted in a musicological environment (see Introduction). Together, there were *210 citations to 67 unique ISMIR papers* that corresponded with this category.

② *Does the citing paper appear in a musicological journal/conference?* Here, we focus on venues instead of in-

[2] `https://www.ismir.net/conferences/`

[3] Due to time constraints, we couldn't thoroughly analyze all ISMIR papers. Instead, we focused on the impact of early 2000s ideas on the MIR and musicology collaboration, exploring new tools, and acknowledging changes due to improved technology and online publication accessibility.

[4] We acknowledge potential exclusions of articles using terms like "music research," "music theory," or "music history", and that ISMIR papers may hold musicological significance without explicitly stating so.

[5] About 20 were excluded due to inaccessibility of the article or lack of information, among which 6 belong to centred dataset.

Journal/Conference	Citations	(Cited ISMIR papers)
Digital Libraries for Musicology (DLfM)*	31	(15)
Journal of New Music Research (JNMR)*	17	(15)
Acta Musicologica	7	(7)
Frontiers in Digital Humanities*	6	(5)
Empirical Musicology Review (EMR)	6	(5)
Folk Music Analysis (FMA)	6	(5)
4: Musicae Scientiae; Zeitschrift der Gesellschaft für Musiktheorie; Digital Scholarship in the Humanities*; McGill University (Schulich School of Music, Music Technology*); Utrecht University (MA or PhD Thesis)*; 3: Music Theory Online (MTO); Computational Music Analysis; UC San Diego*; 2: Computational Phonogram Archiving: Current Research in Systematic Musicology*; The Musical Quarterly; Journal on Computing and Cultural Heritage*; Digital Humanities Quarterly*; +33 more (appear once)		
	70 citations in total	

Table 2. Somewhat musicology-centred journals/conferences/books/institutions, in which ISMIR papers were cited 143 times. The venues marked with (*) have both, musicology/MIR goals.

dividuals, because researchers with musicological backgrounds can have a strong root in MIR as well, while musicological journals mainly target and publish works of primarily musicologically-motivated research activity. We defined "musicological venues" by their primary motivation and targeted audience (Table 2), some of them also have (secondary) MIR motivations (* in Table 2). ISMIR was fully excluded, with the intention to show to which extent these contributions manage to "leave" the ISMIR community. Together, there were *143 citations in rather musicologically relevant publications to 55 ISMIR papers.*

From here, we focus on the 143 citations, as the rest either focused on the MIR audience only (was published in technical, science, MIR conference) or did not imply the musicological utility.

3.3 Filtered Citations Analysis: *What is the type of citation/utility?*

We sorted the 143 citations (or 114 unique citing articles) of previously mentioned 55 ISMIR papers, focusing on *if and how* the first use the latter.

✘ *Only referencing the ISMIR paper*. 92 citations only reference 43 ISMIR papers. The authors referenced the article, because it was relevant to the topic, however, their contribution was not actually used.

The other 51 citations cited *and somewhat used* 28 ISMIR contributions [6], split into the following types:

✓ *Dataset* (10 citations to 5 ISMIR papers): The author(s) of citation (partially) used the dataset, presented in cited ISMIR paper.

[6] Certain ISMIR papers were referenced and utilized in various contexts, and/or classified under multiple utility categories.

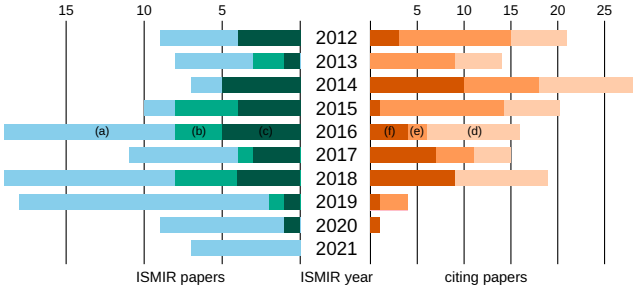


Figure 1. Distribution of the papers reported in this study among the years. Left. (a) 114 ISMIR papers with "musicolog" root and claiming to have some musicological utility; (b) from which 43 ISMIR papers *cited* in musicological venues; (c) from which 28 ISMIR papers *actually used* at least once. Right. (d) 143 citations in 51 citing papers (of the 43 ISMIR papers) from musicological venues; (e) from which 87 citations (74 unique citing papers) with at least one musicologist as an author; (f) from which 35 (or 31 unique citing papers) with *actual usage* (of the 28 ISMIR papers). Even for citations, the considered year is the year of the original ISMIR paper.

✓ *Methodology* (22 citations to 17 ISMIR papers): The author(s) of citation (partially) used the methodology, presented in cited ISMIR paper.

✓ *Code/Tool* (19 citations to 13 ISMIR papers): The author(s) of citation (partially) used the code and/or tool, presented in cited ISMIR paper.

3.4 Statistics on these Papers and Citations

About 10% of ISMIR articles mention "musicolog" every year. As expected, most recent papers are not cited (Figure 1). Despite the limited 10-year time span, papers that received at least one citation showed an average gap of three years between publication and the first citation. If we consider the 81 "older papers" published between 2012 and 2018, about the third of them have been *actually used* at least once in another study.

The list of musicological venues is also revealing (Table 2): The conference that most frequently included ISMIR's contribution was DLfM, a community that started as a satellite event of ISMIR and that "provides a forum for *musicians, musicologists, librarians, and technologists* to share findings and expertise [7]." It is followed by JNMR, which "publishes *systematic, scientific and technological research* on music, musical processes and musical behaviours, including popular, cultural and canon music" [8]. The majority of the 143 citations (see Figure 1) appear in journals/conferences with a clearly stated inclination to MIR and/or digital humanities ((*) in Table 2) and include several MIR scientists.

[7] https://dlfm.web.ox.ac.uk
[8] https://www.tandfonline.com/journals/nnmr20

Among 51 citations that used ISMIR papers, 16 papers were (partial) self-citations, meaning there was at least one common author. However, in 12 cases, new team members were involved (often from outside the initial institution), and in 4 cases, a new musicologist was present.

3.5 Data Availability

The annotated data on the 114 papers, which claim to have some musicological utility, and the one of 143 citing papers of the 28 papers are available on a git repository through open licences (Open Database License, Database Contents License) at `algomus.fr/data`.

4. FOUR EXAMPLES OF KNOWLEDGE AND IDEA TRANSFERS

51 citations in musicological venues were thus *used* one of the 28 ISMIR papers through its dataset, methodology, code and/or tool. We focused on four of these stories: In the qualitative observation, we picked examples that describe the type of utility of ISMIR contribution.

Despite some self-citations, promising collaborations were observed *within* research teams integrating interdisciplinary dynamics between musicology and MIR. These teams included both computer/MIR scientists and "conventionally" trained musicologists.

Tool: VIS Framework. In an ISMIR 2014 paper, researchers from the Distributed Digital Music Archives & Libraries Lab at McGill University introduced the VIS Framework, a Python library for music analysis together with a case study on counterpoint patterns in symbolic music scores [38]. The library was further used and cited by the same group in "musicological" venues, such as a study on encoding and translation issues published in DLfM [39]. Two PhD theses from the Schulich School of Music (McGill University) also used the framework. First proposed a computer-assisted approach to the study of interval-succession treaties [40], while second studied the tonality practice of seventeenth-century Italian composers in trio-sonatas [41] and used VIS to extract features. The VIS GUI was found to be essential in making the analysis task easier for non-computational scientists.

Dataset: The Story of Jingju. The Music Technology Group (UPF, Barcelona, Spain) includes the ethnomusicologist, Repetto. His ISMIR 2017 paper with Serra introduced JMSC, of collection of scores or Jingju (also called "Beijing Opera") [42]. Two citing DLfM 2017 papers [9] analyzed the melodic syllabic contours in JMSC [43, 44], each paper including another member of the MTG joining the two authors of the ISMIR paper.

Multidisciplinary environments have been created by MIR and music teams globally, fostering collaboration with external groups, attracting more scientists, and expanding opportunities for obtaining PhD positions from both sides. The following story exemplifies how a multidisciplinary group can attract new collaborations.

Methodology/Tool: The Lohengrin TimeMachine. An ISMIR 2017 paper by Weigl and Page, from the University of Oxford, presented an update on the MELD framework [23], used to *encode information of and about music* (e.g., digital representations of notation, audio, contextual information) inside MEI. MELD has been cited by 25 other papers. One of the "MELD applications", the Lohengrin TimeMachine was presented at DLfM in 2021 [45] by Lewis and Page, as well as Dreyfus, an American musicologist who was previously not involved with the MIR community. In his late career, he was appointed at the University of Oxford – but in the music department. The application explored a few extracts of Wagner's Lohengrin through scores, motives, orchestration, structure, texts, audio/video, musicological analysis, etc. It offers interesting representations to a wider audience of both musical knowledges but also on the very methodology of the musicological research. This citation is also a good example of the time it may take to cross domains (here, 4 years).

Tool: Mindfulness and Music Performance Study . In ISMIR 2017, researchers from IRCAM presented the PiPo plugin, designed for data stream processing in various domains including interactive audio processing and MIR. This API-based tool facilitates the extraction of low-level descriptors from audio and motion data streams [46]. A 2021 citing paper in *Psychology of Music*, from a completely independent group, in Israel, examined whether short-term mindfulness meditation activity would improve music performance (vocal skills) regarding pitch intonation, dynamics transmission, and vocal resonation [47]. They use the PiPo tool in the processing phase, using PiPo modules for the automatic segmentation of markers by onset (time-tagged frames) for low-level descriptor extraction (pitch, dynamics, timbre ...). Focusing on music psychology, this application doesn't qualify as a musicological study. However, it showcases how MIR methods can be applied to humanities-based music research. Interestingly, out of the 114 ISMIR papers examined, this is the only one reused in a "musicological" context independently of the original authors.

5. DISCUSSION AND CONCLUSION

While ISMIR is not exclusively focused on musicology, certain researchers who publish at ISMIR assert their impact on the field. Our examination of the last five years as well as a ten-year period of ISMIR reveals that the majority of these contributions seldom make their way into musicological or humanities scholarship. Out of the 28 ISMIR papers, which have been cited and used, the majority of them are partly self-cited, and/or are "re-used" within the same group, lab etc. Somehow, *we did not find a single example*

[9] DLfM was a satellite event of ISMIR at that time, meaning the papers and their citations appeared (and were likely prepared) simultaneously.

of independent musicological application of ISMIR 2012-2021 contributions in a traditional musicological journal.

We are aware that our study has some biases. To broadly observe how MIR and other music research interact, we should explore the utility dynamics both ways (ISMIR to musicology, *and* musicology to ISMIR), as well as analyze roots other than "musicolog" in multiple venues (both MIR and music) and thoroughly explore the organizers, institutions, and authors. There are also time [10] and space [11] variables, which could have had an impact on the results. Research and collaboration cannot always be measured solely by points or numbers. Non-citable research and pedagogical activities at universities are valuable components that may not be easily quantifiable. In some cases, tools or datasets may be used for inspiration without being cited in the final report. Similarly, ISMIR-presented tools may be employed without direct citation, with references made to non-ISMIR contributions or other sources.

Various technologies have undoubtedly made their way to musicologists, inspiring the creation of a quasi-common ground with IT and other domains. However, further efforts are necessary to establish a consistent circulation of knowledge. While some are managing this challenge (see Section 4), most still struggle.

This *struggle* could be understood through theories of the game (or play) by Huizinga [48] then Caillois [49]. They discuss how the games we play are not only those of "leisure" (sports, video games, ...) but also "law and order, commerce and profit, craft and art, [...] and science. All are rooted in the primaeval soil of play [48]." Caillois considers day-to-day games people play in the light of competitive examinations and economic competition [49], and his six rules very much resemble the scientific atmosphere. Like play, it is 1. *not obligatory to participate* in science, which 2. must be conducted (or "played") in an environment, *pre-defined in time and space*. 3. The strategy (research development) is left to the *individual ideas* 4. and is generally locked in an infinite loop of *"unproductivity"* (meaning, it is largely being developed and executed and re-executed within itself). Both (games and science) follow *conventional rules* and take place in 6. "make-believe" world, which is accompanied by a special *awareness of a second reality*. For example, this may as well be the daily shift from one's research to mundane events. Games (or science) can only be played *when all parties are in agreement with the particular rules*.

Several of these may be incompatible between the MIR and musicology, one of them being, as mentioned in [50], the language of new media (similar idea in [13]). As later elaborated by [51] and [52], this language has, "in the process of epochal technological change" never been immediate, but instead adopted "through a process of transition [52]." Since the majority of new technologies (or languages) for music analysis "skipped" the transitional era, and are, for an average musicologist, incomprehensible or non-intuitive (algorithmic codes), the computational products "do not manage to address them [musicologists] in an intelligible way." There seems to be a "clear disconnect between how MIR tasks are designed to evaluate systems, and how end users are supposed to use those systems [...] [making them] difficult and costly to implement [12]". Consequently, the results, produced by such processes also become unusable, as the "involvement in the wheel of algorithms is indispensable for musicological research [13, 52]." It is this kind of disruption alone, that can disable the multidisciplinary game.

Reflecting on our discussion in Section 2, Huron, imposed the obligation for both parties (MIR and musicology) to familiarize themselves with each other's methodologies [35]. Additionally, [30] highlights the importance of knowing which parts of whose methodology are to be used for a fruitful collaboration. Leman suggests solving the gap by inducing multi-modality, introducing context-based approaches into empiricism [15]; and a more reserved Parncutt, explains that the wall is set by the feeling of superiority on both sides [14], and so on. Still, the rules of the playground must first reach consensus (starting with the transition towards a common "language"). And this is where these "common grounds" come to light. ISMIR in itself is a multidisciplinary environment, however, most of the participants (deriving from natural rather than humanities or social sciences), already play by similar rules (or speak the same language). Consequently, the multidisciplinary activity within MIR remains rather limited and, despite numerous surveys [12] has yet been unable to properly address all of the (reasons for) constraints mentioned by [2] about 14 years ago. As seen in 3.4, the most cited papers in musicological venues are derived from DLfM and JNMR. This is not a coincidence, as these are "institutions", whose "rules" derive from a compromise between both disciplines, as well as the majority of yearly contributions, manage to speak the language of both. It hence makes sense, that one of the mentioned papers addressing these matters [30] is structured as a *dialogue*, as it is exactly that, finding a practical working consent among (the two) sciences, that can endorse a fertile collaboration. Merely adapting to each other's rules seems like trying to simultaneously play football and handball, where similar "material" surely cannot and will not bring a consensus between the two games. The successful examples (Section 4) and mentioned discussions, should be considered to help us advance our fundamental goals on institutional grounds and go beyond both MIR and musicology. In the process of transductive ergomimesis, "new digital media drastically reposition the people" [13] and repeatedly evoke new (motor) skills and techniques, professions, and multidisciplinary actions (see also [53]). The change is hence indispensable for the two fields, "but we've got to put in place the [institutional] conditions to make it actually happen" [1]. It seems that it is, in the end, this game (digital) musicologists may want actually want to play.

[10] The contributions we examine may be applied in the future.

[11] Some venues cannot be observed through Google Scholar, and some contributions may not have cited the source when applying their tools or databases in their research.

Acknowledgements. We thank Ken Déguernel, Louis Couturier, the Algomus team, and the anonymous reviewers for their thought-provoking remarks.

6. REFERENCES

[1] N. Cook, "Towards the compleat musicologist," in *International Conference on Music Information Retrieval (ISMIR 2005)*, 2005.

[2] J. S. Downie, D. Byrd, and T. Crawford, "Ten years of ismir: Reflections on challenges and opportunities." in *ISMIR*, 2009, pp. 13–18.

[3] T. C. Duguid, M. Feustle, F. Giannetti, and E. Grumbach, "Music scholarship online (MuSO): a research environment for a more democratic digital musicology," *Digital Humanities Quarterly*, vol. 13, no. 1, 2019.

[4] D. Huron, "Music information processing using the Humdrum toolkit: Concepts, examples, and lessons," *Computer Music Journal*, vol. 26, no. 2, pp. 11–26, 2002.

[5] M. S. Cuthbert and C. Ariza, "music21: A toolkit for computer-aided musicology and symbolic music data," *International Society for Music Information Retrieval Conference (ISMIR 2010)*, pp. 637–642, 2010.

[6] 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 *Python in Science Conference (SCIPY 2015)*, vol. 8, 2015, pp. 18–25.

[7] D. Bogdanov, N. Wack, E. Gómez Gutiérrez, S. Gulati, H. Boyer, O. Mayor, G. Roma Trepat, J. Salamon, J. R. Zapata González, X. Serra *et al.*, "Essentia: An audio analysis library for music information retrieval," in *International Society for Music Information Retrieval Conference (ISMIR 2013)*, 2013, pp. 493–498,.

[8] O. Lartillot, "Automated motivic analysis: An exhaustive approach based on closed and cyclic pattern mining in multidimensional parametric spaces," in *Computational Music Analysis*. Springer, 2016, pp. 273–302.

[9] M. Mongeau and D. Sankoff, "Comparison of musical sequences," *Computers and the Humanities*, vol. 24, no. 3, pp. 161–175, 1990.

[10] E. Cambouropoulos, "The local boundary detection model (LBDM) and its application in the study of expressive timing," in *International Computer Music Conference (ICMC 2001)*, 2001, pp. 7–22.

[11] C. Finkensiep, K. Déguernel, M. Neuwirth, and M. Rohrmeier, "Voice-leading schema recognition using rhythm and pitch features," in *International Society for Music Information Retrieval Conference (ISMIR 2020)*, 2020, pp. 520–526.

[12] 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.

[13] T. Magnusson, *Sonic writing: technologies of material, symbolic, and signal inscriptions*. Bloomsbury Publishing USA, 2019.

[14] R. Parncutt, "Systematic musicology and the history and future of Western musical scholarship," *Journal of interdisciplinary music studies*, vol. 1, no. 1, pp. 1–32, 2007.

[15] M. Leman, "Systematic musicology at the crossroads of modern music research." in *Systematic and comparative musicology: Concepts, methods, findings*, P. Lang, Ed., 2008, pp. 89–115.

[16] K. Neubarth, M. Bergeron, and D. Conklin, "Associations between musicology and music information retrieval." in *International Society for Music Information Retrieval Conference (ISMIR 2011)*, 2011, pp. 429–434.

[17] A. Volk, F. Wiering, and P. Kranenburg, "Unfolding the potential of computational musicology," in *International Conference on Informatics and Semiotics in Organisations (ICISO 2011)*, 2011, pp. 137–144.

[18] J. Becker, "Crossing boundaries: An introductory essay," *Empirical Musicology Review*, vol. 4, no. 2, pp. 45–48, 2009.

[19] N. Schüler, "Reflections on the history of computer-assisted music analysis 1: Predecessors and the beginnings," *Musicological Annual*, vol. 41, no. 1, pp. 31–43, 2005.

[20] M. Urberg, "Pasts and futures of digital humanities in musicology: Moving towards a "bigger tent"," *Music Reference Services Quarterly*, vol. 20, no. 3-4, pp. 134–150, 2017.

[21] H. H. Tan and D. Herremans, "Music fadernets: Controllable music generation based on high-level features via low-level feature modelling," in *International Society for Music Information Retrieval Conference (ISMIR 2020)*, 2020.

[22] A. Srinivasamurthy, A. Holzapfel, and X. Serra, "Informed automatic meter analysis of music recordings," in *International Society for Music Information Retrieval Conference (ISMIR 2017)*, 2017, pp. 679–685.

[23] D. Weigl and K. Page, "A framework for distributed semantic annotation of musical score:" take it to the bridge!"," in *International Society for Music Information Retrieval Conference (ISMIR 2017)*, 2017, pp. 221–228.

[24] C. McKay, J. Cumming, and I. Fujinaga, "jSymbolic 2.2: Extracting features from symbolic music for use in musicological and MIR research." in *International Society for Music Information Retrieval Conference (ISMIR 2018)*, 2018, pp. 348–354.

[25] C. Inskip and F. Wiering, "In their own words: Using text analysis to identify musicologists' attitudes towards technology," in *International Society for Music Information Retrieval Conference (ISMIR 2015)*, 2015, pp. 455–461.

[26] E. Dahlig-Turek, S. Klotz, R. Parncutt, and F. Wiering, *Musicology (Re-) Mapped: Discussion Paper*. European Science Foundation, 2012.

[27] P. Van Kranenburg, J. Garbers, A. Volk, F. Wiering, L. P. Grijp, and R. C. Veltkamp, "Collaboration perspectives for folk song research and music information retrieval: The indispensable role of computational musicology," *Journal of Interdisciplinary Music Studies*, vol. 4, no. 1, pp. 17–43, 2010.

[28] A. Marsden, "Music analysis by computer: Ontology and epistemology," in *Computational Music Analysis*, 2016, pp. 3–28.

[29] J. E. Dobson, *Critical digital humanities: the search for a methodology*. University of Illinois Press, 2019.

[30] J.-J. Aucouturier and E. Bigand, "Mel Cepstrum & Ann Ova: The difficult dialog between MIR and music cognition." in *International Society for Music Information Retrieval Conference (ISMIR 2012)*, 2012, pp. 397–402.

[31] J. Drucker, "Humanistic theory and digital scholarship," *Debates in the digital humanities*, vol. 150, pp. 85–95, 2012.

[32] 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*, vol. 4, no. 1, pp. 105–113, 2021.

[33] L. Pugin, "The challenge of data in digital musicology," *Frontiers in Digital Humanities*, vol. 2, p. 4, 2015.

[34] S. Münnich, "FAIR for whom? Commentary on Hofmann et al. (2021)," *Empirical Musicology Review*, vol. 16, no. 1, pp. 151–153, 2021.

[35] D. Huron, "The new empiricism: Systematic musicology in a postmodern age," *The 1999 Ernest Bloch Lectures*, pp. 1–32, 1999.

[36] S. Ahlbäck, "Melody beyond notes: A study of melody cognition," Ph.D. dissertation, Göteborgs Universitet, 2004.

[37] F. Moretti, *Distant reading*. Verso Books, 2013.

[38] C. Antila and J. Cumming, "The VIS framework: Analyzing counterpoint in large datasets." in *International Society for Music Information Retrieval Conference (ISMIR 2014)*, 2014, pp. 71–76.

[39] N. Nápoles, G. Vigliensoni, and I. Fujinaga, "Encoding matters," in *Digital Libraries for Musicology (DLfM 2018)*, 2018, pp. 69–73.

[40] A. Morgan, "Renaissance interval-succession theory: Treatises and analysis," Ph.D. dissertation, Schulich School of Music, McGill University, 2017.

[41] S. Howes, "Tonality and transposition in the seventeenth-century trio sonata," Ph.D. dissertation, Schulich School of Music , McGill University, 2021.

[42] R. C. Repetto and X. Serra, "A collection of music scores for corpus based jingju singing research." in *International Society for Music Information Retrieval Conference (ISMIR 2017)*, 2017, pp. 46–52.

[43] R. Gong, R. C. Repetto, and X. Serra, "Creating an a cappella singing audio dataset for automatic Jingju singing evaluation research," in *Digital Libraries for Musicology (DLfM 2017)*, 2017, pp. 37–40.

[44] R. C. Repetto, S. Zhang, and X. Serra, "Quantitative analysis of the relationship between linguistic tones and melody in Jingju using music scores," in *Digital Libraries for Musicology (DLfM 2017)*, 2017, pp. 41–44.

[45] D. Lewis, K. Page, and L. Dreyfus, "Narratives and exploration in a musicology app: Supporting scholarly argument with the Lohengrin TimeMachine," in *Digital Libraries for Musicology (DLfM 2021)*, 2021, pp. 50–58.

[46] N. Schnell, D. Schwarz, J. Larralde, and R. Borghesi, "Pipo, a plugin interface for afferent data stream processing modules," in *International Society for Music Information Retrieval Conference (ISMIR 2017)*, 2017.

[47] E. Ornoy and S. Cohen, "The effect of mindfulness meditation on the vocal proficiencies of music education students," *Psychology of Music*, vol. 50, no. 5, 2021.

[48] J. Huizinga, *Homo ludens*. Routledge, 1949.

[49] R. Caillois, *Man, Play, and Games*. University of Illinois Press, 2001.

[50] L. Manovich, *The language of new media*. MIT press: Cambridge, USA, 2002.

[51] P. Krašovec, *Tujost kapitala*. Sophia, 2021.

[52] V. N. Borsan and L. Stefanija, "Introduction," *Musicological Annual*, vol. 58/2, pp. 10–14, 2022.

[53] F. A. Kittler, *Gramophone, film, typewriter*. Stanford University Press, 1999.

CARNATIC SINGING VOICE SEPARATION USING COLD DIFFUSION ON TRAINING DATA WITH BLEEDING

Genís Plaja-Roglans[⋆] **Marius Miron**[†] **Adithi Shankar**[⋆] **Xavier Serra**[⋆]
[⋆]Music Technology Group, Universitat Pompeu Fabra, Barcelona, Spain
[†]Earth Species Project
genis.plaja@upf.edu

ABSTRACT

Supervised music source separation systems using deep learning are trained by minimizing a loss function between pairs of predicted separations and ground-truth isolated sources. However, open datasets comprising isolated sources are few, small, and restricted to a few music styles. At the same time, multi-track datasets with source bleeding are usually found larger in size, and are easier to compile. In this work, we address the task of singing voice separation when the ground-truth signals have bleeding and only the target vocals and the corresponding mixture are available. We train a *cold diffusion* model on the frequency domain to iteratively transform a mixture into the corresponding vocals with bleeding. Next, we build the final separation masks by clustering spectrogram bins according to their evolution along the transformation steps. We test our approach on a Carnatic music scenario for which solely datasets with bleeding exist, while current research on this repertoire commonly uses source separation models trained solely with Western commercial music. Our evaluation on a Carnatic test set shows that our system improves Spleeter on interference removal and it is competitive in terms of signal distortion. Code is open sourced. [1]

1. INTRODUCTION

Music source separation (MSS) is a core task in the field of music information retrieval (MIR) in which the aim is to automatically separate the different sources in a musical mixture. In this work, we focus on separating the singing voice. In recent years, impressive performance for this difficult and highly undetermined problem has been achieved through the use of deep learning (DL) approaches [1]. Traditionally, MSS models operate on time-frequency representations [1–3], and more recently on waveforms [4, 5], however, the latter are prone to introduce artifacts to the estimated sources. While the combination of both domains has also shown impressive performance [6, 7], these models tend to be large in size and require extensive amounts of computational power, especially for the training stage.

Supervised MSS approaches, which currently lead the field, require fully-isolated multi-track recordings for the target sources. Data of this kind are scarce and constrained to few musical repertoires [8] because recording these at high quality without bleeding is expensive. One solution is to synthesize the signals [8–10], however, these datasets may not be fully realistic and may produce domain mismatch. On the other hand, multi-track datasets with source bleeding, where the track corresponding to a source is contaminated by the leakage from other sources, are easier to build, since these may be compiled through a less complex process, and can be recorded in live performances. We observe large multi-track datasets with bleeding for diverse domains in the literature [11–14], therefore, dedicated MSS systems to be trained with these would be beneficial. In fact, MSS in the presence of bleeding has recently gained interest: a dedicated leaderboard for this problem – albeit in a slightly different context than here – has been included in the Music Demixing Challenge 2023 [15].

In this work, we propose to address the MSS problem for a repertoire that lacks clean isolated tracks: Carnatic Music (CM). The computational analysis of CM has received growing attention in recent years [16]. MSS is a useful pre-processing step in many computational research pipelines on CM. However, researchers use the available models in the literature, typically the pip-installable version of Spleeter [3] – some examples being [17–22] –, which is trained on a large private dataset, presumably including few or no CM examples. Despite not having information on that latter matter, we make the assumption because the 4/5-stem Spleeter models target an instrument arrangement not applicable to CM (vocals, bass, drum, piano, and other), and CM is rarely recorded stem-by-stem in a studio. The domain mismatch between repertories here may hinder the generalization given the unseen instruments and playing/singing techniques. That may also produce a negative effect on the analysis of the separated sources, as well as on further processes such as melody estimation or pattern recognition. Existing works focusing on CM have pointed out the domain mismatch problem for related tasks currently lead by data-driven models [23].

We propose an MSS model to be trained using the Saraga dataset [11] which is, to the best of our knowledge,

[1] https://github.com/MTG/carnatic-separation-ismir23

© G. Plaja-Roglans, M. Miron, A. Shankar, and X. Serra. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** G. Plaja-Roglans, M. Miron, A. Shankar, and X. Serra, "Carnatic singing voice separation using cold diffusion on training data with bleeding", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

the largest open dataset for the computational analysis of Indian Art Music (IAM). Saraga comprises multi-track audio data recorded in live performances and is larger in size ($\approx$ 36h) than the rest of the real-audio MSS datasets in the literature. However, in all multi-track audio signals in Saraga, there is bleeding from the rest of the sources. Our goal is to use the real-world data with bleeding in Saraga to train an MSS model for this domain, while proposing a strategy to output clean isolated signals, even though no bleeding-free signals are available for development.

To achieve so, we train a *cold diffusion* model, followed by unsupervised clustering on the resulting output. Cold diffusion has shown promising results in recovering data samples from a given distribution that have been iteratively perturbed, in T steps, using a deterministic signal [24–26]. We apply said process to iteratively convert the amplitude spectrogram of a mixture to that of a target source with bleeding. This yields a separation as good as the target source with bleeding in the training data. To address this issue, we take advantage of all the intermediary cold diffusion steps to further improve the output. In doing so, we rely on the fact that the energy of the target source, which is predominant, will evolve differently throughout the transformation than the energy of the source bleeding. Note that this process is cumbersome in a single-step non-diffusion separation system, given the overlapping between vocal and accompaniment at various time-frequency bins.

The key contribution of this study is an MSS system that can be developed solely using data with bleeding. With regards to that, solely the mixture and the target source containing bleeding from other instruments are required. Given its relevance to the repertoire, we focus on separating the singing voice. Also, the proposed model is adaptable: the user may choose to be more restrictive with interferences – at the expense of loss of vocal quality – or vice versa. We put special emphasis on being able to characterize the ubiquitous instruments in CM, to reliably remove the interferences from the singing voice. In a computational musicology context, that would improve the musicologically-relevant research done on the separated vocal signal.

2. METHOD

Our separation pipeline assumes the existence of m, the audio signal of the mixture, and the target source with bleeding s^b which is contained in the mixture, while we may not have the remaining sources at hand. In our case, s^b is the singing voice with source bleeding. We present a two-step method to estimate the isolated source $\hat{s}$ by only having m and s^b during training, and solely m during inference.

(1) Cold diffusion process: we aim at running a cold diffusion process to recursively convert the magnitude spectrogram of a mixture M into the magnitude spectrogram of the singing voice with bleeding $\hat{S}^b$.

(2) Unsupervised mask estimation: Note that step **(1)** can only yield estimations as good as the source with bleeding S^b used as ground truth. Toward refining these estimations, we build the final estimation mask by clustering the frequency bins using the entire cold diffusion process to understand how the energy of each bin is evolving during the transformation.

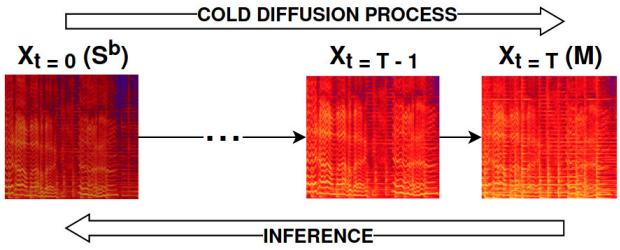


Figure 1. The **spectrogram cold diffusion** transforms, in T steps, a mixture M into a target source with bleeding S^b.

2.1 Feature extraction

2.1.1 Spectrogram cold diffusion

We propose an approach inspired by diffusion models, a class of generative models that define a Markov chain of T steps to iteratively convert samples from a given data distribution into Gaussian noise while learning to conduct the reverse process [27]. The model learns to generate a sample of the given input data distribution from a random sample of noise. Recently, deterministic signals have been successfully used in place of Gaussian noise for the diffusion process [24–26], a technique known as *cold diffusion*.

In [25], the authors apply a transformative cold diffusion process for SVS, using the mixture as the perturbation signal to gradually convert a singing voice to the corresponding mixture, while learning to conduct the reverse process, yielding improved separations for the evaluated model. The process operates in the waveform domain. Here, we propose an updated version of the cold diffusion paradigm in [25] to apply it in time-frequency domain. The cold diffusion process begins at X_0 which is the target data point at inference, in our case S^b, and ends at X_T, in our case M. Let α_t be the perturbation schedule to control the amount of perturbation added at each step and therefore determining the intermediate states of the variable X_t, being t the cold diffusion step. We define α_t as a 1D vector of linearly spaced values from 1 to 0, and of length T. We compute any step in the cold diffusion process as:

$$q_t(X_t|M, X_0) = \alpha_t X_0 + (1 - \sqrt{\alpha_t})M \qquad (1)$$

The process is depicted in Figure 1. In other words, the proposed cold diffusion process gradually converts the amplitude spectrogram of the singing voice with source bleeding into the corresponding mixture, while at inference we aim at reverting said transformation. The use of cold diffusion is motivated by the successful attempts to iteratively transform two-dimensional signals using a diffusion process through a U-Net [24], a well-known network for source separation [2, 3]. Moreover, we can train the model in a supervised fashion, which may lead to more consistent performance than unsupervised procedures. Approaches to

extract features from the spectrogams for clustering [28] are a problem on its own, which in this context may be hindered by source bleeding and the musical and spectral characteristics of CM instruments, e.g. violin or tanpura.

Note that given Eq. 1, the singing voice stays predominant, while the accompaniment gradually increases – in the direction of the cold diffusion process – or decreases – in the direction of the inference process–. This equation is also aimed at amplifying the energy difference throughout the steps between the singing voice and accompaniment frequency bins, and that explains why X_0 and M have different trajectories assigned. The weighting $(1 - \sqrt{\alpha_t})$ applied to the mixture M ensures larger steps at the start of the inference process, while more fine-grained estimations are performed at the latter steps [27], aiming at obtaining more refined separation outputs. Note also that the expected inference input of a singing voice extraction model – in our case corresponding to X_T – is a mixture. Given the expressions in Eq. 1, the perturbation M ensures that $X_T = M$, otherwise the said condition is not given.

2.1.2 Reverse process

The reverse process iteratively removes the deterministic perturbation, aiming at reaching S^b receiving the corresponding mixture M as input. We directly chain the model estimations, so that the model input at a particular step t is the raw prediction of the model at the previous step $t + 1$ (note that the reverse process begins from step T to reach step 1). Therefore, given a trained model D with parameters θ, the reverse process can be defined as follows:

$$r_t(\hat{X}_{t-1}|X_t) = D_\theta(X_t, t) \tag{2}$$

This process is iteratively performed for $t = [T, T - 1..., 1]$, using M as input corresponding to X_T.

2.1.3 Training algorithm

We aim at training a model that learns a mask K_t for each diffusion step t so that $X_t * K_t = \hat{X}_{t-1}$. For each t, we predict a different mask that transforms the signal into the next step in the reverse process until we reach $\hat{X}_0$, which ideally is as close as possible to S^b. Given Eq. 1, we effectively optimize the model using the following objective [27]:

$$L(\theta) = \|X_{t-1} - D_\theta(q_t(X_t|M, X_0), t)\|^2 \tag{3}$$

where X_{t-1} is the known next step in the reverse process computed following $q_t(X_t|M, X_0)$, whereas the model D_θ predicts the next step $\hat{X}_{t-1}$ based on $q_t(X_t|M, X_0)$, the step t, the mixture M, and the input of the cold diffusion process X_0, corresponding to S^b.

We employ a U-Net to learn the reverse process, which has been shown useful for the problem of MSS [2]. We use a U-Net with 7 levels of depth and 4 residual blocks at each level. Both frequency and time dimensions are encoded and then expanded by a factor of 2. The last layer is a sigmoid in order to output the mask K_t of values $\in [0, 1]$, which is multiplied by the input X_t to get $\hat{X}_{t-1}$. We estimate masks instead of spectrograms to obtain a more consistent and linear evolution of the bins energy. Estimating spectrograms may lead to unstable removal of accompaniment, which adds complexity to the proposed approach. To inform the network about the current diffusion step t in the reverse process, we encode it using a 16-dimension sinusoidal positional vector [29]. Said embedding is processed through two dense layers of 64 units. Next, the embedded t is projected to the corresponding channel size at each level of depth of the U-Net and added to the input of each residual block. We inject the time-step embedding to all residual blocks in the encoder, decoder, and bottleneck.

2.1.4 Inference

In standard diffusion models, the output of the last step is considered to be the cleanest signal. We argue that we can achieve better separation by studying how the time-frequency bins evolve throughout the inference process.

We run inference using the trained D_θ to automatically convert an input M to a predicted $\hat{S}^b$ while capturing and stacking all intermediate representations, in order, in a feature matrix $\hat{X}_{T,...,0}$. These features are sized $I \times J \times T$, where I is time size, J number of frequency bins, and T is the number of cold diffusion steps, and represent how the iterative transformation of the magnitude spectrogram of M changes over the cold diffusion steps until reaching predicted $\hat{S}^b$. We normalize the features by dividing all $\hat{X}^{(i,j)}_{T,...,0}$ – being (i, j) the coordinates of a given frequency bin in $\hat{X}_{T,...,0}$ – by $max(\hat{X}^{(i,j)}_{T,...,0})$. Therefore, the energy vectors are studied on the same scale.

2.2 Unsupervised mask estimation

The final mask estimation is performed on top of the cold diffusion feature matrix $\hat{X}_{T,...,0}$, as seen in Figure 2. Existing works use diffusion models to generate features or embeddings for downstream tasks [30], however, to our best knowledge, this is the first attempt to use an entire diffusion process rather than relying only on the output signal.

Note that in the proposed cold diffusion paradigm, we iteratively convert the accompaniment into bleeding – much lower in presence but not removed –, while preserving the cleanest possible voice. Therefore, the energy of the time-frequency bins $\hat{X}^{(i,j)}_{T,...,0}$ across the diffusion steps fluctuates less for the voice than for the accompaniment, which is iteratively lowered by the model. To this end, we propose to cluster the frequency bins based on the evolution of these in $\hat{X}_{T,...,0}$. Clustering techniques have been previously used in a separation context [28, 31, 32], aiming at grouping the components belonging to the same source.

We use K-means clustering to automatically create groups of frequency bins associated with sources, given the computed features $\hat{X}_{T,...,0}$. For example, if a binary separation mask is desired, one may use two clusters and multiply by 0 the clustered bins belonging to the accompaniment, while leaving the rest unchanged. For a soft mask, we consider more than two clusters, and the bins classified in the middle clusters may be shared between the singing voice and other sources, as seen in Figure 2, where we use three clusters. In our case, ideally, the cold diffusion process iteratively reduces the energy of accompaniment bins

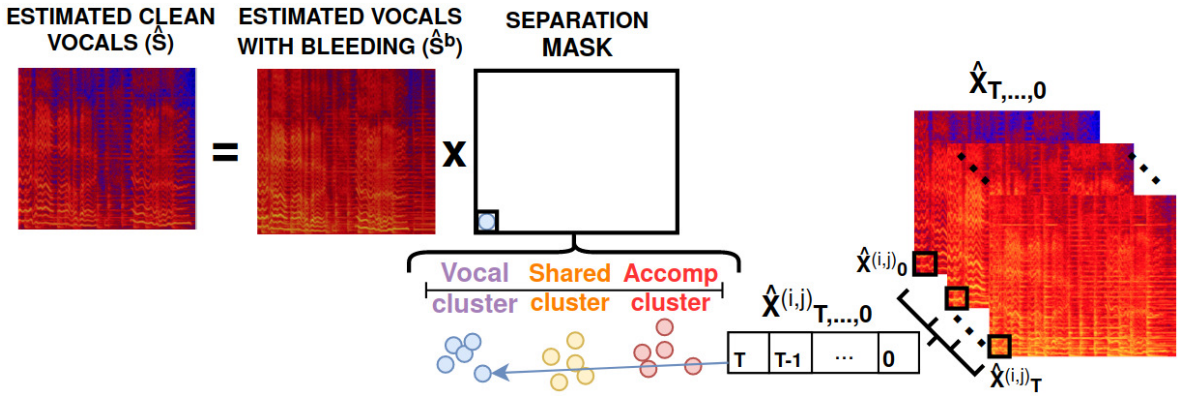


Figure 2. The **unsupervised mask estimation** step clusters the frequency bins given vector $\hat{X}_{T,\ldots,0}$, which stores the evolution from mixture M to the predicted source with bleeding $\hat{S}^b$. The example in this figure uses $C = 3$, being C the number of clusters. The cluster with centroid with lower value is considered the *accompaniment cluster* and assigned 0 in the mask and removed, while the furthest cluster to that is the *vocal cluster* and assigned value 1, so values are left untouched. The bins in the *shared clusters* (we have more than one shared cluster for $C > 3$) are weighted given w^F. Therefore, the user can navigate, given parameter F, through the interference/artifacts trade-off. Using larger C (i.e. considering more clusters) delivers a more granular masking.

while preserving the singing voice. Therefore, the features per bin $\hat{X}^{(i,j)}_{T,\ldots,0}$ have higher values for those corresponding to the singing voice. In this case, the centroid of the closest cluster to the voice centroid has the largest L1 norm. We can then sort the clusters by the L1 norm of the centroid.

Having the clusters ordered, a weight $\in [0, 1]$ must be assigned to each cluster to create the soft mask. Rather than normalizing the cluster centroids, we discover that it is desirable to assign a balanced weighting to the clusters. Thus, we define w, a 1D array linearly spaced values $\in [0, 1]$ of length C, which is the number of clusters. Note that 0 and 1 are both included to directly give a weight of 0 to the *accompaniment cluster* and 1 to the *vocal cluster*, which are the two furthest clusters. Now let F be an integer representing weight factor that is used to control how restrictive we want to be with the intermediate clusters. Given w and F, we compute the final cluster weight array as w^F. For an $F > 1$, we are being more restrictive, especially with the clusters closer to the accompaniment one, and the bigger we set F, the more restrictive we are. When evaluating the clustering, we experiment with various parameter configurations. However, C and F may also be given by the users to control the trade-off between interference and artifacts depending on their needs. Intuitively, the more clusters are considered and removed, we obtain an output with less interference from other sources, at the expense of a loss of quality from the target source.

To take advantage of the first separation run given by the cold diffusion process, we multiply the final mask with the last step of the inference process $\hat{X}_0$, or $\hat{S}^b$. Preliminary results confirmed that this is beneficial over masking the input mixture, and it does not imply added computational expense since $\hat{X}_0$ is contained in the features $\hat{X}_{T,\ldots,0}$.

Note that the use of the cold diffusion process allows the development of differentiable operations for estimating the final separation mask in the context of bleeding. We observed that clustering is not feasible when using a one-step prediction, e.g. two spectrograms do not yield enough information to study the energy change between a vocal and an accompaniment frequency bin.

3. EXPERIMENTS

3.1 Experimental setup

We perform our experimentation using $q_t(X_t|M, X_0)$ with $T = 8$. Generative diffusion typically uses larger T, e.g. 1000 [27]. Using large values for T in this context produces two consecutive steps in the process practically identical, and the optimization of the model becomes extremely complex. We compute the STFT of m and s^b with window size 1024 and hop 256, at a sampling rate of 22050Hz. We use ADAM optimizer with a learning rate of 2^{-4} and batch size of 8, and we run the training process for 1M steps.

The larger in time the input mixture spectrograms are, the more bins to cluster for the final mask estimation. While using an oversized spectrogram may lead to a complex clustering problem given the variations in playing intensity, few points may hinder the estimation of the clusters. Given the improvisatory nature of CM, we propose to use chunks of 3 seconds in order to be robust to the recurrent changes in intensity and dynamics of the performers.

Since we operate on magnitude spectrograms, we require the phase information to reconstruct the estimated audio signals. Here we reuse the phase from m, which is not ideal but it is fast and broadly used in the MSS [2].

3.1.1 *Objective evaluation*

We evaluate the models on a real-audio test set we record for the purpose of this work. It includes $\approx 2h$ of music and two different singers (male and female). Bleeding-free tracks for violin, mridangam, and tanpura are also available. We split the tracks into chunks of 30s, slightly mod-

ifying the mixing parameters to enrich the diversity in the dataset. The tracks are mixed with the assistance of an audio engineer. The testing set is made available for reproducibility and further MSS research.

MSS is commonly evaluated objectively using the `BSS_Eval` metrics [33]: (1) SDR: overall quality, (2) SIR: intrusiveness of the other sources in the estimated source, and (3) SAR: quality of the estimated source. For particular music genres SDR may not correlate with perceptual quality [34–37]. Thus, we run a subjective evaluation in which we contrast the two dimensions captured by the objective evaluation: interference removal (SIR) and signal quality (SAR). This is a common experimental setup in perceptual evaluation of MSS [38]. Therefore, we put more emphasis on these metrics on our objective evaluation as well, rather than comparing solely SDR. Note that our method allows for selecting the desired level of interference at the expense of signal artifacts. Therefore, we aim at covering two scenarios: creative tasks e.g. practicing or mixing, and analysis tasks, e.g. melody estimation.

In a first experiment we compare our system using three different configurations with a baseline U-Net model trained with raw Saraga as regular MSS models are. A second experiment is intended to compare our system with: **(1)** our cold diffusion model skipping the unsupervised clustering mask estimation, and **(2)** Spleeter [3], a widely used model in the literature, also in computational analysis works for CM. We include this comparison considering that Spleeter is trained using a much larger dataset with an unknown distribution. To the best of our knowledge, no Carnatic-specific separation models are available in the literature. In the latter experiment we report the absolute SIR and SAR difference of our models w.r.t. the alternatives aiming at providing an intuitive comparison in terms of interference removal and vocal signal quality. We compute the global MSS metrics [15] for all testing samples using the latest `museval` version [39], and we compute the median to be robust to extreme cases in the testing set.

3.1.2 Perceptual evaluation

Despite the efforts to enhance the variety within our testing set, it is restricted in size and all recordings are obtained from the same source. This is added to the fact that the objective metrics in [40] may not always correlate with the perceptual quality of MSS estimations [34]. For these reasons, we run a perceptual test with subjects including samples from the non-multi-track recordings in Saraga ($\approx$ 17h) – which were not included in the training set for our models, and ensuring there are no overlapping artists – and from the private collection of the Dunya database [41]. We first randomly sample 50 recordings from the said data collections, and we extract the singing voice from a randomly selected 30s chunk for each recording. Using the mixture as reference, we manually collect 6 examples from the batch of separations ensuring that the test includes different audio qualities, gender balance, and tonic diversity.

We design an online survey based on the MUSHRA framework [42]. We request the participants to rate, from 1 to 5, the vocal quality and the intrusiveness of other sources *separately*. The participants are shown the mixture as a reference and two stimuli: our system with $C = 5$ and $F = 4$, and Spleeter. The test includes a tutorial stage with examples – these are not shown during the actual test and are not passed through any of the evaluated models – to make sure the participants have the difference between distortion and intrusiveness from other sources clear. We randomize the order of the stimuli at each example, to prevent the order from having an impact on the ratings. The proposed subjective evaluation follows closely the ITU-T P.835. We include a short survey in the test to collect information on the expertise of the subjects on MSS and CM.

For each testing example, we compute the mean and standard deviation of all rankings. We finally report the mean and standard deviation over the 6 excerpts. The deviation serves as an indicator of the sparsity of the opinions.

	C	***F***	**SDR**	**SIR**	**SAR**
Baseline	-	-	**6.10**	10.71	**8.16**
Ours	3	1	5.88	13.69	6.72
Ours	4	2	5.12	14.94	5.57
Ours	5	3	4.56	**15.84**	4.68

Table 1. Comparison of the baseline with three configurations for our system. C is no. of clusters and F the weight factor. Results are given in dB.

3.2 Results

3.2.1 Objective results

We first compare, on our testing set, our system with $T = 8$ and three different cluster configurations with the baseline U-Net separation model. Results are shown in Table 1. The baseline system is more prone to leak other sources in the estimated vocals given the source bleeding in the training data, while it better preserves the quality of the target source. On the other hand, our system further eliminates the CM instrumentation from the input signal. However, additional masking comes with a drawback and especially in the case of CM where all instruments are pitched and tuned in the same tonic. That produces an important overlap, especially between vocals and violin. Therefore, by removing more interference, we are penalizing the quality of the singing voice.

Related to the latter observation, we confirm the adaptability of our system. The more clusters we consider and remove, we achieve better interference removal at the expense of a loss of vocal quality. However, as seen in Table 1, this is translated into worse SDR values. In the perceptual evaluation we study how these metrics correlate with the perceived quality of the estimations.

In Table 2 we report the difference in SIR and SAR (denoted, respectively, SIRd and SARd), first between two versions of our system (with and without clustering), and second between our system and Spleeter. Using roughly all tested configurations, our system is able to outperform the

	No clustering		Spleeter [3]	
	SIR	**SAR**	**SIR**	**SAR**
	9.39	10.28	14.21	10.95
Comparison of our system with *T=8*				
Config	**vs. No clustering**		**vs. Spleeter [3]**	
C ***F***	**SIRd**	**SARd**	**SIRd**	**SARd**
2 1	+4.70	**-3.43**	+0.14	**-4.09**
3 1	+4.31	-3.56	+0.52	-4.22
3 2	+5.46	-4.49	+0.64	-5.16
4 2	+5.55	-4.71	+0.72	-5.38
4 3	+6.41	-5.53	+1.60	-6.20
5 2	+5.61	-4.69	+0.78	-5.35
5 3	+6.45	-5.59	+1.63	-6.26
5 4	**+7.14**	-6.25	**+2.32**	-6.91

Table 2. SIR and SAR difference of our full system with (1) our system with no unsup. mask estimation and (2) Spleeter. Results given in dB, + indicate that we improve. On top, we provide the absolute metrics of the alternatives for reference. C is no. of clusters and F weight factor.

alternatives in terms of interference removal, better characterizing and cleaning the Carnatic accompaniment from the singing voice, suggesting that we are taking advantage of the in-domain data despite the bleeding. Note also the SIR improvement – more than 4dB in the worst case – that the unsupervised masking provides on top of the last step of the cold diffusion model, which can only estimate, at most, the vocals with bleeding. That is the problem of using data with bleeding for training supervised MSS systems.

However, our system tends to perform worse in signal quality. This may be given by frequency components of the singing voice that are being removed while performing the unsupervised mask estimation, especially those living in the bins shared with other sources. On the other hand, Spleeter maintains a more complete singing voice despite being more prone to interference. We perceptually note that our estimations are *dryer*, while Spleeter is able to better capture components such as reverb and high-frequency details. This may be explained by the much larger training dataset comprising several different vocal styles and effects. In our case, given the proposed schedule and diffusion steps, these components may be partially living on the shared clusters and therefore negatively affected as we use a more restrictive parametrization.

Note the small difference in SAR between our system with no clustering-based masking and Spleeter. Said observation suggests that the cold diffusion process preserves the vocal quality roughly as Spleeter achieves so. That may also explain why masking the last cold diffusion step $\hat{X}_0$ provides an improved output over masking the mixture M.

3.2.2 Perceptual results

We run the MUSHRA test on 25 subjects. From the population, $\approx 44\%$ of the subjects have mid-to-high expertise in MSS, while $\approx 48\%$ have listened CM at least once.

	Mean Opinion Scores (MOS)	
	Vocal quality	**Vocal isolation**
Ours (*C=5, F=4*)	2.80 ± 0.29	**3.72** ± 0.31
Spleeter [3]	**3.73** ± 0.17	1.97 ± 0.19

Table 3. Comparison between our system and Spleeter [3] on a perceptual test. Min=1 / Max=5, the higher the better.

The results of the MUSHRA test (see Table 3) on the intrusiveness of other sources – or how well the vocals are isolated – present a notable correlation with the SIRd in Table 2, suggesting that our model is able to better eliminate the Carnatic instruments from the separated singing voice. Another relevant aspect that we observe is that while Spleeter is still leading on source quality, the scores are more balanced between both models than what the SARd metrics in Table 2 suggest. That may be an indicator that the singing voice components erroneously removed by our model – which notably penalize metrics-wise – are not notably perceivable to the naked ear. All deviations of participant rankings per example are < 1, suggesting that generally there is a disagreement of 1 point at most. Additionally, we run the Wilcoxon signed-rank test for paired data on each example, observing for all cases a p-value < 0.05, indicating that the subject ratings were not given randomly.

4. CONCLUSIONS

We present a system that uses an entire cold diffusion process as features to perform singing voice separation when no isolated ground-truth sources are available, and we solely have the mixture and the target source with bleeding at hand for training. The cold diffusion process, which iteratively transforms a mixture into the target source with bleeding, allows for unsupervised clustering to build the final separation masks. We run our approach on the Saraga dataset, a large Carnatic collection of multi-track audio with bleeding. Despite being trained solely using these data, our model is able to better eliminate the Carnatic instruments from the singing voice than Spleeter, the most commonly used model in computational research for this repertoire, which is trained on a much larger private dataset of clean signals. Albeit the source separation metrics suggest that our system performs worse in terms of vocal distortion, perceptual tests on a dedicated test set suggest that the proposed system trained with noisy and considerably fewer data than Spleeter is competitive with the said system. This will allow to scale up our system since new in-domain data with bleeding are easier to compile than clean data, especially for under-represented music cultures.

As further research, we propose to investigate different schedules, while exploring more sophisticated clustering techniques, aiming at improving source distortion. We also aim at running the proposed pipeline for the other available instrument tracks in Saraga: violin and mridangam.

5. ACKNOWLEDGEMENTS

This work was carried out under the projects Musical AI - PID2019-111403GB-I00/AEI/10.13039/501100011033 funded by the Spanish Ministerio de Ciencia, Innovación y Universidades (MCIU) and the Agencia Estatal de Investigación (AEI). We would also like to acknowledge the 25 subjects that took the perceptual test and Xavier Lizarraga for the assitance on mixing the testing set.

6. REFERENCES

[1] Y. Luo and J. Yu, "Music source separation with band-split RNN," 2022. [Online]. Available: http://arxiv.org/abs/2209.15174

[2] A. Jansson, E. Humphrey, N. Montecchio, R. Bittner, A. Kumar, and T. Weyde, "Singing voice separation with deep U-Net convolutional networks," in *Proc. of the 18th Int. Society for Music Information Retrieval Conf. (ISMIR),* Suzhou, China, 2017, pp. 745–751.

[3] 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*, pp. 1–4, 2020.

[4] D. Stoller, S. Ewert, and S. Dixon, "Wave-U-Net: A multi-scale neural network for end-to-end audio source separation," in *Proc. of the 19th Int. Society for Music Information Retrieval Conf. (ISMIR),* Paris, France, 2018, pp. 334–340.

[5] A. Défossez, N. Usunier, L. Bottou, and F. Bach, "Demucs: Deep Extractor for Music Sources with extra unlabeled data remixed," 2019. [Online]. Available: http://arxiv.org/abs/1909.01174

[6] A. Défossez, "Hybrid spectrogram and waveform source separation," 2021. [Online]. Available: http://arxiv.org/abs/2111.03600

[7] M. Kim, W. Choi, J. Chung, D. Lee, and S. Jung, "KUIELab-MDX-Net: a two-stream neural network for music demixing," 2021. [Online]. Available: http://arxiv.org/abs/2111.12203

[8] M. Miron, J. Janer, and E. Gómez, "Generating data to train convolutional neural networks for classical music source separation," in *Proc. of the 14th Sound and Music Computing Conf.,* Espoo, Finland, 2017, pp. 227–233.

[9] 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. of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2019.

[10] S. Sarkar, E. Benetos, and M. Sandler, "EnsembleSet: a new high quality synthesised dataset for chamber ensemble separation," in *Proc. of the 23rd Int. Conf. on Music Information Retrieval (ISMIR),* Bengaluru, India, 2022.

[11] A. Srinivasamurthy, S. Gulati, R. C. Repetto, and X. Serra, "Saraga: Open Datasets for Research on Indian Art Music," *Empirical Musicology Review*, 2020.

[12] E. Gómez, M. Grachten, A. Hanjalic, J. Janer, S. Jordà, C. F. Julià, C. Liem, A. Martorell, M. Schedl, and G. Widmer, "PHENICX: Performances as Highly Enriched and Interactive Concert Experiences," in *Proc. of the 10th Sound and Music Computing Conf. (SMC),* Stockholm, Sweden, 2013.

[13] O. Mayor, Q. Llimona, M. Marchini, P. Papiotis, and E. M. Gómez, "repoVizz: a framework for remote storage, browsing, annotation, and exchange of multimodal data," in *Proc. of the ACM Int. Conf. on Multimedia (MM'13)*, 2013.

[14] T. Prätzlich, M. Müller, B. W. Bohl, and J. Veit, "Freischütz Digital: Demos of audio-related contributions," in *Demos and Late Breaking News of the Int. Society for Music Information Retrieval Conf. (ISMIR),* Málaga, Spain, 2015.

[15] 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.

[16] G. Tzanetakis, "Computational ethnomusicology: A music information retrieval perspective," in *Proc. of the 40th Int. Computer Music Conf.,* Athens, Greece, 2014.

[17] 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 *Proc. of the 15th Int. Symposium on Computer Music Multidisciplinary Research (CMMR),* Tokyo, Japan.

[18] M. Clayton, P. Rao, N. Shikarpur, S. Roychowdhury, and J. Li, "Raga classification from vocal performances using multimodal analysis," in *Proc. of the 23rd Int. Conf. on Music Information Retrieval (ISMIR),* Bengaluru, India, 2022.

[19] S. John, M. Sinith, S. R.S., and L. P.P., "Classification of indian classical carnatic music based on raga using deep learning," *IEEE Recent Advances in Intelligent Computational Systems (RAICS)*, 2020.

[20] N. Shikarpur, A. Keskar, and P. Rao, "Computational analysis of melodic mode switching in raga performance," in *Proc. of the 22th Int. Society for Music Information Retrieval Conf. (ISMIR),* Online, 2021, pp. 657–664.

[21] R. M.A., V. T.P., and P. Rao, "Structural segmentation of dhrupad vocal bandish audio based on tempo," in *Proc. of the 21th Int. Society for Music Information Retrieval Conf. (ISMIR),* Montréal, Canada, 2020, pp. 678–684.

[22] D. P. Shah, N. M. Jagtap, P. T. Talekar, and K. Gawande, "Raga recognition in indian classical music using deep learning," *Artificial Intelligence in Music, Sound, Art and Design*, pp. 248–263, 2021.

[23] 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 Conf. (TISMIR)*, 2023.

[24] A. Bansal, E. Borgnia, H.-M. Chu, J. S. Li, H. Kazemi, F. Huang, M. Goldblum, J. Geiping, and T. Goldstein, "Cold diffusion: Inverting arbitrary image transforms without noise," 2022. [Online]. Available: http://arxiv.org/abs/2208.09392

[25] G. Plaja-Roglans, M. Miron, and X. Serra, "A diffusion-inspired training strategy for singing voice extraction in the waveform domain," in *Proc. of the 23rd Int. Conf. on Music Information Retrieval (ISMIR),* Bengaluru, India, 2022.

[26] H. Yen, F. G. Germain, G. Wichern, and J. Le Roux, "Cold diffusion for speech enhancement," 2022. [Online]. Available: https://arxiv.org/abs/2211.02527

[27] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Proc. of the 33th Advances in Neural Information Processing Systems (NeurIPS),* Online, 2020, pp. 6840–6851.

[28] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP),* Shangai, China, 2016, pp. 31–35.

[29] 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 Advances in Neural Information Processing Systems (NeurIPS),* Long Beach, USA, 2017, pp. 5999–6009.

[30] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR),* New Orleans, Louisiana, USA, 2021.

[31] Y. Luo, Z. Chen, J. R. Hershey, J. Le Roux, and N. Mesgarani, "Deep clustering and conventional networks for music separation: Stronger together," *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP),* New Orleans, USA, pp. 61–65, 2017.

[32] K. Chen, G. Wichern, F. G. Germain, and J. L. Roux, "Pac-hubert: Self-supervised music source separation via primitive auditory clustering and hidden-unit bert," 2023. [Online]. Available: http://arxiv.org/abs/2304.02160

[33] E. Vincent, R. Gribonval, and C. Févotte, "Performance measurement in blind audio source separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 4, no. 14, pp. 1462–1469, 2006.

[34] 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. IEEE, 2016, pp. 1758–1762.

[35] U. Gupta, E. Moore, and A. Lerch, "On the perceptual relevance of objective source separation measures for singing voice separation," in *Proc. of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2015.

[36] H. Wierstorf, D. Ward, R. Mason, E. M. Grais, C. Hummersone, and M. D. Plumbley, "Perceptual evaluation of source separation for remixing music," *Journal of the Audio Engineering Society (AES)*, 2017.

[37] E. Gusó, J. Pons, S. Pascual, and J. Serrà, "On loss functions and evaluation metrics for music source separation," *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 306–310, 2022.

[38] 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

[39] F.-R. Stöter, A. Liutkus, D. Samuel, L. Miner, and F. Voituret, "sigsep/sigsep-mus-eval: museval 0.4.0," Feb. 2021.

[40] 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.

[41] A. Porter, M. Sordo, and X. Serra, "Dunya: A system for browsing audio music collections exploiting cultural context," in *Proc. of the 14th Int. Society for Music Information Retrieval Conf. (ISMIR),* Curitiba, Brazil, 2013.

[42] 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.

UNVEILING THE IMPACT OF MUSICAL FACTORS IN JUDGING A SONG ON FIRST LISTEN: INSIGHTS FROM A USER SURVEY

Kosetsu Tsukuda **Tomoyasu Nakano** **Masahiro Hamasaki** **Masataka Goto**
National Institute of Advanced Industrial Science and Technology (AIST), Japan
{k.tsukuda, t.nakano, masahiro.hamasaki, m.goto}@aist.go.jp

ABSTRACT

When a user listens to a song for the first time, what musical factors (e.g., melody, tempo, and lyrics) influence the user's decision to like or dislike the song? An answer to this question would enable researchers to more deeply understand how people interact with music. Thus, in this paper, we report the results of an online survey involving 302 participants to investigate the influence of 10 musical factors. We also evaluate how a user's personal characteristics (i.e., personality traits and musical sophistication) relate to the importance of each factor for the user. Moreover, we propose and evaluate three factor-based functions that would enable more effectively browsing songs on a music streaming service. The user survey results provide several reusable insights, including the following: (1) for most participants, the melody and singing voice are important factors in judging whether they like a song on first listen; (2) personal characteristics do influence the important factors (e.g., participants who have high openness and are sensitive to beat deviations emphasize melody); and (3) the proposed functions each have a certain level of demand because they enable users to easily find music that fits their tastes. We have released part of the survey results as publicly available data so that other researchers can reproduce the results and analyze the data from their own viewpoints.

1. INTRODUCTION

When a user listens to a song for the first time on a music streaming service and it matches her taste, she may listen to it until the end or add it to her favorites or a playlist. On the other hand, if the song does not match the user's preferences, she may stop playing it partway through [1,2]. By accumulating logs of such listening behaviors, music streaming services can estimate users' music preferences and implement functions such as recommendations [3,4].

However, when a user first listens to a song and decides whether or not she likes it, which musical factors influence the decision? For example, one user may like a song because of its lyrics, another may like it because of its melody, and third may like it because of the sound of a musical instrument. Several prior studies investigated people's preferred musical factors [5–7]. However, those studies targeted songs that the study participants already liked and investigated the reasons for liking those songs in terms of factors that were specific to the songs. Accordingly, when a participant answered that she liked a certain song because of its lyrics, it was unclear that she would always judge whether she liked or disliked a song because of its lyrics. Thus, despite those studies, there is a lack of research on the musical factors that influence people's judgment on whether they like a song on first listen. This lack of research motivates our first research question:

RQ1 When people listen to a song for the first time and judge whether they like it, which musical factors affect this judgment, and to what extent?

To more deeply understand how people interact with music, the effects of users' personality traits and musical sophistication on their music preferences and listening behaviors have also been studied [5, 8–21]. For example, it has been reported that people with high openness tend to show a preference for folk music [16] and that musical sophistication positively influences recommendation acceptance [20]. Following such studies, we address the second research question:

RQ2 How do people's personality traits and musical sophistication affect the importance of each musical factor in judging whether they like a song?

If a certain musical factor influences judgments about song preferences, it would be useful to propose practical examples of its engineering use. In fact, proposed improvements to the functions of music streaming services from user study results have provided useful insights to the music information retrieval (MIR) community [22–35]. Hence, we investigate a third research question:

RQ3 What are the implications of musical factors for the functions of music streaming services?

To address these research questions, we targeted 10 musical factors and conducted a questionnaire-based online user survey involving 302 participants. Our main contributions can be summarized as follows.

- We reveal that the factors of *melody* and *singing voice* have large influences on music preference judgment, whereas the factor of *danceability* has a small influence.
- From a psychological perspective, we show that both personality traits and musical sophistication affect the importance of the various musical factors. Given these results, we discuss the possibility that the important factors for a particular user could be estimated from the user's listening behaviors on a music streaming service.

© K. Tsukuda, T. Nakano, M. Hamasaki, and M. Goto. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** K. Tsukuda, T. Nakano, M. Hamasaki, and M. Goto, "Unveiling the Impact of Musical Factors in Judging a Song on First Listen: Insights from a User Survey", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

- From an engineering perspective, we propose three functions that would enable users to effectively browse songs by leveraging musical factors, and we show that each function has a certain level of demand.
- We have made the English translation of the survey questionnaire and the survey results publicly available on the web to support future studies [1] .

2. RELATED WORK

2.1 Musical Factors

Understanding why people listen to music has been of interest to researchers. One typical research direction focuses on the motivation to listen to music in daily life. The main reasons include emotional reasons such as relaxation [18, 36–39] and relief [40, 41]. People also listen to music to concentrate and to pass time [42].

Another research direction investigates the reasons for listening to specific preferred songs in terms of musical factors. Greasley et al. [6] conducted interviews about participants' music collections. Among the main reasons why the participants liked their collections were musical factors such as the lyrics and instruments. Sanfilippo et al. [7] asked participants to sample two songs from their music library on a listening device and answer questions such as "why do you enjoy listening to the track?" The participants often answered the questions by using a vocabulary of musical factors. Boyle et al. [5] investigated the influence of musical factors on young people's pop music preferences. Each participant listed his/her three favorite pop songs and rated the importance of various musical factors in liking those songs. The results revealed that melody, mood, and rhythm had large influences. Although these studies investigated the influences of musical factors, they focused on only songs that the participants already liked. Our study is different in that we focus on the musical factors that people emphasize when they listen to a song for the first time. Since there is a vast number of songs that people have not yet listened to, investigating such factors is beneficial to support finding songs that match their preferences.

2.2 Personal Characteristics

In the music domain, user's preferences, interests, and behaviors are influenced by personal characteristics. In particular, many studies have investigated the influences of personality traits measured by the Big Five Inventory [8–14, 16, 17, 43–47]. For example, personality has significant associations with genre preferences [11, 13, 14, 16, 43] and audio preferences [47]. It also influences the desired level of diversity in a recommended song list [46]. Ferwerda et al. [45] revealed that when a user browses for music, the preferred taxonomy (mood, activity, and genre) depends on the user's personality. Such personality-based results can be used for personalization. In fact, several studies have shown increased recommendation quality when personality is incorporated [48–51]. Musical sophistication is another typical personal characteristic that influences music preferences. For example, musically sophisticated users listen to more diverse songs on both the artist and genre levels [52], are more familiar with the songs in a recommended song list [53], and prefer a less personalized playlist [19]. These findings can also be used to improve music recommendations and user interfaces. Following those studies, we investigate the influences of personality traits and musical sophistication on the importance of musical factors, and we suggest how its results can be used to improve the recommendations.

2.3 Design and Function Proposals

For user studies on music listeners' needs, preferences, and behaviors, it is common to not only report the results but also propose designs and functions to improve music services by applying the results [22–35]. Such proposals have provided reusable insights for the MIR community. Examples of these proposals include song recommendations according to the user's attention level [27], support for remote co-listening with a friend [31], and support for users to add their interpretations of lyrics [33]. Inspired by those prior studies, we propose three functions that enable music streaming services to leverage musical factors. Whereas the above studies only proposed designs and functions, we also conducted a user study to evaluate users' willingness to use the proposed functions.

3. PARTICIPANTS

We recruited participants for our user study via an online research company in Japan. We limited the participants to those who were Japanese and listened to music an average of at least one day per week via any music streaming service. The participants answered our questionnaire through a web browser. We paid about 13.21 USD (1,750 JPY) to each participant. Although 354 participants answered the survey, to make the analysis results more reliable, we removed the answers from 52 participants who submitted improper responses to a free-response question. The remaining 302 participants were diverse in both gender and age range: 147 male (10s: 4; 20s: 31; 30s: 33; 40s: 44; 50s: 35) and 155 female (10s: 9; 20s: 39; 30s: 35; 40s: 34; 50s: 38). Hereafter, we report the results obtained from the 302 participants including section 6.

4. INFLUENCE OF MUSICAL FACTORS

4.1 Musical Factors

Referring to prior studies on people's favorite songs [5–7, 54], we targeted the following 10 musical factors that may influence a person's judgment of liking or disliking music on first listen: *melody*, *singing voice*, *rhythm*, *lyrics*, *mood*, *tempo*, *harmony*, *sentiment*, *instruments*, and *danceability*. Although these 10 factors are not completely independent each other (e.g., there would be relatively high correlation between *mood* and *sentiment*), we adopted them to analyze as many factors as possible. In this study, all of these factors were determined entirely from the music. That is, we did not consider social factors that depend on the context of the music or the listener (e.g., the artist's image, the popularity of music, and whether music was introduced by a

[1] They can be downloaded from `https://github.com/ktsukuda/musical_factor`.

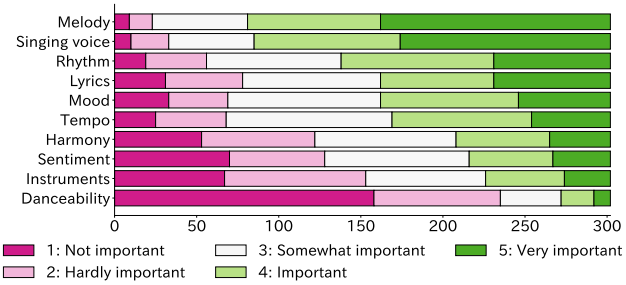


Figure 1. Importance distributions of musical factors (x-axis: number of participants).

friend). Rather, as this is an initial study on the influence of musical factors for judging a song on first listen, we leave the investigation of such social factors for future work.

4.2 Procedure

For each musical factor, we first showed the participants the factor's name, its meaning, and a question. In the case of *instruments*, for example, we showed the following description to represent its meaning: "*Instruments* means the type of instruments used in the piece and their sounds." Similarly, we showed the following question: "How important is the *instruments* in judging whether you like or dislike a song on first listen?" The possible answers were "not important," "hardly important," "somewhat important," "important," and "very important." When the answer for a factor was "not important" or "hardly important," the participant was asked to respond freely on why it was unimportant. On the other hand, when the answer was "somewhat important," "important," or "very important," the participant was asked to respond freely with at least one criterion for judging that he/she liked or disliked a song according to the factor. The 10 musical factors were displayed in a random order to each participant.

Note that in this survey, we asked the participants to answer the questions without actually listening to music to avoid answer bias caused by the music they listened to for the survey. Instead, they were asked to imagine daily situations where they listen to a song for the first time and rate the importance of each factor. This type of survey, which involves imagining a certain situation, is an established survey method in the MIR community [27, 31, 55–59].

4.3 Results

Figure 1 shows the importance distribution for each factor. We can see that the importance was high for *melody* and *singing voice*; in fact, paired Wilcoxon signed-rank tests with Bonferroni correction revealed that their medians (i.e., 4) were statistically higher than the medians of the remaining eight factors at $p < 0.01$. Among the remaining eight factors, more than half of the participants gave a rating of 3, 4, or 5 for *rhythm*, *lyrics*, *mood*, *tempo*, *harmony*, and *sentiment*. To more deeply understand the relationships between factors, we show the Spearman's rank correlations between them in Figure 2. There were high (> 0.4) correlations between *rhythm* and *tempo*, *mood* and *sentiment*, and *melody* and *singing voice*. Although *lyrics* had a relatively high average importance, it had low (< 0.3) correlations with all other factors. *Danceability*, which had

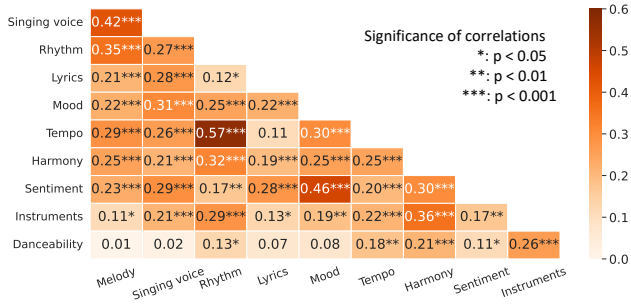


Figure 2. Spearman's rank correlations of importance between musical factors.

the lowest average importance, showed a similar tendency.

For each factor, to analyze the free responses on criteria for liking a song, we manually grouped the responses. Because we allowed the participants to give more than one criterion, each participant's response could be assigned to more than one group. Similarly, we grouped the responses on criteria for disliking a song and reasons for the unimportance of certain factors. Here, we omit the reasons for unimportance, because the most common response for all factors was "I am not interested in this factor." On the other hand, the criteria for liking or disliking a song were diverse, as seen in Table 1, which lists the top three criteria for each factor in terms of the group size. Many criteria involved opposite terms for liked and disliked songs: in the case of *tempo*, for example, participants who gave "fast" as a criterion for liking a song tended to give "slow" as a criterion for disliking a song. In addition, the second column indicates that, for all factors, more participants gave criteria for liking a song than for disliking a song, which means that it was more common to have criteria for liking a song than to have criteria for disliking a song. An interesting application of this finding would be to use criteria for liking a song in explainable recommendation. For example, when a song is recommended to a user who emphasizes *melody*, she may be more willing to listen to it if it appears with an explanation such as "this song is recommended to you because the melody is easy to remember."

The results in Figure 1 are somewhat similar to those reported by Boyle et al. [5] (e.g., *melody* and *rhythm* had high importance, while *danceability* had low importance). Nevertheless, we provide three contributions that are distinct from their results: (1) our results are more generalized, because we did not focus on a specific genre and age group, whereas they focused on young people's pop music preferences; (2) we analyzed the correlations between factors and the criteria for each factor; and (3) we will publish the survey results on the web to support later studies.

5. INFLUENCE OF PERSONAL FACTORS

5.1 Personality Traits

Procedure. We measured the participants' personality traits in terms of five aspects (i.e., *openness*, *conscientiousness*, *extraversion*, *agreeableness*, and *neuroticism*) by using the 29-item Big Five Inventory (BFI) on a 7-point scale (1: strongly disagree - 7: strongly agree) [60]. We used the BFI because of its popularity in past studies [8–14, 16, 17, 43–47] compared to other traits such as

Table 1. Top three criteria for judging "like" and "dislike," for each musical factor. Each number in parentheses indicates the number of participants who responded with the corresponding criterion.

Factor		1st	2nd	3rd
Melody	Like (265)	Easy to remember (35)	Easy to sing or hum (33)	Feels comfortable (28)
	Dislike (193)	Too loud (18)	Difficult to sing or hum (16)	Feels uncomfortable (15)
Singing voice	Like (261)	Specific type (beautiful, powerful, soft, etc.) (74)	Voice to my liking (54)	Feels comfortable (51)
	Dislike (203)	Feels uncomfortable (50)	Specific type (raspy, piercing, etc.) (47)	Voice not to my liking (28)
Rhythm	Like (237)	Groovy (53)	Feels comfortable (23)	Rhythm to my liking (19)
	Dislike (167)	Rhythm not to my liking (17)	Slow (16)	Not groovy (15)
Lyrics	Like (218)	Sympathetic (71)	Inspirational (41)	Positive (10)
	Dislike (164)	Unclear meaning (41)	Lack empathy (30)	Pedestrian (26)
Mood	Like (219)	Cheerful (51)	Fits my mood/situation (25)	Calm (21)
	Dislike (162)	Gloomy (32)	Too loud (29)	Feels uncomfortable (12)
Tempo	Like (220)	Fast (40)	Groovy (29)	Feels comfortable (24)
	Dislike (163)	Slow (48)	Fast (31)	Feels uncomfortable (15)
Harmony	Like (174)	Feels comfortable (43)	Beautiful (23)	Harmonious (22)
	Dislike (116)	Feels uncomfortable (25)	Monotonous (7)	Inharmonious (6)
Sentiment	Like (163)	Positive (33)	Inspirational (30)	Sympathetic (25)
	Dislike (114)	Negative (32)	Evokes no emotion (12)	Doesn't fit my mood/situation (7)
Instruments	Like (146)	Include specific instruments (24)	Fit the song (17)	Feel comfortable (15)
	Dislike (102)	Too loud (24)	Feel uncomfortable (11)	Don't fit the song (7)
Danceability	Like (66)	Body moves naturally to music (13)	Groovy (11)	Rhythmic (9)
	Dislike (46)	Not groovy (6)	Gloomy (5)	Rhythm is bad (4)

Table 2. Spearman's rank correlations between personality traits and musical factor importance (N=302). Significant correlations are shown in bold (*: p< 0.05; **: p< 0.01; ***: p< 0.001).

Trait	Melody	Singing voice	Rhythm	Lyrics	Mood	Tempo	Harmony	Sentiment	Instruments	Danceability
Openness	**0.127***	**0.135***	**0.155****	**0.177****	0.107	0.109	**0.255*****	0.050	**0.157****	**0.151****
Conscientiousness	0.076	**0.128***	0.062	0.031	**0.127***	**0.128***	**0.125***	**0.119***	0.028	0.013
Extraversion	0.062	**0.130***	**0.172****	**0.175****	0.098	**0.114***	**0.254*****	0.107	**0.151****	**0.219*****
Agreeableness	0.025	**0.123***	0.048	0.088	**0.158****	0.029	0.021	0.060	0.049	0.065
Neuroticism	0.003	0.010	-0.081	0.036	-0.025	-0.004	**-0.142***	0.109	-0.072	**-0.120***

opinion leadership [15].

Results. Table 2 lists the Spearman's rank correlations between the personality traits and the importance of each musical factor. *Openness* had significant correlations with as many as seven factors. That is, participants with higher *openness* had more diverse criteria for judging whether a song fits their taste. This result is similar to a previous finding that people with high *openness* tended to listen to more diverse songs in terms of genres [16]. Similarly, *extraversion* also had significant correlations with many factors, particularly, *danceability*. This result echoes a report that people with high *extraversion* tended to listen to songs with high danceability on a music streaming service [51]. *Conscientiousness* was the only trait that had a significant correlation with *sentiment*. Both *agreeableness* and *neuroticism* had significant correlations with as few as two factors. These results are similar to a previous finding that those traits showed significant correlations with few genres [16].

Prior studies correlated personality traits with genre preferences and music audio preferences [16, 47]. For example, people who often listen to folk music were found to have high *openness* [16]. As seen in Table 2, people with high *openness* emphasize *lyrics*; accordingly, for a user who often listens to folk songs, it would be helpful to recommend songs according to the similarity of lyrics.

5.2 Musical Sophistication

Procedure. To measure the musical sophistication, we used the following nine questions on a 7-point scale.

1. InstExp: I engage in regular, daily practice of a musical instrument (1: never - 7: $\geq$ 10 years).
2. DanceExp: I engage in regular, daily dancing (1: never - 7. more than 10 years).
3. NoticeBeat: I can tell when people sing or play out of time with the beat (1: strongly disagree - 7: strongly agree).
4. NoticeTune: I can tell when people sing or play out of tune (1: strongly disagree - 7: strongly agree).
5. LsnMusic: I listen to music (1: $<$ 15 minutes per day - 7: $\geq$ 4 hours per day).
6. LsnNew: I listen to music that is new to me (1: $<$ 1 song per month - 7: $\geq$ 31 songs per month).
7. ViewLyrics: I view lyrics while listening to music (1: $<$ 1 song per month - 7: $\geq$ 31 songs per month).
8. Karaoke: I sing karaoke (1: $<$ 1 time per year - 7: $\geq$ 4 times per week).
9. AttEvt: I attend live music events as an audience member (1: $<$ 1 time per year - 7: $\geq$ 11 times per year).

Questions 1, 3, 4, 5, and 9 derive from the Goldsmiths Musical Sophistication Index (Gold-MSI) [61]. In addition, we asked four questions of our own (questions 2, 6, 7, and 8). For questions 5-9, we asked the participants to give the average frequencies of those behaviors.

Results. Table 3 lists the Spearman's rank correlations between musical sophistication and the importance of each musical factor. Overall, many of the results matched our intuition. For example, DanceExp had a significantly high correlation with *danceability*; participants who were sensitive to beat and tune deviations emphasized audio-based factors such as *melody*, *singing voice*, and *harmony*; and ViewLyrics had the highest correlation with *lyrics*. It is also convincing that participants who often sang karaoke emphasized *lyrics*; those who often attended live music

Table 3. Spearman's rank correlations between musical sophistication and the importance of each musical factor (N=302). Significant correlations are shown in bold (*: p< 0.05; **: p< 0.01; ***: p< 0.001).

Question	Melody	Singing voice	Rhythm	Lyrics	Mood	Tempo	Harmony	Sentiment	Instruments	Danceability
InstExp	0.100	0.061	-0.019	0.108	0.037	-0.099	**0.134***	0.093	0.091	0.101
DanceExp	-0.041	0.039	-0.047	**0.126***	0.030	-0.024	0.044	0.098	-0.005	**0.341*****
NoticeBeat	**0.228*****	**0.228*****	**0.126***	0.082	0.107	0.073	**0.302*****	**0.205*****	**0.147***	0.072
NoticeTune	**0.272*****	**0.231*****	0.099	0.088	**0.121***	0.039	**0.276*****	**0.167****	0.078	0.001
LsnMusic	0.041	0.054	0.111	**0.141***	**0.135***	0.101	0.078	0.108	0.051	0.090
LsnNew	0.003	0.107	**0.152****	**0.152****	0.112	**0.194*****	**0.115***	0.101	**0.126***	**0.169****
ViewLyrics	0.001	0.085	**0.118***	**0.243*****	**0.120***	**0.147***	**0.136***	**0.128***	0.101	0.110
Karaoke	0.085	0.087	0.005	**0.210*****	**0.154****	-0.015	0.057	**0.129***	-0.033	0.081
AttEvent	-0.038	0.037	-0.023	**0.200*****	0.004	0.016	0.039	-0.005	0.088	**0.179****

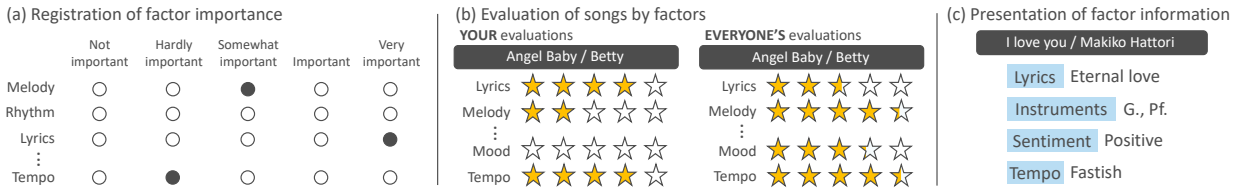


Figure 3. Overview of the three proposed functions. In the user study, these images were presented to the participants.

events emphasized both *lyrics* and *danceability*; and InstExp had a significant correlation with *harmony*. Table 3 also indicates certain high correlations that are not obvious (e.g., between LsnMusic/LsnNew and *lyrics* and between LsnNew and *danceability*).

Certain metrics, such as LsnMusic, LsnNew, and ViewLyrics, can be computed for each user on a music streaming service [59,62,63]. Thus, the results in Table 3 can also be used to increase the confidence in estimating the importance of each factor to a user without explicitly asking the importance. For example, if a user often listens to folk music (i.e., the user would have high *openness* as has been reported by Ferwerda et al. [16]) and new songs, we can estimate from the results in Tables 2 and 3 that *rhythm* is one of the user's important factors. Hence, the user would be more likely to accept recommendations by recommending songs according to the similarity of their rhythms.

6. FUNCTIONS BASED ON MUSICAL FACTORS

In section 4, we showed that certain musical factors influence a person's judgment of liking or disliking a song on first listen. Following those results, in this section, we propose three functions, illustrated in Figure 3, that could enrich and diversify the music listening experience on streaming services. Then, we investigate the usefulness of these functions from the results of a user study.

6.1 Functions

6.1.1 Function 1: Registration of Factor Importance

With this function, shown in Figure 3 (a), users register the importance of each of the 10 musical factors on a 5-point scale when judging whether they like or dislike music on first listen. It is not necessary to register the importance of all factors. For example, the importance of *rhythm* is not registered in Figure 3 (a). The registration process only needs to be done once, and the registered information can be changed later.

This function supports the users as follows. Suppose that a user is listening to her favorite song s. The user has registered *lyrics* as "very important" and *tempo* as "hardly important." Hence, among songs that are new to this user, we can recommend songs that have various tempos and similar lyrics to s. By listening to the recommended songs, the user can find new favorite songs.

6.1.2 Function 2: Evaluation of Songs by Factors

This proposed function allows users to rate their song preferences on a factor-by-factor basis, as shown in Figure 3 (b). The ratings are not mandatory: users only need to rate the songs that they want to rate. In addition, they do not need to rate songs in terms of all 10 factors. For example, in the figure, the user does not rate *mood*. For each song, by computing the average value of all users' rating results for each factor, we can display others' evaluations (averaged ratings) like those shown in Figure 3 (b).

This function supports the users as follows. Suppose that a user is interested in an artist named "Betty," and that *danceability* is an important factor for the user. Then, songs by "Betty" can be sorted and displayed in order of the averaged ratings for *danceability*. This enables efficient discovery of songs that match the user's preferences.

6.1.3 Function 3: Presentation of Factor Information

With this function, information on factors that a user wants to know for a song is displayed as shown in Figure 3 (c). The information on each of the 10 factors can be automatically estimated by using techniques from existing studies [64–70]. Thus, unlike the two previous functions, this one does not require the user to input any information.

This function supports the users as follows. When a user checks a list of newly released songs, usually only basic information such as the artist and title is displayed for each song. In contrast, our proposed function can display information on the musical factor for each song. For example, if the user prefers slow-tempo songs with piano, she can listen only to such songs by referring to the displayed information on *tempo* and *instruments*. This allows the user to efficiently find songs that match her preferences among a vast number of new songs.

Table 4. Top three free-response reasons for "reasonably willing" or "willing" to use each of the proposed functions. Each number in parentheses indicates the number of participants who gave that reason.

	Function 1: registration of factor importance	Function 2: evaluation of songs by factors	Function 3: presentation of factor information
1st	Easy to find music that fits my taste. (46)	Would like to refer to others' evaluations. (22)	Easy to find music that fits my mood/situation. (27)
2nd	Helpful for listening to new songs. (33)	Easy to understand others' evaluations. (14)	Easy to find music that fits my taste. (26)
3rd	Looks interesting to use. (11)	Easy to find music that fits my taste. (13)	Helpful for listening to new songs. (16)

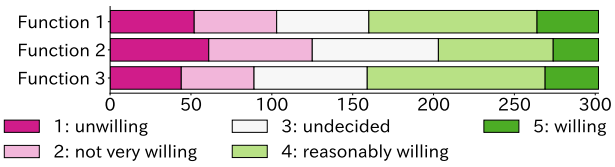


Figure 4. Distribution of the willingness to use each of the proposed functions (x-axis: number of participants).

6.2 Procedure

For each function, we showed the participants an overview of the function and examples of the user support that the function would enable as we described in section 6.1 [2]. The participants were asked to indicate their willingness to use the function, on a 5-point scale ("unwilling," "not very willing," "undecided," "reasonably willing," and "willing"), if it were implemented on the music streaming service that they used regularly. They were also asked to provide free responses on their willingness. The three functions were displayed in a random order to each participant.

6.3 Results

Figure 4 shows the answer distribution for each function. Functions 1 and 3 were more positively received than function 2. To analyze the results, we manually grouped negative responses (i.e., the free responses for "unwilling" and "not very willing"). As we had anticipated, a reason of "I do not need the function" was common for all three functions. Regarding function 2, although we explained that the ratings were not mandatory, a response of "It is tedious to rate songs" was also common. This is why the distribution for function 2 was more biased in the negative direction. Here, note that our goal was not to propose functions that all participants would be willing to use. Rather, we sought to confirm that the proposed functions would have a certain level of demand; accordingly, the results in Figure 4 indicate that we achieved our objective.

We also manually grouped the positive responses (i.e., the free responses for "willing" and "reasonably willing"). Table 4 lists the top three responses in terms of the group size for each function. We can see that, in general, the participants tended to appreciate functions that would make it easy to find music that fits their taste (all functions) and easy to listen to new songs (functions 1 and 3). The responses for function 2 also indicate that they were interested in referring to other users' evaluations of a song. We can also see that the participants felt it was valuable to be able to find music according to their mood or situation (function 3). These responses provide reusable insights for later studies: when researchers or streaming services propose a new function, such user demand could serve as a useful guideline for its design.

If function 3 were implemented on a music streaming service, it might be difficult to estimate the information for all factors because of the platform's resource limitations. In such a case, a possible solution would be to decrease the number of displayed factors according to the results shown in Figure 2. For example, *rhythm* information could be omitted, because *tempo* has a high correlation with *rhythm*, and users who emphasize *rhythm* could thus refer to *tempo* information instead. In contrast, *lyrics* should not be eliminated because it has low correlations with the other factors, and there would not be no alternative factor for users who emphasize *lyrics*.

7. CONCLUSION

In this paper, we conducted an online user survey involving 302 participants. The reusable insights obtained from our user survey can be summarized as follows.

- We showed that the *melody* and *singing voice* are important for most participants. Because there were trends in the criteria for each factor, as seen in Table 1, the criteria could be used to increase the explainability of song recommendations, as discussed in section 4.3.
- Personality and musical sophistication influence the importance of each musical factor. As discussed in sections 5.1 and 5.2, these results would be useful for estimating which factors are important to a user from the user's listening behaviors on a streaming service.
- The evaluation results for our proposed functions show that there is a certain demand for functions that enable users to browse songs according to musical factors. The reasons for each function's demand in Table 4 could provide guidelines for other researchers and services to propose novel factor-based functions.

Finally, we acknowledge a limitation of this paper in that all the participants in our user study were Japanese. Because peoples' music preferences and listening behaviors, as well as music itself, vary widely from country to country [26, 71–76], not all of the findings reported here can be generalized. Nevertheless, we believe that our study provides a worthwhile contribution to the MIR community as a first step toward understanding how musical factors influence whether people like a song on first listen. At the same time, the above limitation can guide future work such as investigating the differences in important musical factors among countries and cultures. The publicly available dataset of results from our user study will enable researchers not only to perform such comparisons but also to analyze and compare results from different viewpoints.

[2] We leave it as future work to actually implement these functions and conduct a long-term user study on them including how to visualize the information.

8. ACKNOWLEDGMENTS

This work was supported in part by JST CREST Grant Number JPMJCR20D4 and JSPS KAKENHI Grant Number JP21H04917, Japan.

9. REFERENCES

[1] H. Yakura, T. Nakano, and M. Goto, "FocusMusicRecommender: A system for recommending music to listen to while working," in *Proceedings of the 23rd International Conference on Intelligent User Interfaces*, ser. IUI 2018, 2018, pp. 7–17.

[2] B. Brost, R. Mehrotra, and T. Jehan, "The music streaming sessions dataset," in *Proceedings of the World Wide Web Conference*, ser. WWW 2019, 2019, pp. 2594–2600.

[3] 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.

[4] Y. Deldjoo, M. Schedl, P. Cremonesi, and G. Pasi, "Recommender systems leveraging multimedia content," *ACM Computing Surveys*, vol. 53, no. 5, pp. 1–38, 2020.

[5] J. D. Boyle, G. L. Hosterman, and D. S. Ramsey, "Factors influencing pop music preferences of young people," *Journal of Research in Music Education*, vol. 29, no. 1, pp. 47–55, 1981.

[6] A. Greasley, A. Lamont, and J. A. Sloboda, "Exploring musical preferences: An in-depth qualitative study of adults' liking for music in their personal collections," *Qualitative Research in Psychology*, vol. 10, no. 4, pp. 402–427, 2013.

[7] K. R. M. Sanfilippo, N. Spiro, M. Molina-Solana, and A. Lamont, "Do the shuffle: Exploring reasons for music listening through shuffled play," *PLOS ONE*, vol. 15, no. 2, pp. 1–21, 2020.

[8] T. Chamorro-Premuzic and A. Furnham, "Personality and music: Can traits explain how people use music in everyday life?" *British Journal of Psychology*, vol. 98, no. 2, pp. 175–185, 2007.

[9] M. J. Delsing, T. F. Ter Bogt, R. C. Engels, and W. H. Meeus, "Adolescents' music preferences and personality characteristics," *European Journal of Personality*, vol. 22, no. 2, pp. 109–130, 2008.

[10] R. L. Zweigenhaft, "A do re mi encore: A closer look at the personality correlates of music preferences," *Journal of individual differences*, vol. 29, no. 1, pp. 45–55, 2008.

[11] R. A. Brown, "Music preferences and personality among japanese university students," *International Journal of Psychology*, vol. 47, no. 4, pp. 259–268, 2012.

[12] T. Chamorro-Premuzic, V. Swami, and B. Cermakova, "Individual differences in music consumption are predicted by uses of music and age rather than emotional intelligence, neuroticism, extraversion or openness," *Psychology of Music*, vol. 40, no. 3, pp. 285–300, 2012.

[13] A. Langmeyer, A. Guglhör-Rudan, and C. Tarnai, "What do music preferences reveal about personality?" *Journal of Individual Differences*, vol. 33, no. 2, pp. 119–130, 2012.

[14] A. Laplante, "Improving music recommender systems: What can we learn from research on music tastes?" in *Proceedings of the 15th International Society for Music Information Retrieval Conference*, ser. ISMIR 2014, 2014, pp. 451–456.

[15] A. E. Krause and A. C. North, "Music listening in everyday life: Devices, selection methods, and digital technology," *Psychology of Music*, vol. 44, no. 1, pp. 129–147, 2016.

[16] B. Ferwerda, M. Tkalcic, and M. Schedl, "Personality traits and music genres: What do people prefer to listen to?" in *Proceedings of the 25th ACM Conference on User Modeling, Adaptation and Personalization*, ser. UMAP 2017, 2017, pp. 285–288.

[17] T. Schäfer and C. Mehlhorn, "Can personality traits predict musical style preferences? A meta-analysis," *Personality and Individual Differences*, vol. 116, pp. 265–273, 2017.

[18] W. M. Randall and N. S. Rickard, "Reasons for personal music listening: A mobile experience sampling study of emotional outcomes," *Psychology of Music*, vol. 45, no. 4, pp. 479–495, 2017.

[19] Y. Liang and M. C. Willemsen, "Personalized recommendations for music genre exploration," in *Proceedings of the 27th ACM Conference on User Modeling, Adaptation and Personalization*, ser. UMAP 2019, 2019, pp. 276–284.

[20] Y. Jin, N. Tintarev, and K. Verbert, "Effects of personal characteristics on music recommender systems with different levels of controllability," in *Proceedings of the 12th ACM Conference on Recommender Systems*, ser. RecSys 2018, 2018, pp. 13–21.

[21] Y. Liang and M. C. Willemsen, "The role of preference consistency, defaults and musical expertise in users' exploration behavior in a genre exploration recommender," in *Proceedings of the 15th ACM Conference on Recommender Systems*, ser. RecSys 2021, 2021, pp. 230–240.

[22] J. S. Downie and S. J. Cunningham, "Toward a theory of music information retrieval queries: System design implications," in *Proceedings of the 3rd International Conference on Music Information Retrieval*, ser. ISMIR 2002, 2002, pp. 299–300.

[23] S. J. Cunningham, N. Reeves, and M. Britland, "An ethnographic study of music information seeking: Implications for the design of a music digital library," in *Proceedings of the 3rd ACM/IEEE-CS Joint Conference on Digital Libraries*, ser. JCDL 2003, 2003, pp. 5–16.

[24] S. Jones, S. J. Cunningham, and M. Jones, "Organizing digital music for use: An examination of personal music collections," in *Proceedings of the 5th International Conference on Music Information Retrieval*, ser. ISMIR 2004, 2004, pp. 397–402.

[25] C. Inskip, R. Butterworth, and A. MacFarlane, "A study of the information needs of the users of a folk music library and the implications for the design of a digital library system," *Information Processing & Management*, vol. 44, no. 2, pp. 647–662, 2008.

[26] X. Hu, J. H. Lee, and L. K. Y. Wong, "Music information behaviors and system preferences of university students in Hong Kong," in *Proceedings of the 15th International Society for Music Information Retrieval Conference*, ser. ISMIR 2014, 2014, pp. 579–584.

[27] J. H. Lee and R. Price, "Understanding users of commercial music services through personas: Design implications," in *Proceedings of the 16th International Society for Music Information Retrieval Conference*, ser. ISMIR 2015, 2015, pp. 476–482.

[28] J. H. Lee, H. Cho, and Y.-S. Kim, "Users' music information needs and behaviors: Design implications for music information retrieval systems," *Journal of the Association for Information Science and Technology*, vol. 67, no. 6, pp. 1301–1330, 2016.

[29] J. H. Lee, Y. Kim, and C. Hubbles, "A look at the cloud from both sides now: An analysis of cloud music service usage," in *Proceedings of the 17th International Society for Music Information Retrieval Conference*, ser. ISMIR 2016, 2016, pp. 299–305.

[30] L. Spinelli, J. Lau, L. Pritchard, and J. H. Lee, "Influences on the social practices surrounding commercial music services: A model for rich interactions," in *Proceedings of the 19th International Society for Music Information Retrieval Conference*, ser. ISMIR 2018, 2018, pp. 671–677.

[31] J. H. Lee, L. Pritchard, and C. Hubbles, "Can we listen to it together?: Factors influencing reception of music recommendations and post-recommendation behavior," in *Proceedings of the 20th International Society for Music Information Retrieval Conference*, ser. ISMIR 2019, 2019, pp. 663–669.

[32] J. H. Lee and A. T. Nguyen, "How music fans shape commercial music services: A case study of BTS and ARMY," in *Proceedings of the 21st International Society for Music Information Retrieval Conference*, ser. ISMIR 2020, 2020, pp. 837–845.

[33] J. H. Lee, A. Bhattacharya, R. Antony, N. Santero, and A. Le, ""Finding home": Understanding how music supports listeners' mental health through a case study of BTS," in *Proceedings of the 22nd International Society for Music Information Retrieval Conference*, ser. ISMIR 2021, 2021, pp. 358–365.

[34] X. Hu, J. Chen, and Y. Wang, "University students' use of music for learning and well-being: A qualitative study and design implications," *Information Processing & Management*, vol. 58, no. 1, pp. 1–14, 2021.

[35] S. Y. Park and B. Kaneshiro, "Social music curation that works: Insights from successful collaborative playlists," *Proceedings of the ACM on Human-Computer Interaction*, vol. 5, no. CSCW1, pp. 1–27, 2021.

[36] J. A. Sloboda, S. A. O'Neill, and A. Ivaldi, "Functions of music in everyday life: An exploratory study using the experience sampling method," *Musicae Scientiae*, vol. 5, no. 1, pp. 9–32, 2001.

[37] A. Lamont and R. Webb, "Short- and long-term musical preferences: What makes a favourite piece of music?" *Psychology of Music*, vol. 38, no. 2, pp. 222–241, 2010.

[38] A. B. Haake, "Individual music listening in workplace settings: An exploratory survey of offices in the UK," *Musicae Scientiae*, vol. 15, no. 1, pp. 107–129, 2011.

[39] T. Schäfer, "The goals and effects of music listening and their relationship to the strength of music preference," *PloS ONE*, vol. 11, no. 3, pp. 1–15, 2016.

[40] A. J. Lonsdale and A. C. North, "Why do we listen to music? A uses and gratifications analysis," *British Journal of Psychology*, vol. 102, no. 1, pp. 108–134, 2011.

[41] S. Y. Park, E. Redmond, J. Berger, and B. Kaneshiro, "Hitting pause: How user perceptions of collaborative playlists evolved in the united states during the covid-19 pandemic," in *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, ser. CHI 2022, 2022, pp. 1–16.

[42] A. C. North, D. J. Hargreaves, and J. J. Hargreaves, "Uses of music in everyday life," *Music Perception: An Interdisciplinary Journal*, vol. 22, no. 1, pp. 41–77, 2004.

[43] P. J. Rentfrow and S. D. Gosling, "The do re mi's of everyday life: The structure and personality correlates of music preferences." *Journal of Personality and Social Psychology*, vol. 84, no. 6, pp. 1236–1256, 2003.

[44] M. Tkalčič, B. Ferwerda, D. Hauger, and M. Schedl, "Personality correlates for digital concert program notes," in *Proceedings of the 23rd ACM Conference on User Modeling, Adaptation and Personalization*, ser. UMAP 2015, 2015, pp. 364–369.

[45] B. Ferwerda, E. Yang, M. Schedl, and M. Tkalcic, "Personality traits predict music taxonomy preferences," in *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*, ser. CHI EA 2015, 2015, pp. 2241–2246.

[46] B. Ferwerda, M. Graus, A. Vall, M. Tkalcic, and M. Schedl, "The influence of users' personality traits on satisfaction and attractiveness of diversified recommendation lists," in *Proceedings of the 4th Workshop on Emotions and Personality in Personalized Systems*, ser. EMPIRE 2016, 2016, pp. 43–47.

[47] A. B. Melchiorre and M. Schedl, "Personality correlates of music audio preferences for modelling music listeners," in *Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization*, ser. UMAP 2020, 2020, pp. 313–317.

[48] R. Hu and P. Pu, "Enhancing collaborative filtering systems with personality information," in *Proceedings of the 5th ACM Conference on Recommender Systems*, ser. RecSys 2011, 2011, pp. 197–204.

[49] I. Fernández-Tobías, M. Braunhofer, M. Elahi, F. Ricci, and I. Cantador, "Alleviating the new user problem in collaborative filtering by exploiting personality information," *User Modeling and User-Adapted Interaction*, vol. 26, no. 2, pp. 221–255, 2016.

[50] F. Lu and N. Tintarev, "A diversity adjusting strategy with personality for music recommendation." in *Proceedings of the 5th Joint Workshop on Interfaces and Human Decision Making for Recommender Systems*, ser. IntRS 2018, 2018, pp. 7–14.

[51] A. B. Melchiorre, E. Zangerle, and M. Schedl, "Personality bias of music recommendation algorithms," in *Proceedings of the 14th ACM Conference on Recommender Systems*, ser. RecSys 2020, 2020, pp. 533–538.

[52] B. Ferwerda and M. Tkalčič, "Exploring online music listening behaviors of musically sophisticated users," in *Proceedings of the 27th Conference on User Modeling, Adaptation and Personalization*, ser. UMAP 2019, 2019, pp. 33–37.

[53] B. Ferwerda, M. P. Graus, A. Vall, M. Tkalcic, and M. Schedl, "How item discovery enabled by diversity leads to increased recommendation list attractiveness," in *Proceedings of the 32nd ACM SIGAPP Symposium on Applied Computing*, ser. SAC 2017, 2017, pp. 1693–1696.

[54] A. LeBlanc, "Outline of a proposed model of sources of variation in musical taste," *Bulletin of the Council for Research in Music Education*, no. 61, pp. 29–34, 1980.

[55] J. H. Lee and J. S. Downie, "Survey of music information needs, uses, and seeking behaviours: Preliminary findings," in *Proceedings of the 5th International Conference on Music Information Retrieval*, ser. ISMIR 2004, 2004, pp. 441–446.

[56] A. Laplante, "Users' relevance criteria in music retrieval in everyday life: An exploratory study," in *Proceedings of the 11th International Society for Music Information Retrieval Conference*, ser. ISMIR 2010, 2010, pp. 601–606.

[57] J. H. Lee and N. M. Waterman, "Understanding user requirements for music information services," in *Proceedings of the 13th International Society for Music Information Retrieval Conference*, ser. ISMIR 2012, 2012, pp. 253–258.

[58] M. Kamalzadeh, D. Baur, and T. Möller, "A survey on music listening and management behaviours," in *Proceedings of the 13th International Society for Music Information Retrieval Conference*, ser. ISMIR 2012, 2012, pp. 373–378.

[59] K. Tsukuda, M. Hamasaki, and M. Goto, "Toward an understanding of lyrics-viewing behavior while listening to music on a smartphone," in *Proceedings of the 22nd International Society for Music Information Retrieval Conference*, ser. ISMIR 2021, 2021, pp. 705–713.

[60] T. Namikawa, I. Tani, T. Wakita, R. Kumagai, A. Nakane, and H. Noguchi, "Development of a short form of the Japanese Big-Five Scale, and a test of its reliability and validity," *The Japanese Journal of Psychology*, vol. 83, no. 2, pp. 91–99, 2012.

[61] 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, pp. 1–23, 2014.

[62] G. Vigliensoni and I. Fujinaga, "The music listening histories dataset," in *Proceedings of the 18th International Society for Music Information Retrieval Conference*, ser. ISMIR 2017, 2017, pp. 96–102.

[63] M. Schedl, S. Brandl, O. Lesota, E. Parada-Cabaleiro, D. Penz, and N. Rekabsaz, "LFM-2b: A dataset of enriched music listening events for recommender systems research and fairness analysis," in *Proceedings of the 2022 Conference on Human Information Interaction and Retrieval*, ser. CHIIR 2022, 2022, pp. 337–341.

[64] K. Tsukuda, K. Ishida, and M. Goto, "Lyric Jumper: A lyrics-based music exploratory web service by modeling lyrics generative process," in *Proceedings of the 18th International Society for Music Information Retrieval Conference*, ser. ISMIR 2017, 2017, pp. 544–551.

[65] 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 19th International Society for Music*

Information Retrieval Conference, ser. ISMIR 2018, 2018, pp. 370–375.

[66] F. Karsdorp, P. van Kranenburg, and E. Manjavacas, "Learning similarity metrics for melody retrieval," in *Proceedings of the 20th International Society for Music Information Retrieval Conference*, ser. ISMIR 2019, 2019, pp. 478–485.

[67] M. Hamasaki, K. Ishida, T. Nakano, and M. Goto, "Songrium RelayPlay: A web-based listening interface for continuously playing user-generated music videos of the same song with different singers," in *Proceedings of the International Computer Music Conference 2020*, ser. ICMC 2020, 2020, pp. 426–429.

[68] A. A. Correya, D. Bogdanov, L. Joglar-Ongay, and X. Serra, "Essentia.js: A javascript library for music and audio analysis on the web," in *Proceedings of the 21st International Society for Music Information Retrieval Conference*, ser. ISMIR 2020, 2020, pp. 605–612.

[69] G. Micchi, K. Kosta, G. Medeot, and P. Chanquion, "A deep learning method for enforcing coherence in automatic chord recognition," in *Proceedings of the 22nd International Society for Music Information Retrieval Conference*, ser. ISMIR 2021, 2021, pp. 443–451.

[70] H. F. Garcia, A. Aguilar, E. Manilow, and B. Pardo, "Leveraging hierarchical structures for few-shot musical instrument recognition," in *Proceedings of the 22nd International Society for Music Information Retrieval Conference*, ser. ISMIR 2021, 2021, pp. 220–228.

[71] X. Hu and J. H. Lee, "A cross-cultural study of music mood perception between American and Chinese listeners," in *Proceedings of the 13th International Society for Music Information Retrieval Conference*, ser. ISMIR 2012, 2012, pp. 535–540.

[72] T. Schäfelr, A. Tipandjan, and P. Sedlmeier, "The functions of music and their relationship to music preference in India and Germany," *International Journal of Psychology*, vol. 47, no. 5, pp. 370–380, 2012.

[73] Y. 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*, ser. ISMIR 2012, 2012, pp. 19–24.

[74] X. Hu, J. H. Lee, K. Choi, and J. S. Downie, "A cross-cultural study on the mood of K-POP songs," in *Proceedings of the 15th International Society for Music Information Retrieval Conference*, ser. ISMIR 2014, 2014, pp. 385–390.

[75] M. Liu, X. Hu, and M. Schedl, "Artist preferences and cultural, socio-economic distances across countries: A big data perspective," in *Proceedings of the 18th International Society for Music Information Retrieval Conference*, ser. ISMIR 2017, 2017, pp. 103–111.

[76] C. Bauer and M. Schedl, "Global and country-specific mainstreaminess measures: Definitions, analysis, and usage for improving personalized music recommendation systems," *PLOS ONE*, vol. 14, no. 6, pp. 1–36, 2019.

TOWARDS BUILDING A PHYLOGENY OF GREGORIAN CHANT MELODIES

Jan Hajič jr.[1] **Gustavo A. Ballen**[2] **Klára Hedvika Mühlová**[3]
Hana Vlhová-Wörner[1]
[1] Masaryk Institute and Archive, Czech Academy of Sciences, Czechia
[2] School of Biological and Behavioural Sciences, Queen Mary University of London, UK
[3] Faculty of Arts, Masaryk University, Czechia
hajic@mua.cas.cz

ABSTRACT

The historical development of medieval plainchant melodies is an intriguing musicological topic that invites computational approaches to study it at scale. Plainchant melodies can be represented as strings from a limited alphabet, hence making it technically possible to apply bioinformatic tools that are used to study the relationships of biological sequences. We show that using phylogenetic trees to study relationships of plainchant sources is not merely possible, but that it can indeed produce meaningful results. We develop a simple plainchant substitution model for Multiple Sequence Alignment, adapt a Bayesian phylogenetic tree building method, and demonstrate the promise of this approach by validating the resultant phylogenetic tree built from a set of Divine Office sources for the Christmas Vespers against musicological knowledge.

1. INTRODUCTION

Gregorian chant is the universal sacred liturgical monody of the Roman Catholic church, which exerts strong control over this musical tradition. There is an authoritative edition of chant: if singers from multiple countries and continents sing together, each from their print of liturgical books, they should encounter no conflicts in performance. However, this was not always so. During the five hundred years of notated Gregorian chant manuscript culture, between Guidonian staff notation and the introduction of the post-Tridentine printed liturgical books, rarely was a chant melody written exactly the same in two sources. [1] Despite its stated role as a unifying element of the Roman Catholic church, Gregorian chant was a diverse tradition.

The diversity of Gregorian chant, both in terms of repertoire and melody, has been a staple of musicological study of plainchant [1, 2]. Already the relative importances of chronology, geography, and *cursus* [2] are, aside from select topics such as the Cistercian reform, not well understood. Recent chant scholarship thrives on the large-scale digitization effort centered around the Cantus Index network of databases [3], and there are ongoing efforts to apply digital methods to the problem of chant transmission such as the DACT project. [3]

In this pilot study, we present a novel pipeline to model the relationships between chant sources using tools from bioinformatics: we adapt multiple sequence alignment and phylogenetic tree inference for chant melodies. We qualitatively evaluate the method on a dataset of sources for Christmas Eve vespers. [4]

2. RELATED WORK

The study of melodic dialects of chant has a long tradition, most prominently in the distinction proposed between "West Frankish" and "East Frankish" chant [4], as has the theory of chant melody in general (i.a. the centonization hypothesis [5, 6] and its criticism [7], [8, pp. 74-75]), but has not yet been performed with computational models at the scales that these enable. The fact that the diversity within chant melodies is a subject worthy of study is further reinforced by the debate on early chant as an orally transmitted tradition [9, 10], justifying an ethnomusicological perspective [11], although the extent of orality of the tradition has since been contested [12]. The formulaic structure of great responsories has been studied in detail [13], even in the pre-computer era [14].

Work on larger-scale computational analysis of chant melodies has recently been done in the area of melody segmentation [15], measurement of the melodic arch hypothesis [16] and of the relationship between antiphons and differentiae across modes [16]. Importantly, these works also provide the Cantus Corpus v0.2 database, which presents the contents of the Cantus Database in a manner ready

[1] See cf. a sample of melodies of an antiphon: https://cantusindex.org/id/004237

© J. Hajič jr., G. A. Ballen, K. H. Mühlová, and H. Vlhová-Wörner. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** J. Hajič jr., G. A. Ballen, K. H. Mühlová, and H. Vlhová-Wörner, "Towards Building a Phylogeny of Gregorian Chant Melodies", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

[2] The ecclesiastical environment of a mansucript, such as a monastery of a specific order, a city church, a cathedral.

[3] https://dact-chant.ca/

[4] Data is available at github.com/Genome-of-Melody/christmas/releases/tag/ISMIR2023, tree inference code at github.com/Genome-of-Melody/mrbayes_volpiano.

for further processing. [5] Cantus Index [6] also provides the Cantus Analysis tool, [7] which is however built solely for analyzing repertoire, not melodies.

In MIR, the potential for applying bioinformatic tools as string processing models has been previously noted in the context of music similarity search. Tune family identification using Multiple Sequence Alignment (MSA) has been tried [17, 18], and MSAs and and BLAST search has been used for melody classification and fast melody retrieval [19], with mixed results.

More closely related to this work in terms of scientific goals, the field of cultural evolution has also been mapping patterns of musical diversity [20], with roots in the Cantometrics project [21]. Most notably, the evolution of folk melodies in English/US and Japanese traditions has been found to exhibit similar properties, using MSAs [22], and phylogeny of electronic music has been mapped using dynamic community detection rather than phylogenetic trees [23], citing limitations of the tree model in light of horizontal cultural transmission. Cultural and biological evolution was correlated in a study comparing populations in terms of genetics and their folk music [24].

Few works in MIR go beyond leveraging MSA as a tool for melodic similarity applications, and in one instance also on chant [25]. From the cultural evolution field have used some phylogenetic models to study music, but so far, not on chant.

3. METHOD

We model the relationships among melodies from a set of chant sources as a phylogenetic tree. The leaves of the chant sources, which carry (artificially ordered) melodies of the selected Cantus IDs in an analogy to how living species carry genes. Each instance of a chant with a certain Cantus ID in each source is a homologous sequence; the collection of melodies from one Cantus ID across sources is here termed a *locus*. The pipeline consists of the following steps:

1. Concatenate cleaned melodies per source (in an arbitrary but fixed order of Cantus IDs)
2. Compute a (partitioned) multiple sequence alignment (MSA) of the concatenated melodies
3. Infer a phylogenetic tree over the MSA

An overview of the pipeline is shown in Fig. 1.

In the Cantus network of databases, chant melodies are transcribed as strings using Volpiano [26]. Volpiano is both a standard for encoding chant melodies in a plain text format, [8] and a font that renders these strings. [9] The encoding uses several non-tone characters, such as hyphens to indicate boundaries between neumes, syllables, and words, or barline characters to indicate sections. For our experiments, we have removed non-note characters (retaining syllable and word separators did not have an appreciable effect on alignment, and would thus unnecessarily complicate the state space).

Any string distance metric can then be used to model between two melodies, and between any two sources (by aggregating the distances between melodies). However, we specifically chose Bayesian phylogenetic trees as the model because (1) their inference procedure can distinguish between similarities that are substantial and those that are the product of chance, (2) the resulting trees, while perhaps not ideal as a model of transmission itself, are optimal results in terms of a clearly defined probabilistic model, and thus have a probabilistic interpretation that directly allows testing hypotheses about the dataset, rather than post-inference normalization of arbitrary similarity scores, (3) the software tools are readily available. [10]

3.1 MSA and Score Matrix

Multiple sequence alignment (MSA) was carried out with MAFFT v7.505 [27]. Mafft is used for the alignment of melodies because it is a state-of-the-art MSA tool that allows aligning arbitrary text using custom score matrices, thus allowing to process data which are not standard biological sequences such as DNA and aminoacids. (It has already been used in MIR, precisely for these advantages [28].) with our custom score matrix described below. We used a maximum of 1000 iterations and global pairing.

By default, Mafft aligns arbitrary text by checking whether a given symbol is equal or not to other entries in the alignment. This is not a good model for melodies, as substitutions are *not* equally likely. The default choice for melodic distance, Mongeau-Sankoff distance [29] addresses the unequal costs of substituting different steps of the scale, but it and others are designed for tonal music, which chant predates by several hundred years. We have not in fact found sufficient music-theoretical understanding of chant melodies (and mode) to design a similar scoring function. Therefore, we resort to a basic physical reality: the cost of a substitution is the number of steps between the two notes, thus crudely mimicking how physically different the position in the melody might feel for a singer familiar with the alternative (see Fig. 2). The application of B flats were assigned a low cost because they were commonly applied without modifying considerably the melody. Liquescences were treated as regular notes of the same pitch. We stress that this is by no means a definitive chant scoring matrix, but rather a starting point to search for one.

3.2 Bayesian Inference of Phylogenetic Trees

A *phylogenetic tree* (hereafter: tree) is a graph representing the evolutionary relationships between the objects of study. These trees have long been used in evolutionary biology as a means to depict evolutionary relationships

[5] https://github.com/bacor/cantuscorpus
[6] https://cantusindex.org/
[7] https://cantusindex.org/analyse
[8] https://cantus.uwaterloo.ca/sites/default/files/documents/2.\%20Volpiano\%20Protocols.pdf
[9] http://www.fawe.de/volpiano/

[10] This article is not meant to inspire the impression that a large amount of technical work was performed: rather, we find it notable that already with a limited amount of technical work, this method already seems to obtain plausible results.

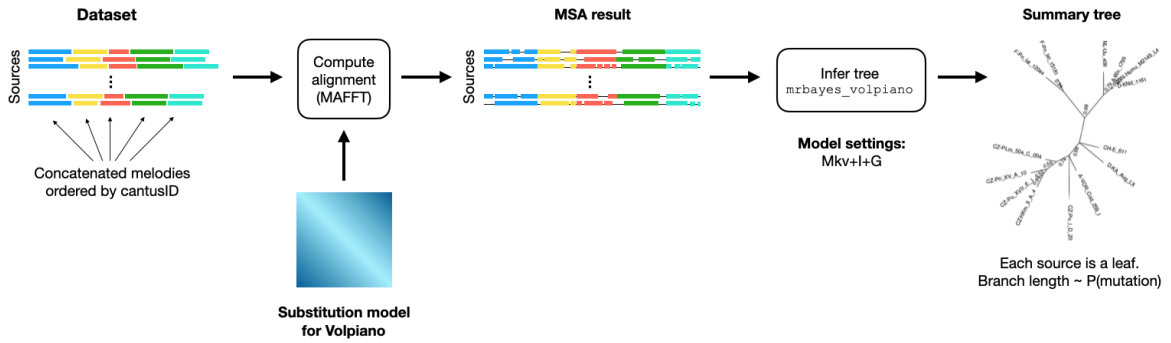


Figure 1. Outline of the pipeline for inference of phylogenetic trees of sources from melodies.

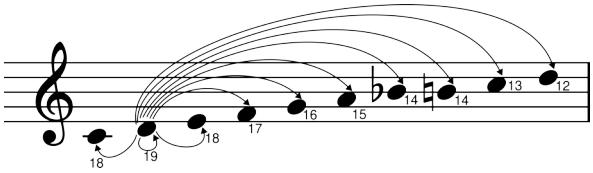


Figure 2. Instances of scores from a reference pitch D. Arrows represent the final pitch and numbers under note bodies represent the score in the matrix. For MAFFT, scores are positive: the furthest distance between pitches is from G3 to D6, which has a score of 1, and the unison has the largest score, which is 19.

among species, and we are here using it to represent the evolution of sources containing melodies in a similar way.

Trees are composed of s leaf nodes of degree 1, called *tips* or *terminals*, and up to $s-1$ internal nodes of degree 3 which represent ancestors. Trees may have a root node: an internal node of degree 2. If a tree is unrooted, there are up to $s-2$ internal nodes. Edges are called branches, and they have lengths that represent some amount of evolutionary change – usually (as in our case) the expected number of changes per site.

Bayesian inference has been applied to phylogenetic tree estimation since the 1990s [30] and consists of calculating the posterior probability of the tree parameters (topology and branch lengths) for a given tree τ [31]:

$$f(\tau_i, Q|\mathbf{X}) = \frac{f(\mathbf{X}|\tau_i, Q)f(\tau_i)f(Q)}{\sum_{j=1}^{B(s)} f(\mathbf{X}|\tau_j, Q)f(\tau_j)f(Q)} \quad (1)$$

In this model, $f(\mathbf{X}|\tau_i)$ is called the phylogenetic likelihood function, which gives the likelihood of observing the alignment data given the model of evolution parameters Q and a particular tree τ_i [32, 33]. Both $f(\tau_i)$ and $f(Q)$ are priors for the topology and model of evolution. The topology prior is usually set to be uninformative (uniform over all possible trees). The prior $f(Q)$ is derived from the Mkv+G model, which is the state of the art for morphology-based phylogeny. [11] As can be anticipated by the fast-growing number of possible trees for a given set of s terminals $B(s) = \frac{(2s-3)!}{2^{n-2}(n-2)}$, this problem cannot be solved by visiting all possible topologies in order to calculate the normalising constant in the Bayes' equation, which also, cannot be analytically solved even for a single topology. Therefore, MCMC sampling is used to construct the posterior densities for all the parameters of the model. The output of MCMC consists of inferred parameter values (mostly branch lengths), sampled trees, and the summary tree.The summary tree summarizes the posterior density of topologies in using a majority-rule consensus: it includes all bi-partitions which are at least in 50% of the sampled topologies. Node posterior probabilities are calculated from the relative frequency of bipartitions in the posterior tree density. Inferred trees are in principle unrooted. [12]

Unfortunately, all existing software for Bayesian phylogenetics restricts the input data to some sort of biological data, be it DNA, aminoacid, or morphological data; therefore, we had to adapt an existing tool to process Volpiano-encoded chants. We chose to modify MrBayes v3.2.7a [34]. We call our fork `mrbayes_volpiano`, in order to make clear that it is intended for use *only* with data in volpiano format. [13] `mrbayes_volpiano` accepts Volpiano-encoded chant melodies as input and analyse them using a Markov model of evolution for an arbitrary number of the discrete character states [35]. It uses all the tools from MrBayes available for standard coding, which is the one applied to melody sequence data and can carry out inference of both single-partition or concatenated settings composed of multiple partitions. It processes alignments in nexus format and can be run both in interactive and scripting mode.

4. CHRISTMAS VESPERS DATASET

In order to test the ability of our pipeline to resolve substantial relationships between chant sources, we apply it on a dataset of Christmas divine office, specifically Vespers for Vigilia Nativitatis Domini. The dataset was originally collected in order to study relationships between

[11] As opposed to phylogenies built from sequences of nucleotides (DNA/RNA) or amino-acids (proteins), morphology-based phylogeny models mostly model transition probabilities from one to a different character as equally likely.

[12] They can be rooted for visualisation e.g. using FigTree (`https://github.com/rambaut/figtree`).

[13] The source code is available at `https://github.com/gaballench/mrbayes_volpiano`

late medieval Bohemian sources with the data including transcribed melodies available in the Hymnologica database, [14] and we combined this data with all further melodies available for Vig. Nat. Domini vespers from the Cantus Index interface, in order to cover a broader European context.

The combined dataset contained 14 sources, and a total of 78 chants falling under 6 distinct Cantus IDs. Because the repertoire in office sources is not entirely consistent across sources and our system aligns melodies directly, we had to select a subset of chants contained in as many sources as possible, and then reduce the set of sources to those that contained as many of these chants as possible. The resulting dataset contained **14 sources**, each of which had fully transcribed melodies for the following Cantus IDs: 001737, 002000, 003511, 004195, 007040a, and 605019. [15]

Some sources contained multiple instances of chants of one Cantus ID: In that case, we retained the version with the most complete version of the melody (as repeated instances of the same chant are sometimes only written as incipits in the sources), and if multiple full melodies were available, we selected the melody that was directly in the Vig. Nat. Dom. section (see Tab. 1).

Why use such a limited dataset, when the entirety of the Cantus Database is available? We originally intended to use the CantusCorpus dataset [15] of Office melodies. However, the authors of the Cantus Database preferred transcribing entire sources, so while there are more than 13000 fully transcribed antiphons in CantusCorpus v0.2 , the vast majority comes from less than 20 sources. This is further compounded by the surprising diversity of office repertoire. Thus, in the entirety of CantusCorpus, it is only possible to find 10 different sources that have transcribed melodies for 5 antiphons. Hence, we decided to use the Christmas dataset, with its advantage of having been collected specifically in order to make the comparison between different sources possible. [16]

4.1 Sources and Evaluation

Our methodology differs from machine learning experiments in *when* data is used. The phylogenetic tree model is selected and parametrized *a priori*, and only then we use a dataset to *validate* the model: is the tree inferred on the dataset plausible according to musicological expectations? Given that chant transmission provides few hard predictions, these expectations are *not* expressed in terms of target values. The evaluation of a tree's plausibility is qualitative.

Since we base the claim of valid results on comparing the inferred tree against known relationships between the sources, we must give an overview of the 14 sources in terms of their placement along the three major dimensions of chant culture: place, time, and liturgical context. This section essentially describes our "evaluation data".

A-Wn 1799.** A 13th century Cistercian antiphoner from the Rein monastery in Austria.

A-VOR Cod. 259/I. A 14th century antiphoner of the collegiate chapter church of Vyšehrad, Prague. In the early 15th century, it was moved to Vorau because of Hussite wars. In 1490-1500, it was adapted for Salzburg liturgy. [17]

CDN-Hsmu M2149.L4. Cistercian antiphoner from the Abbey of Salzinnes, Namur, in the Diocese of Liège, central Belgium, completed in 1554-1555. [18]

CH-E 611. A 14th-century antiphoner from the Benedictine monastery of Einsiedeln, central Switzerland.

CZ-HKm II A 4. An antiphoner from the 1470s, belonging to the municipal Church of the Holy Spirit in Hradec Králové, eastern Czechia. [19]

CZ-PLm 504 C 004. A late antiphonary from the St. Bartholomew municipal church in Pilsen, western Czechia, from 1616. [20]

CZ-Pu XVII E 1. A mixed Latin and Czech antiphonary from the early 16th century, of Czech (but further unspecified) provenance. [21]

CZ-Pn XV A 10. Late 15th century notated breviary from the cathedral cursus in Prague, Czechia. [22]

CZ-Pu I D 20. An antiphonary from the Augustinian monastery in Třeboň, southern Czechia, created in the 2nd half of the 14th century. [23]

D-KA Aug. LX. A complex 12th-century antiphoner, of which the musical notation was almost completely rewritten in the 13th or 14th centuries. From the Zwiefalten Benedictine monastery in southwestern Germany, moved to the abbey of Reichenau in the 15th century. [24]

D-KNd 1161. A late 12th- and early 13th-century Cistercian antiphoner, possibly written for use by the female abbey of Saint Mechtern in Cologne, western Germany, renamed Saint Apern in 1477. [25]

F-Pn lat. 12044. An early 12th-century antiphoner from the Benedictine abbey of St.-Maur-de-Fossés, close to Paris, France. [26]

F-Pn lat. 15181. An early 14th-century notated breviary belonging to the Notre Dame cathedral in Paris, France. [27]

NL-Uu 406. A 12th-century antiphonary from St. Mary's church in Utrecht, Netherlands. Later 13th-15th-century changes. Complex source that has multiple versions of some melodies. [28]

What results should one expect from a phylogeny of these chant sources? The three major dimensions of "ex-

[14] http://hymnologica.cz/jistebnice

[15] All available via https://cantusindex.org/id/(...).

[16] This quest for data also highlights the major limitation of our pipeline so far: we need comparable melodies from each source.

[17] https://manuscripta.at/hs_detail.php?ID=6267

[18] https://cantus.uwaterloo.ca/source/123723

[19] http://hun-chant.eu/source/1481?page=1

[20] https://rukopisy.zcm.cz/view.php?ID=504C004

[21] https://www.manuscriptorium.com/apps/index.php?direct=record&pid=AIPDIG-NKCR__XVII_E_1___32Y2B65-cs#search

[22] http://hymnologica.cz/source/47

[23] http://hymnologica.cz/source/10721

[24] https://cantus.uwaterloo.ca/source/123612

[25] https://cantus.uwaterloo.ca/source/601861

[26] https://cantus.uwaterloo.ca/source/123628

[27] https://cantus.uwaterloo.ca/source/123631

[28] https://cantus.uwaterloo.ca/source/123641

Source	Provenance	Date	Cursus	605019	001737	002000	003511	004195	007040a
A-Wn 1799**	Rein	1200s	Cistercian	1	NA	1	1	1	1
A-VOR Cod. 259/I	Prague	1360	Secular	1	2	1	1	1	1
CDN-Hsmu M2149.L4	Salzinnes	1554	Cistercian	1	NA	1	1	1	1
CH-E 611	Einsiedeln	1300s	Benedictine	1	3	1	1	1	1
CZ-HKm II A 4	Hr. Král.	1400s	Secular	1	1	1	1	1	1
CZ-PLm 504 C 004	Plsen	1616	Secular	1	1	1	1	1	1
CZ-Pu XVII E 1	Bohemia	1516	Unknown	1	NA	1	1	NA	1
CZ-Pn XV A 10	Prague	1300s	Secular	1	1	1	1	1	1
CZ-Pu I D 20	Passau	1300s	Augustinian	1	1	1	1	1	1
D-KA Aug. LX	Zwiefalten	1100s	Benedictine	1	1	1	1	1	1
D-KNd 1161	Köln	1200s	Cistercian	1	NA	1	1	1	1
F-Pn lat. 12044	Paris	1100s	Benedictine	1	1	1	1	2	1
F-Pn lat. 15181	Paris	1300s	Secular	1	NA	1	1	2	1
NL-Uu 406	Utrecht	1150	Secular	1	2	1	3	2	1

Table 1. Sources of the Christmas Vespers dataset with their provenance, approximate date, cursus, and presence of the chant in each source (1 or more instances per source). NA represents chants not present in a given source.

ternal" similarity between chant sources, in terms of how similar the segments of culture represented in these sources are expected to be, are geography, chronology, and cursus – space, time, and the liturgical context within which the books were used. It is not entirely clear in chant scholarship how strongly each of these factors should influence chant melodies (the exception where cursus is clearly expected to dominate other factors is that of the Cistercian order, which mandated that all monasteries must have identical liturgical books [36, p. 99]), but these organizing principles should be observed in the resulting tree.

5. EXPERIMENTS AND RESULTS

For all our experiments, we set up Bayesian inference using an Mkv model of evolution with options +I+G. Metropolis-Hastings MCMC sampling was carried out with four independent runs, each with four chains (one cold and three hot), with 10.000.000 generations, sampling each 1000 generations. Parameter and tree summaries were generated combining the four trace files after a burn-in of 50 % was applied to each. Parameter convergence was assessed by examining the potential scale reduction factor (PSRF) [37] which should approach 1.0 as runs converge, and the average standard deviation of split frequencies (ASDPF) [38] which should be below 0.01 for topological convergence. Other parameters had effective sample size (ESS) values above 600. We do not root the summary trees, because there is no clear outgroup in our dataset.

5.1 Single-locus tree inference

We first computed a tree for sets of melodies under each of the six Cantus IDs separately. In this setting, we examine whether the model can resolve the structure of diversity of individual melodies.

For each cantus ID, we aligned the sets of melodies to obtain a nexus matrix that is then used as input for tree inference. This resulted in six different summary trees, one for each chant. We found varying but overall low degrees of resolution in topology. Some chants had nearly no variation and consequently the majority-rule consensus tree is almost a complete polytomy [29] (003511, 004195). Other chants had several internal nodes resolved, therefore representing some degree of information contained in a single melodic line which shows changes across sources However, at the scale of individual melodies, there was insufficient signal for the model to find meaningful differences.

5.2 Multi-locus tree inference

A concatenated experiment, in which the set of 14 sources was chosen to represent the terminals, was then conducted. We prepared individual alignments for each of the loci so that the boundaries for the same locus (Cantus ID) in the resulting nexus matrix were in fixed positions. Here, a tree was resolved (Fig. 3) that exhibits several properties that we believe make it a plausible model of how chant melodies in these sources are related.

First, cursus. All the Cistercian manuscripts ("white monks") are grouped tighlty together, with the lowest probability of differences – regardless of geographical area and century of origin. This is not entirely the case for the Benedictine manuscripts: the tree does keep together a S. German and a N. Swiss source, but the French Benedictine source is grouped with a French cathedral source. The probability of changes (expressed as branch length) is also much greater between the two closely related Benedictine manuscripts. Finally, there is an interesting case of the Augustinian CZ-Pu I D 20 manuscript and A-VOR Cod. 259/1. The latter is not from an Augustinian monastery, but belonged to a community of canon regulars – a type of clerical community from which the Augustinian order was organized in 1244. They are not particularly close – they do not have an extra internal node like e.g. the French manuscripts – but they are not separated by one, either, and they lie in between the rest of the Czech group and the rest of the tree.

Second, geography. If one briefly disregards the Cistercian branch, the topology of the rest of the manuscripts does roughly correspond to their geographical distribution, from the French group in the west to the Czech group in the east. Note also that while there is some resolution in the group of Czech secular manuscripts, it is barely there: the internal nodes occur at most in six out of ten MCMC samples.

Finally, chronology seems to exert a relatively weak influence, but the dataset is not well suited to study the development of chant melodies in time, as most of the Czech

[29] Star graph: a tree with only one internal node.

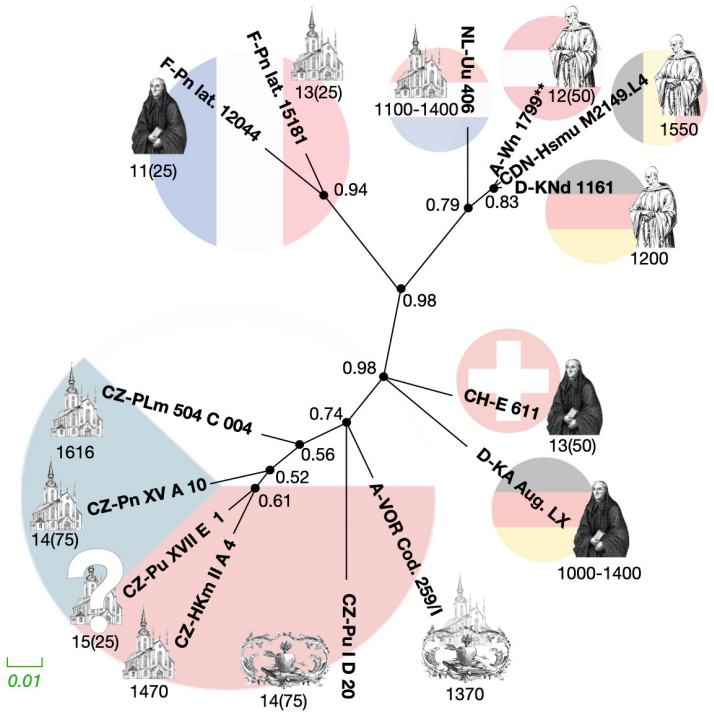


Figure 3. Main experimental result: summary of the posterior tree density as an unrooted majority-rule consensus tree for the concatenated dataset where each chant is a partition. All bipartitions present in at least 50% of the posterior trees are shown as internal nodes, with their nodal posterior probability. Terminals – tree leaves – are sources. Length of edges corresponds to probability of mutation; scale bar (bottom left) for 1 % expected mutation rate. Flags indicate geographical provenance, icons indicate cursus (black monks – Benedictines, white monks – Cistercians, heart – Augustinian, church building – secular cursus). Century (or half-century) indicated directly; some sources (D-KA Aug. LX, NL-Uu 406) have complex histories – see sec. 4.1.

sources are later than most other sources, so it is not clear how to distinguish geographical and chronological factors, and there is only one non-Cistercian clearly pre-1300 old source (F-Pn lat. 12044) that was not modified in the later centuries (which is the case both with D-KA Aug. LX and NL-Uu 406).

6. CONCLUSIONS AND FUTURE WORK

The proposed chant phylogeny pipeline produced a musicologically plausible model of the melodic diversity within the Christmas chant dataset. We do not claim that the resulting tree in Fig. 3 is the *only* or *best* possible way to model the relationships between the sources from our Christmas Vigil dataset; however, while further work should primarily focus on assembling a larger dataset and designing a more robust validation procedure, we believe that based on the current result, the proposed method can meaningfully enrich digital chant scholarship.

A major limitation is that the model requires homologous melodies (indicated by a shared Cantus ID). For the study of melodies to bypass this limitation, "morphological" features derived from melodies would be needed, so that we can process sources that do not share as many (transcribed) melodies. This is especially important for Office sources, where repertoire is strongly differentiated.

Provided one is not interested in the of melodic diversity but only in repertoire structures of chant culture, one can build trees from binary features representing the presence/absence of Cantus IDs at given liturgical positions, using the same Bayesian model but with an alphabet of two rather than 19 characters.

Another limitation is that the current method does not model chronology: it is not yet a model of chant melody *evolution through time*. This complicates interpreting the tree: one potentially attractive idea is that the internal nodes correspond to likely manuscript copying events, but without a more explicit chronology, this remains speculative. Chronology can be incorporated by using Bayesian divergence time estimation (BDTE), an extension of topology inference that produces branch lengths in absolute time rather than the expected number of substitutions per site by using time priors for either nodes or terminals. Furthermore, BDTE could infer a posterior distribution for nodes without observed time values, and thus we could estimate e.g. the times of origin of different layers of a more complex source (such as D-KA Aug. LX or NL-Uu 406) by using time priors rather than precise time values.

Many methodological choices merit further exploration (such as the alignment scoring matrix, choice of tree model, or different ways of combining individual chants). However, based on the already plausible results of this pilot study, we are confident that chant phylogeny is a viable and exciting opportunity for digital chant scholarship.

7. ACKNOWLEDGMENTS

This publication was made possible through the support of a grant from the John Templeton Foundation (project Genome of Melody). The opinions expressed in this publication are those of the author(s) and do not necessarily reflect the views of the John Templeton Foundation. Jan Hajič and Hana Vlhová-Wörner further acknowledge support from the Czech Science Foundation project no. 19-28306X Old Myths, New Facts.

8. REFERENCES

[1] D. Hiley, *Western Plainchant: A Handbook*, ser. Clarendon Paperbacks Series. Clarendon Press, 1995.

[2] W. Apel, *Gregorian chant*. London: Burns & Oates, 1958, vol. 601.

[3] D. Lacoste, "The Cantus Database and Cantus Index Network," in *The Oxford Handbook of Music and Corpus Studies*. Oxford University Press, 2022. [Online]. Available: https://doi.org/10.1093/oxfordhb/9780190945442.013.18

[4] P. Wagner, "Germanisches und romanisches im frühmittelalterlichen kirchengesang," in *Bericht über den I. Musikwissenschaftlichen Kongreß der deutschen Musikgesellschaft in Leipzig 1925*, Leipzig, Germany, 1925, pp. 21–34.

[5] ——, *Einführung in die gregorianischen Melodien: t. Gregorianische Formenlehre; eine choralische Stilkunde*. Breitkopf & Härtel, 1921, vol. 3.

[6] P. Ferretti, *Estetica gregoriana: ossia, Trattato delle forme musicali del canto gregoriano*, ser. Estetica gregoriana: ossia, Trattato delle forme musicali del canto gregoriano. Pontificio istituto di musica sacra, 1934, no. sv. 1. [Online]. Available: https://books.google.cz/books?id=vOWCnQEACAAJ

[7] L. Treitler, "Centonate" chant:" übles flickwerk" or" e pluribus unus?" *Journal of the American Musicological Society*, vol. 28, no. 1, pp. 1–23, 1975.

[8] D. Hiley, *Western plainchant: a handbook*. Oxford, United Kingdom: Clarendon Press, 1993.

[9] H. Hucke, "Toward a new historical view of gregorian chant," *Journal of the American Musicological Society*, vol. 33, no. 3, pp. 437–467, 1980.

[10] L. Treitler, "The early history of music writing in the west," *Journal of the American Musicological Society*, vol. 35, no. 2, pp. 237–279, 1982. [Online]. Available: http://www.jstor.org/stable/831146

[11] P. Jeffery, *Re-Envisioning Past Musical Cultures: Ethnomusicology in the Study of Gregorian Chant*, ser. Chicago Studies in Ethnomusicology. University of Chicago Press, 1992. [Online]. Available: https://books.google.cz/books?id=2pfB84aGUnMC

[12] D. G. Hughes, "Evidence for the traditional view of the transmission of gregorian chant," *Journal of the American Musicological Society*, vol. 40, no. 3, pp. 377–404, 1987.

[13] K. Helsen, "The use of melodic formulas in responsories: constancy and variability in the manuscript tradition," *Plainsong & Medieval Music*, vol. 18, no. 1, pp. 61–76, 2009.

[14] W. H. Frere, *Antiphonale Sarisburiense: a reproduction in facsimile of a manuscript of the 13th century, with a dissertation and analytical index*. Gregg Press Limited, 1901.

[15] B. Cornelissen, W. H. Zuidema, J. A. Burgoyne *et al.*, "Mode classification and natural units in plainchant." in *Proceedings of the 21st Int. Society for Music Information Retrieval Conf.*, Montreal, Canada, 2020, pp. 869–875.

[16] B. Cornelissen, W. Zuidema, and J. A. Burgoyne, "Studying large plainchant corpora using chant21," in *7th International Conference on Digital Libraries for Musicology*, 2020, pp. 40–44.

[17] P. E. Savage and Q. D. Atkinson, "Automatic tune family identification by musical sequence alignment," in *Proceedings of the 16th ISMIR Conference*, vol. 163, 2015.

[18] D. Bountouridis, D. G. Brown, F. Wiering, and R. C. Veltkamp, "Melodic similarity and applications using biologically-inspired techniques," *Applied Sciences*, vol. 7, no. 12, 2017. [Online]. Available: https://www.mdpi.com/2076-3417/7/12/1242

[19] D. Bountouridis, D. Brown, H. V. Koops, F. Wiering, and R. C. Veltkamp, "Melody retrieval and classification using biologically-inspired techniques," in *Computational Intelligence in Music, Sound, Art and Design*, J. Correia, V. Ciesielski, and A. Liapis, Eds. Cham: Springer International Publishing, 2017, pp. 49–64.

[20] P. E. Savage, "Cultural evolution of music," *Palgrave Communications*, vol. 5, no. 1, pp. 1–12, 2019.

[21] A. Lomax, "Factors of musical style," in *Theory & practice: Essays presented to gene weltfish*. Mouton The Hague, 1980, pp. 29–58.

[22] P. E. Savage, S. Passmore, G. Chiba, T. E. Currie, H. Suzuki, and Q. D. Atkinson, "Sequence alignment of folk song melodies reveals cross-cultural regularities of musical evolution," *Current Biology*, vol. 32, no. 6, pp. 1395–1402.e8, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0960982222000926

[23] 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, 2021.

[24] H. Pamjav, Z. Juhász, A. Zalán, E. Németh, and B. Damdin, "A comparative phylogenetic study of genetics and folk music," *Molecular Genetics and Genomics*, vol. 287, no. 4, pp. 337–349, Mar. 2012. [Online]. Available: https://doi.org/10.1007/s00438-012-0683-y

[25] K. Szabová, *Analytical tools for Gregorian chant*. Bachelor thesis. Univerzita Karlova, Matematicko-fyzikální fakulta, 2021.

[26] K. Helsen and D. Lacoste, "A report on the encoding of melodic incipits in the cantus database with the music font 'volpiano'," *Plainsong amp; Medieval Music*, vol. 20, no. 1, p. 51–65, 2011.

[27] K. Katoh and D. M. Standley, "MAFFT Multiple Sequence Alignment Software Version 7: Improvements in Performance and Usability," *Molecular Biology and Evolution*, vol. 30, no. 4, pp. 772–780, 01 2013.

[28] D. Bountouridis *et al.*, "Music information retrieval using biologically inspired techniques," Ph.D. dissertation, Utrecht University, 2018.

[29] M. Mongeau and D. Sankoff, "Comparison of musical sequences," *Computers and the Humanities*, vol. 24, pp. 161–175, 1990.

[30] Z. Yang and B. Rannala, "Bayesian phylogenetic inference using DNA sequences: a Markov Chain Monte Carlo Method." *Molecular Biology and Evolution*, vol. 14, no. 7, pp. 717–724, 07 1997.

[31] J. P. Huelsenbeck and F. Ronquist, "MRBAYES: Bayesian inference of phylogenetic trees ," *Bioinformatics*, vol. 17, no. 8, pp. 754–755, 08 2001.

[32] J. Felsenstein, "Maximum-likelihood estimation of evolutionary trees from continuous characters." *American Journal of Human Genetics*, vol. 25, no. 5, pp. 471–492, Sep. 1973.

[33] ——, "Evolutionary trees from gene frequencies and quantitative characters: Finding maximum likelihood estimates," *Evolution*, vol. 35, no. 6, pp. 1229–1242, 1981.

[34] F. Ronquist and J. P. Huelsenbeck, "MrBayes 3: Bayesian phylogenetic inference under mixed models," *Bioinformatics*, vol. 19, no. 12, pp. 1572–1574, 08 2003.

[35] P. O. Lewis, "A likelihood approach to estimating phylogeny from discrete morphological character data," *Systematic biology*, vol. 50, no. 6, pp. 913–925, 2001.

[36] J. Glasenapp, *To Pray without Ceasing: A Diachronic History of Cistercian Chant in the Beaupré Antiphoner (Baltimore, Walters Art Museum, W. 759–762)*. Columbia University, 2020.

[37] A. Gelman and D. B. Rubin, "Inference from iterative simulation using multiple sequences," *Statistical Science*, vol. 7, no. 4, pp. 457–472, 1992. [Online]. Available: http://www.jstor.org/stable/2246093

[38] F. Ronquist, P. van der Mark, and J. P. Huelsenbeck, *Bayesian phylogenetic analysis using MRBAYES*, 2nd ed. Cambridge University Press, 2009, p. 210–266.

AUDIO EMBEDDINGS AS TEACHERS FOR MUSIC CLASSIFICATION

Yiwei Ding
Music Informatics Group
Georgia Institute of Technology
yding402@gatech.edu

Alexander Lerch
Music Informatics Group
Georgia Institute of Technology
alexander.lerch@gatech.edu

ABSTRACT

Music classification has been one of the most popular tasks in the field of music information retrieval. With the development of deep learning models, the last decade has seen impressive improvements in a wide range of classification tasks. However, the increasing model complexity makes both training and inference computationally expensive. In this paper, we integrate the ideas of transfer learning and feature-based knowledge distillation and systematically investigate using pre-trained audio embeddings as teachers to guide the training of low-complexity student networks. By regularizing the feature space of the student networks with the pre-trained embeddings, the knowledge in the teacher embeddings can be transferred to the students. We use various pre-trained audio embeddings and test the effectiveness of the method on the tasks of musical instrument classification and music auto-tagging. Results show that our method significantly improves the results in comparison to the identical model trained without the teacher's knowledge. This technique can also be combined with classical knowledge distillation approaches to further improve the model's performance.

1. INTRODUCTION

The classification of music has always been a widely popular task in the field of Music Information Retrieval (MIR). Music classification serves as an umbrella term for a variety of tasks, including music genre classification [1], musical instrument classification [2], and music auto-tagging [3]. The last decade has seen dramatic improvements in a wide range of such music classification tasks due to the increasing use of artificial neural networks [4–7].

One major contributing factor to these impressive accomplishments is the increased algorithmic complexity of the machine learning models which also means that the training process requires an increased amount of data. As not all tasks have this abundance of annotated data, transfer learning has been widely and successfully applied to various music classification tasks [8]. In transfer learning, a model is first pre-trained on a large-scale dataset for a (source) task that is somewhat related to the (target) task and then fine-tuned with a comparably smaller dataset of the target task [9]. This enables knowledge to be transferred across datasets and tasks. Transfer learning has been repeatedly shown to result in state-of-the-art performance for a multitude of MIR tasks [10–12].

Another side effect of the increasing model complexity is the slow inference speed. One way to address this issue is model compression by means of knowledge distillation. Here, a low-complexity (student) model is trained while leveraging the knowledge in the high-complexity (teacher) model [13, 14]. The teacher-student paradigm has met with considerable success in reducing the model complexity while minimizing performance decay [15, 16].

In this study, we integrate ideas and approaches from both transfer learning and knowledge distillation and apply them to the training of low-complexity networks to show the effectiveness of knowledge transfer for music classification tasks. More specifically, we utilize pre-trained audio embeddings as teachers to regularize the feature space of low-complexity student networks during the training process. Thus, the main contributions of this paper are a systematic study of

- the effectiveness of various audio embeddings as teachers for knowledge transfer,
- different ways to apply the knowledge transfer from teachers to students, and
- the impact of data availability on the performance of the investigated systems.

The models and experiments are publicly available as open-source code. [1]

2. RELATED WORK

This section first briefly introduces transfer learning and knowledge distillation, which are both often used to transfer knowledge between tasks and models, respectively, and then surveys the application of feature space regularization in the training of neural networks.

2.1 Transfer Learning

In transfer learning approaches, a model is pre-trained on a source task with a large dataset and subsequently fine-tuned on a (different but related) target task with a (typically

© Y. Ding and A. Lerch. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Y. Ding and A. Lerch, "Audio Embeddings as Teachers for Music Classification", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

[1] https://github.com/suncerock/EAsT-music-classification. Last accessed on June 21, 2023.

smaller) dataset [9]. By utilizing the knowledge learned from the source task, models trained following the transfer learning paradigm can often achieve significantly better results than the same models trained directly on the target task [17]; this is especially the case if these models have a large number of parameters and the training data for the target task is limited. In the case where fine-tuning the whole model might be too computationally expensive, another way to do transfer learning is to use the pre-trained embeddings and train only the classification head. This allows for a separation of the tasks of computing the embeddings and the classification itself.

Transfer learning has been successfully applied to a wide variety of areas ranging from computer vision [18, 19] to natural language processing [20]. In MIR, transfer learning has been used for a multitude of target tasks [8, 10, 11, 21]. Besides fine-tuning the whole model, pre-trained embeddings such as VGGish [22] and Jukebox [23] have also shown good performance on many tasks including auto-tagging [12, 24], instrument classification [4, 12], and music emotion recognition [12, 24–26].

One disadvantage of transfer learning is the slow inference speed. In most cases, the model has a large number of parameters, which means that both fine-tuning (if done on the whole model) and inference potentially lead to a high computational workload.

2.2 Knowledge Distillation

Approaches for knowledge distillation aim at model compression, i.e., reducing the complexity of the network. The knowledge of a (usually high-complexity) pre-trained network (the teacher) is transferred to a different (low-complexity) network (the student) during the training phase, in which the student not only learns from the ground truth labels but also from the teacher predictions. This is achieved by adding a "distillation loss" term to the student's loss function to learn from the teacher's prediction [13, 14].

The most popular distillation loss is the Kullback-Leibler divergence between the logits of the student and the teacher, with a hyperparameter called temperature to soften the probability distribution of the teacher's prediction over classes [13]. The soft target provides more "dark" knowledge than the ground truth hard label [27, 28]. The Pearson correlation coefficient has also been proposed as a distance measure between the logits as an alternative to the Kullback-Leibler divergence [29].

Besides learning from logits, the student network can also try to learn from the feature map from the intermediate layers of the teacher network [30–32]. As the feature maps of the student and teacher do not necessarily share the same dimension and the same size, a variety of ways to match the feature space of the student and the teacher have been proposed [31, 33, 34]. Therefore, feature-based knowledge distillation has more flexibility than the logits-based traditional approach, which, at the same time, also makes it more challenging to find the best way of matching the feature space [35, 36].

2.3 Feature Space Regularization

Feature-based knowledge distillation is a technique of regularizing the feature space of the network during training. Besides knowledge distillation, there exists a wide variety of other ways to implement regularization. One example is contrastive learning, which aims at contrasting the features of instances with positive labels against negative labels [37, 38]. Contrastive learning has been shown to improve the performance of neural networks on music auto-tagging [39, 40] and music performance assessment [41].

Regularizing the feature space using pre-trained audio embeddings has also been reported to be effective in music classification [42] and music source separation [43], where Hung and Lerch proposed to use pre-trained embeddings to help structure the latent space during training. This technique is similar to but different from both transfer learning and knowledge distillation. In transfer learning, the same model is used on two different datasets, and a typical setting is that knowledge from the large dataset will be transferred to the small dataset. In knowledge distillation, only one dataset is used and the typical setting is that the knowledge will be transferred from a large model to a small model. In comparison, regularizing the feature space using embeddings requires neither the dataset nor the model to be the same, yet still allows to transfer knowledge learned by the teacher model from a large dataset to the low-complexity student network for a different (small) dataset.

3. METHODS

Inspired by the promising preliminary results of prior work [42], we integrate the idea of transfer learning and knowledge distillation by using pre-trained audio embeddings as teachers to regularize the feature space of the student network during training. The overall pipeline is illustrated in Figure 1.

3.1 Loss Function

Similar to knowledge distillation [13], we rewrite our loss function as

$$\mathcal{L} = (1 - \lambda)\mathcal{L}_{\text{pred}} + \lambda\mathcal{L}_{\text{reg}} \tag{1}$$

where $\mathcal{L}_{\text{pred}}$ is the loss function for conventional neural network training, $\mathcal{L}_{\text{reg}}$ is the loss function that measures the distance between the student network's feature map and the pre-trained embeddings, and $\lambda \in [0, 1]$ is a weighting hyper-parameter.

3.2 Regularization Location

Different stages in a neural network output different feature maps, and the optimal location to apply regularization continues to be controversially discussed in feature-based knowledge distillation [36]. In this study, we investigate either regularizing only the final feature map before the classification head as shown in Figure 1 or regularizing the feature maps at all stages of the student network.

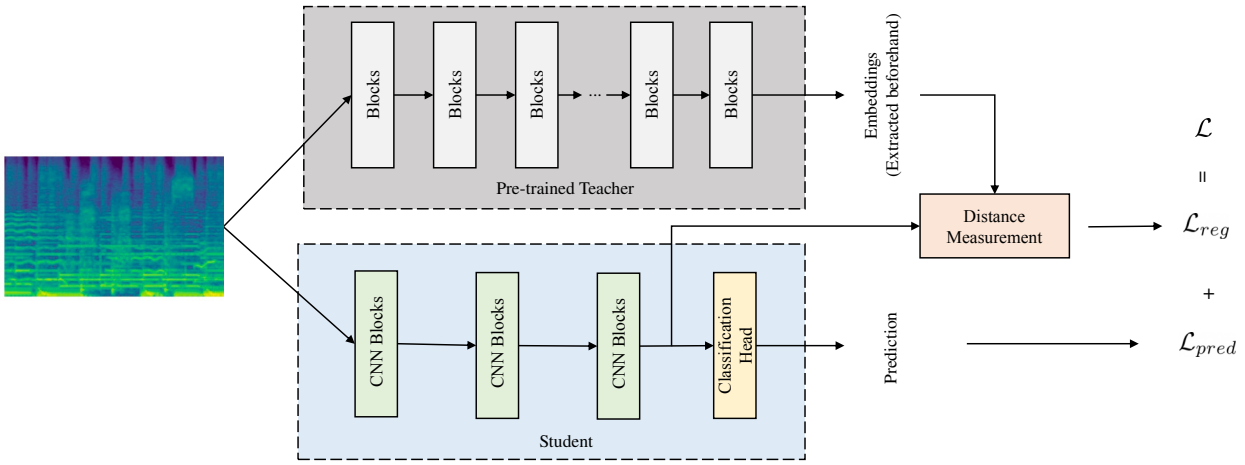


Figure 1: Overall pipeline of training a model by using pre-trained embeddings as teachers. The training loss is a weighted sum (weighting factor omitted in the figure) of prediction loss and regularization loss. The regularization loss measures the distance between pre-trained embedding and the output feature map after the feature alignment. During inference, only the bottom part with the blue background is used.

3.3 Feature Alignment

To measure the distance between the student feature map $l \in \mathbb{R}^{T_\mathrm{s} \times C_\mathrm{s}}$ and the pre-trained teacher embeddings $v \in \mathbb{R}^{T_\mathrm{t} \times C_\mathrm{t}}$ which might have different numbers of time frames (i.e., $T_\mathrm{s} \neq T_\mathrm{t}$), we first align the intermediate feature map with the pre-trained embeddings in time by repeating the one with fewer time frames, then compute the distance for each frame and finally average them along the time axis.

3.4 Distance Measure

Considering that pre-trained embeddings and feature maps have often different dimensionalities, the use of distance measures that are independent of dimensionality allows for easier application.

3.4.1 Cosine Distance Difference

Cosine distance difference [2] as proposed in previous work [42, 43] measures the difference in the cosine distance between pairs of samples. Given n pairs of samples of single-time-frame features $l_1, l_2, ..., l_n$ and pre-trained embeddings $v_1, v_2, ..., v_n$, the cosine distance difference for one pair is

$$D_{ij} = |d_{\cos}(l_i, l_j) - d_{\cos}(v_i, v_j)|, \tag{2}$$

and the distance for this time frame is averaged among all pairs.

3.4.2 Distance Correlation

Distance correlation was proposed as a generalization of classical correlation to measure the independence between two random vectors in arbitrary dimensions [44]. It is capable of handling features of different dimensionality; furthermore, correlation-based distance measures have been shown to be effective in knowledge distillation [29, 32]. Using the same notation as above, we define

$$a_{ij} = \|l_i - l_j\|, \tag{3}$$

$$\bar{a}_{i.} = \frac{1}{n}\sum_{j=1}^{n} a_{ij}, \quad \bar{a}_{.j} = \frac{1}{n}\sum_{i=1}^{n} a_{ij}, \quad \bar{a}_{..} = \frac{1}{n^2}\sum_{i,j=1} a_{ij} \tag{4}$$

$$A_{ij} = a_{ij} - \bar{a}_{i.} - \bar{a}_{.j} + \bar{a}_{..} \tag{5}$$

where $i, j \in \{1, 2, ..., n\}$, and similarly, $b_{ij} = \|v_i - v_j\|$ and $B_{ij} = b_{ij} - \bar{b}_{i.} - \bar{b}_{.j} + \bar{b}_{..}$ [3] The distance for the time frame is then

$$\mathcal{L}_{\mathrm{reg}} = 1 - \mathcal{R}_n^2(l, v) = 1 - \frac{\mathcal{V}_n^2(l, v)}{\sqrt{\mathcal{V}_n^2(l, l)\mathcal{V}_n^2(v, v)}} \tag{6}$$

where

$$\mathcal{V}_n^2(l, l) = \frac{1}{n^2}\sum_{i,j=1}^{n} A_{ij}^2, \quad \mathcal{V}_n^2(v, v) = \frac{1}{n^2}\sum_{i,j=1}^{n} B_{ij}^2,$$
$$\mathcal{V}_n^2(l, v) = \frac{1}{n^2}\sum_{i,j=1}^{n} A_{ij}B_{ij}.$$

Note that $\mathcal{V}_n^2(l, l)$ and $\mathcal{V}_n^2(v, v)$ will be 0 if and only if all the n samples of features (or embeddings) within one batch are identical [44], which we assume not to occur here.

To optimize both distance measures during training, block stochastic gradient iteration is used, which means that the distance is computed over mini-batches instead of the whole dataset [45, 46]. With stochastic approximation, the computational complexity of the distance measure for n samples is reduced from $\mathcal{O}(n^2)$ to $\mathcal{O}(mn)$ where m is the batch size.

It is worth mentioning that both distance measures ensure that if the distance is zero, the feature maps would

[2] has been referred to in previous work as Distance-based Regularization (Dis-Reg) [42, 43].

[3] Eq. (3) uses 2-norm following the implementation in `https://github.com/zhenxingjian/Partial_Distance_Correlation`.

differ from the pre-trained embeddings by only an orthogonal linear transformation, which can be modeled in a single linear layer. Therefore, if the regularization loss is zero, the student would have the same performance as the teacher in classification.

4. EXPERIMENTAL SETUP

We test the effectiveness of using pre-trained embeddings as teachers on two different tasks, datasets, and models with four different pre-trained embeddings as follows.

4.1 Tasks, Datasets, Models, and Metrics

4.1.1 Musical Instrument Classification with OpenMIC

Musical instrument classification is a multi-label classification problem. We use the OpenMIC dataset [2], which provides weakly labeled audio snippets of length 10 s. Following prior work [4, 49], we use the suggested test set and randomly split 15% of the training data as the validation set, resulting in 12,692 training observations, 2,223 validation observations, and 5085 test observations. To ensure a consistent sample rate, the audio is resampled to 32 kHz [5, 49]. As the dataset is not completely labeled, i.e., parts of the labels are missing and not labeled as positive or negative, the missing labels are masked out when computing the loss function as suggested in previous work [5, 10, 49].

We use receptive field regularized ResNet (CP-ResNet) [5] for this task, as it reaches state-of-the-art performance when trained only on the OpenMIC dataset (i.e., neither trained with transfer learning nor trained with any knowledge distillation). CP-ResNet has a ResNet-like structure [19] with an added hyper-parameter ρ to control the maximum receptive field of the ResNet. We set $\rho = 7$ to match the setting which provides the best results in the original work [5].

The results are reported with the metrics mean Average Precision (mAP) and F1-score. The F1-score is calculated in a macro fashion, which means that for each instrument, the F1-score is computed for both the positive labels and the negative labels and then averaged, and the final F1-score is the mean of the F1-scores of all instruments. The detection threshold for the prediction is set to 0.4 following previous work [5].

4.1.2 Music Auto-Tagging with MagnaTagATune

Similar to musical instrument classification, music auto-tagging is also a multi-label classification problem. We use the MagnaTagATune dataset [3] for this task, which comes with audio clips of approximately 29.1 s. Following previous work, we use only the top 50 labels and exclude all the songs without any positive label from the dataset [7, 50]. For comparability, the data split is adopted from previous work, with audio files in the directories '0' to 'b' being the training set, 'c' being the validation set, and 'e' and 'f' being the test set [48, 51], resulting in 15,247 training clips, 1,529 validation clips, and 4,332 test clips.

We apply a modified fully convolutional neural network (FCN) [6] to this task. It is the simplest model among the benchmark models for the MagnaTagATune dataset [48] and consists of several convolution and max-pooling layers. To further reduce the complexity of the model, we apply the MobileNet-like modification [52] to the network by breaking the 3×3 convolutions into depth-wise separable convolutions and 1×1 convolutions.

The results are evaluated with mAP and ROC-AUC.

4.2 Pre-trained Embeddings

4.2.1 VGGish

VGGish [22] is a widely used embedding in MIR, with a VGG network [53] being trained on a large number of Youtube videos. The open-source PyTorch implementation is used to extract VGGish features [4] which by default extracts 128 principle components and then quantizes them to 8 bit. The time resolution is 960 ms.

4.2.2 OpenL3

The OpenL3 embedding [54,55] is trained on a music subset of AudioSet [56] in a self-supervised paradigm. The audio embeddings are extracted using the open-source Python package `OpenL3` [5] with the dimensionality being 512. To keep consistent with VGGish, the time resolution is set to 960 ms.

4.2.3 PaSST

PaSST [10] is a 7-layer transformer trained on AudioSet for acoustic event detection. It applies the structure of a vision transformer [16, 57] and proposes the technique of Patchout to make the training efficient. We use the open-source code [6] released by the authors to extract the 768-dimensional embeddings. The time resolution is also set to 960 ms.

4.2.4 PANNs

PANNs [11] include several convolutional neural networks and are also trained on AudioSet for acoustic event detection. We use the default CNN14 model from the official repository [7]. The embedding dimensionality is 2048. Different from other embeddings, PANNs provide only one global embedding for each clip of audio. Pilot experiments have shown that extracting the embeddings for short segments and concatenating them does not improve performance.

4.3 Systems Overview

The following systems are evaluated for comparison:

- Baseline: CP ResNet (on OpenMIC) and Mobile FCN (on MagnaTagATune) trained without any extra regularization loss.

[4] `https://github.com/harritaylor/torchvggish`. Last accessed on April 4, 2023.

[5] `https://github.com/marl/openl3/tree/main`. Last accessed on April 4, 2023

[6] `https://github.com/kkoutini/PaSST/tree/main`. Last accessed on April 4, 2023.

[7] `https://github.com/qiuqiangkong/audioset_tagging_cnn`. Last accessed on April 4, 2023.

OpenMIC	None		VGGish		OpenL3		PaSST		PANNs	
	mAP	F1	mAP	F1	mAP	F1	mAP	F1	mAP	F1
CP ResNet* [5]	.819	.809	-	-	-	-	-	-	-	-
SS CP ResNet* [5]	.831	.822	-	-	-	-	-	-	-	-
$\text{Teacher}_{\text{LR}}$	-	-	.803	.799	.803	.798	**.858**	**.837**	.853	**.834**
KD (w/ mask) **	-	-	.829	.820	.823	.813	.851	.834	.848	.823
$\text{EAsT}_{\text{Cos-Diff}}$	-	-	.838	.824	**.838**	.820	.837	.822	.836	.814
$\text{EAsT}_{\text{Final}}$	-	-	**.842**	**.828**	.835	**.822**	.847	.830	.849	.828
EAsT_{All}	-	-	.836	.823	.835	**.822**	.845	.827	.845	.827
EAsT_{KD}	-	-	.836	.825	.836	.821	.852	.834	**.857**	.831

MagnaTagATune	None		VGGish		OpenL3		PaSST		PANNs	
	mAP	AUC	mAP	AUC	mAP	AUC	mAP	AUC	mAP	AUC
FCN† [6]	.429	.900	-	-	-	-	-	-	-	-
Mobile FCN	.437	.905	-	-	-	-	-	-	-	-
$\text{Teacher}_{\text{LR}}$	-	-	.433	.903	.403	.890	**.473**	**.917**	**.460**	.911
KD	-	-	.447	.911	.439	.907	.454	.912	.448	.909
$\text{EAsT}_{\text{Cos-Diff}}$	-	-	.446	.906	.438	.907	.453	.912	.453	.911
$\text{EAsT}_{\text{Final}}$	-	-	.454	**.912**	.447	.910	.459	.912	.449	.909
EAsT_{All}	-	-	**.455**	.911	**.452**	**.911**	.458	.913	.457	.911
EAsT_{KD}	-	-	.441	.908	.437	.904	.461	.915	.459	**.912**

Table 1: Results on OpenMIC (above) and MagnaTagATune (below) dataset for different models regularized with different pre-trained embeddings. Best performances are in bold, and best results excluding the teachers are underlined. *Reported results [5], SS means being trained with shake-shake regularization [47]. **When using KD, the missing labels in OpenMIC were masked to avoid potentially adding more training data. †Results from the open-source re-implementation [48].

- $\text{Teacher}_{\text{LR}}$: logistic regression on the pre-trained embeddings (averaged along the time axis), which can be seen as one way to do transfer learning by freezing the whole model except for the classification head.
- KD: classical knowledge distillation where the soft targets are generated by the logistic regression.
- $\text{EAsT}_{\text{Cos-Diff}}$ (for Embeddings-As-Teachers): feature space regularization as proposed by Hung and Lerch that uses cosine distance difference and regularizes only the final feature map [42].
- $\text{EAsT}_{\text{Final}}$ and EAsT_{All}: proposed systems based on distance correlation as the distance measure, either regularizing only at the final stage or at all stages, respectively.
- EAsT_{KD}: a combination of classical knowledge distillation and our method of using embeddings to regularize the feature space. The feature space regularization is done only at the final stage.

We perform a search of λ for each of the EasT systems and choose the best-performing value on the validation set. [8]

5. RESULTS

This section presents the results of different systems and their performance in the case of limited training data.

[8] For all the hyperparameters, please refer to the config files in our GitHub.

5.1 Results on OpenMIC and MagnaTagATune

Table 1 shows the results on the OpenMIC and the MagnaTagATune datasets.

We can observe that the models trained with the extra regularization loss consistently outperform the non-regularized ones on both datasets, with all features, and all regularization methods. This means that the knowledge in the embeddings is successfully transferred to the student networks and consistently enhances the performance.

Although $\text{EAsT}_{\text{Final}}$ appears to give better results on the OpenMIC dataset while EAsT_{All} seems to have slightly better performance on the MagnaTagATune dataset, the difference between them is very small, meaning that the model does not benefit significantly from being regularized by pre-trained embeddings at earlier stages where the feature maps are still relatively low-level.

The results for the teacher systems show that the older VGGish and OpenL3 embeddings are clearly outperformed by the more recently proposed embeddings PaSST and PANNs. In fact, the teacher systems for the newer embeddings perform so strongly that the students can rarely outperform them, while the student systems trained with VGGish and OpenL3 provide better results than the corresponding teachers. We can see that whether the teachers themselves have an excellent performance or not, students benefit from learning the additional knowledge from these embeddings, and the students' upper limit is not bounded by the performance of teachers.

Comparing KD and the $\text{EAsT}_{\text{Final}}$ or EAsT_{All} systems,

Model	Parameters (M)	Iteration / s
VGGish	72.1	172.2
OpenL3	4.69	117.9
PaSST	86.1	18.7
PANNs	79.7	70.6
Mobile FCN	0.34	319.3
CP ResNet	5.52	205.3

Table 2: Comparison of the model complexity.

we can see that with VGGish and OpenL3 embeddings, regularizing the feature space provides better results than simply using the teachers' soft targets. On the other hand, for the PaSST and PANNs embeddings, classical knowledge distillation provides competitive results. The possible reason is that the soft targets given by "weak" teachers might have provided too much incorrect information to the students while the high-quality soft targets generated by the "strong" teachers provide good guidance for the students' training.

The combination system EAsT_{KD} gives us better results with PaSST and PANNs embeddings (with the exception of no noteworthy improvement with the PaSST embedding on the OpenMIC dataset) while for VGGish and OpenL3 embeddings, the performance is not as good as $\text{EAsT}_{\text{Final}}$ or EAsT_{All} in most cases. This observation is in accordance with our speculation that traditional knowledge distillation performs best with a "strong" teacher. While learning from audio embeddings benefits a student network even more in the presence of a "strong" teacher, learning from "weak" embeddings can still improve the model's performance.

5.2 Comparison of Model Complexity

Table 2 lists the number of parameters as well as rough inference speed measurements [9] of the models.

The numbers of parameters only take the backbone structure (i.e., excluding the final classification head) into account so that it does not vary across datasets with different numbers of classes. Iterations per second are tested with 128×1000 input spectrograms.

We can see that Mobile FCN and CP ResNet are much faster in inference than pre-trained models.

5.3 Limited Training Data

To investigate the impact of limited training data on our methods, we present the system performances for reduced training data, i.e., for 25%, 50%, and 75% of the original training data. The results are shown in Figure 2. We use VGGish and PaSST as the pre-trained embeddings.

We can observe that limiting the training data has the greatest impact on the baseline systems, which show the biggest performance drop.

On the OpenMIC dataset, $\text{EAsT}_{\text{Cos-Diff}}$ and $\text{EAsT}_{\text{Final}}$ have similar decreases in mAP, and the KD system is less affected. An interesting finding is that when the VGGish embedding is used, KD shows better performance for limited data amounts while it is outperformed by $\text{EAsT}_{\text{Cos-Diff}}$ and $\text{EAsT}_{\text{Final}}$ when the whole OpenMIC dataset is available. This means using embeddings as teachers might still require a sufficient amount of data to have good guidance on the student models.

On the MagnaTagATune dataset, however, the $\text{EAsT}_{\text{Cos-Diff}}$ and $\text{EAsT}_{\text{Final}}$ systems show less performance decay than either KD or the baseline when the training data is limited. This suggests that in our training settings, there is no certain answer to which method is least affected by the lack of training data, and the answer might be dependent on specific tasks, models, and data.

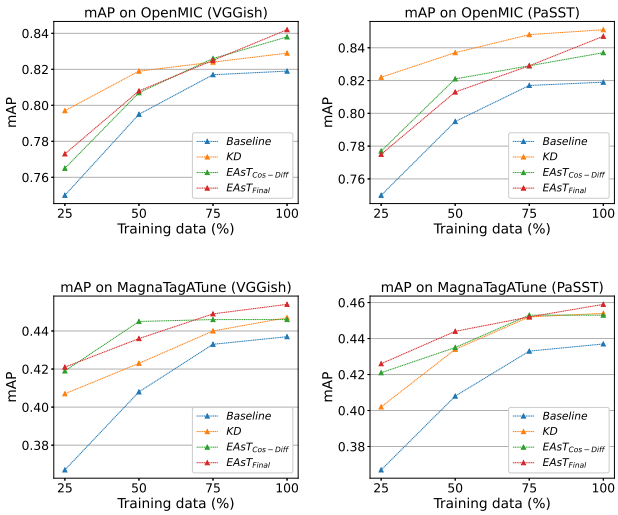


Figure 2: Results with limited training data on two datasets.

6. CONCLUSION AND FUTURE WORK

In this paper, we explored the use of audio embeddings as teachers to regularize the feature space of low-complexity student networks during training. We investigated several different ways of implementing the regularization and tested its effectiveness on the OpenMIC and MagnaTagATune datasets. Results show that using embeddings as teachers enhances the performance of the low-complexity student models, and the results can be further improved by combining our method with a traditional knowledge distillation approach.

Future work will investigate the performance of our method on a wider variety of downstream tasks and embeddings. Moreover, as there have been a wide variety of models to extract audio and music embeddings, we speculate that using an ensemble of different pre-trained embeddings also has considerable potential. Finally, the flexibility of feature-based knowledge distillation offers a wide range of possible algorithmic modifications. Our focus will be on evaluating different distance measures and regularizing the network using features from different stages of the teacher network instead of using only the output embeddings.

[9] reference GPU: NVIDIA 2070 Super

7. REFERENCES

[1] 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.

[2] E. Humphrey, S. Durand, and B. McFee, "Openmic-2018: An open data-set for multiple instrument recognition," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2018, pp. 438–444.

[3] 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 International Society for Music Information Retrieval Conference (ISMIR)*, 2009, pp. 387–392.

[4] S. Gururani, M. Sharma, and A. Lerch, "An attention mechanism for musical instrument recognition," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2019, pp. 83–90.

[5] K. Koutini, H. Eghbal-zadeh, and G. Widmer, "Receptive field regularization techniques for audio classification and tagging with deep convolutional neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 1987–2000, 2021.

[6] K. Choi, G. Fazekas, and M. Sandler, "Automatic tagging using deep convolutional neural networks," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2016, pp. 805–811.

[7] T. Kim, J. Lee, and J. Nam, "Sample-level cnn architectures for music auto-tagging using raw waveforms," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 366–370.

[8] K. Choi, G. Fazekas, M. Sandler, and K. Cho, "Transfer learning for music classification and regression tasks," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2017, pp. 141–149.

[9] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[10] K. Koutini, J. Schlüter, H. Eghbal-zadeh, and G. Widmer, "Efficient training of audio transformers with patchout," in *Proceedings of INTERSPEECH 2022*, 2022, pp. 2753–2757.

[11] 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, 2020.

[12] 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 International Society for Music Information Retrieval Conference (ISMIR)*, 2022, pp. 256–263.

[13] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015. [Online]. Available: http://arxiv.org/abs/1503.02531

[14] J. Ba and R. Caruana, "Do deep nets really need to be deep?" in *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

[15] L. Yu, V. O. Yazici, X. Liu, J. v. d. Weijer, Y. Cheng, and A. Ramisa, "Learning metrics from teachers: Compact networks for image embedding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (ICCV)*, 2019, pp. 2902–2911.

[16] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2021, pp. 10 347–10 357.

[17] A. Kolesnikov, L. Beyer, X. Zhai, J. Puigcerver, J. Yung, S. Gelly, and N. Houlsby, "Big transfer (bit): General visual representation learning," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020, pp. 491–507.

[18] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 248–255.

[19] 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.

[20] J. Devlin, M.-W. 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 (NAACL)*, 2019, pp. 4171–4186.

[21] P. Alonso-Jiménez, D. Bogdanov, and X. Serra, "Deep embeddings with essentia models," in *Late Breaking Demo of the International Society for Music Information Retrieval Conference (ISMIR)*, 2020.

[22] 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 *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 131–135.

[23] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, and I. Sutskever, "Jukebox: A generative model for music," 2020. [Online]. Available: http://arxiv.org/2005.00341

[24] R. Castellon, C. Donahue, and P. Liang, "Codified audio language modeling learns useful representations for music information retrieval," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2021, pp. 88–96.

[25] E. S. Koh and S. Dubnov, "Comparison and analysis of deep audio embeddings for music emotion recognition," in *AAAI Workshop on Affective Content Analysis*, 2021.

[26] 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 *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2022, pp. 650–658.

[27] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2818–2826.

[28] R. Müller, S. Kornblith, and G. E. Hinton, "When does label smoothing help?" in *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[29] T. Huang, S. You, F. Wang, C. Qian, and C. Xu, "Knowledge distillation from a stronger teacher," in *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[30] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, "Fitnets: Hints for thin deep nets," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

[31] B. Heo, J. Kim, S. Yun, H. Park, N. Kwak, and J. Y. Choi, "A comprehensive overhaul of feature distillation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 1921–1930.

[32] B. Peng, X. Jin, J. Liu, D. Li, Y. Wu, Y. Liu, S. Zhou, and Z. Zhang, "Correlation congruence for knowledge distillation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 5007–5016.

[33] J. Yim, D. Joo, J. Bae, and J. Kim, "A gift from knowledge distillation: Fast optimization, network minimization and transfer learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 4133–4141.

[34] J. Kim, S. Park, and N. Kwak, "Paraphrasing complex network: Network compression via factor transfer," in *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[35] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation: A survey," *International Journal of Computer Vision*, vol. 129, pp. 1789–1819, 2021.

[36] L. Wang and K.-J. Yoon, "Knowledge distillation and student-teacher learning for visual intelligence: A review and new outlooks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, pp. 3048–3068, 2022.

[37] A. Dosovitskiy, J. T. Springenberg, M. Riedmiller, and T. Brox, "Discriminative unsupervised feature learning with convolutional neural networks," in *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

[38] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2020, pp. 1597–1607.

[39] J. Spijkervet and J. A. Burgoyne, "Contrastive learning of musical representations," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2021, pp. 673–681.

[40] P. Alonso-Jiménez, X. Serra, and D. Bogdanov, "Music representation learning based on editorial metadata from discogs," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2022, pp. 825–833.

[41] P. Seshadri and A. Lerch, "Improving music performance assessment with contrastive learning," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2021, pp. 634–641.

[42] Y.-N. Hung and A. Lerch, "Feature-informed embedding space regularization for audio classification," in *Proceedings of the European Signal Processing Conference (EUSIPCO)*, 2022, pp. 419–423.

[43] ——, "Feature-informed latent space regularization for music source separation," in *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, 2022.

[44] G. J. Székely, M. L. Rizzo, and N. K. Bakirov, "Measuring and testing dependence by correlation of distances," *The Annals of Statistics*, vol. 35, no. 6, pp. 2769–2794, 2007.

[45] Y. Xu and W. Yin, "Block stochastic gradient iteration for convex and nonconvex optimization," *SIAM Journal on Optimization*, vol. 25, no. 3, pp. 1686–1716, 2015.

[46] X. Zhen, Z. Meng, R. Chakraborty, and V. Singh, "On the versatile uses of partial distance correlation in deep learning," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022, pp. 327–346.

[47] X. Gastaldi, "Shake-shake regularization of 3-branch residual networks," in *Workshop Track of the International Conference on Learning Representations (ICLR)*, 2017.

[48] M. Won, A. Ferraro, D. Bogdanov, and X. Serra, "Evaluation of cnn-based automatic music tagging models," in *Proceedings of the Sound and Music Computing (SMC)*, 2020, pp. 331–337.

[49] H.-H. Chen and A. Lerch, "Music instrument classification reprogrammed," in *Proceedings of the International Conference on Multimedia Modeling (MMM)*, 2023, pp. 345–357.

[50] M. Won, S. Chun, and X. Serra, "Toward interpretable music tagging with self-attention," 2019. [Online]. Available: http://arxiv.org/1906.04972

[51] M. Won, S. Chun, O. Nieto, and X. Serrc, "Data-driven harmonic filters for audio representation learning," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 536–540.

[52] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," 2017. [Online]. Available: http://arxiv.org/abs/1704.04861

[53] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

[54] J. Cramer, H.-H. Wu, J. Salamon, and J. P. Bello, "Look, listen, and learn more: Design choices for deep audio embeddings," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 3852–3856.

[55] R. Arandjelovic and A. Zisserman, "Look, listen and learn," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 609–617.

[56] 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 *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 776–780.

[57] 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 *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

SCOREPERFORMER: EXPRESSIVE PIANO PERFORMANCE RENDERING WITH FINE-GRAINED CONTROL

Ilya Borovik
Skoltech, Russia
ilya.borovik@skoltech.ru

Vladimir Viro
Peachnote GmbH, Germany
vladimir@peachnote.de

ABSTRACT

We present ScorePerformer, an encoder-decoder transformer with hierarchical style encoding heads for controllable rendering of expressive piano music performances. We design a tokenized representation of symbolic score and performance music, the Score Performance Music tuple (SPMuple), and validate a novel way to encode the local performance tempo in a local note time window. Along with the encoding, we extend a transformer encoder with multi-level maximum mean discrepancy variational autoencoder style modeling heads that learn performance style at the global, bar, beat, and onset levels for fine-grained performance control. To offer an interpretation of the learned latent spaces, we introduce performance direction marking classifiers that associate vectors in the latent space with direction markings to guide performance rendering through the model. Evaluation results show the importance of the architectural design choices and demonstrate that ScorePerformer produces diverse and coherent piano performances that follow the control input.

1. INTRODUCTION

Musical expression is the human touch that transforms a written piece of music into an emotionally moving experience. In musical interpretation and performance, the musician interprets a musical score and translates the intended expression through the control of the musical instrument, the sound of which conveys affect and emotion to the listener [1, 2]. However, effective control of musical instruments often requires considerable expertise and training, making musical expression less accessible than it could be.

Deep learning music performance models reduce the need for musical expertise and open up new ways to create and perform music [3, 4]. To render expressive performances of written music [5, 6], the models mix recurrent neural networks to learn temporal dependencies in music with variational autoencoders to encode performance style and enable controllable generation [7–11]. The models are trained on real and categorical score and performance features for aligned score and performance notes.

© I. Borovik and V. Viro. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** I. Borovik and V. Viro, "ScorePerformer: Expressive Piano Performance Rendering with Fine-Grained Control", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

The related task of symbolic music generation is approached differently. Transformer models [12] are primarily utilized due to their ability to effectively learn long-term dependencies in music sequences [13–17]. The symbolic music is encoded as sequences of musical tokens, either individual [15, 18, 19] or stacked into tuples [16, 20]. Similar approaches could be applied to the task of rendering expressive music performances for written compositions.

Aiming to advance the research and make musical expression more accessible, we develop ScorePerformer [1], a piano music performance rendering model with interactive fine-grained performance style control. The model combines encoder and decoder transformers [12] with hierarchical maximum mean discrepancy variational autoencoders [21, 22] that encode performance style representations at the global, bar, beat, and onset levels.

To interpret the learned style embedding spaces, we train embedding classifiers that associate local performance contexts with written musical score direction markings. For each marking, we use the classifier predictions to compute the average delta vectors in the style space from negatively to positively classified style embeddings. These vectors provide quantified model control inputs to move the performance rendering per given direction marking.

For data encoding, we design a tokenized representation of score and performance music, a Score-Performance Music tuple (SPMuple). It introduces a local window onset tempo function that produces smoother and more robust tempos than inset-bar, -beat, or -onset tempo functions.

The experiments and evaluation results show that the model trained on the designed encoding successfully captures different performance styles, can sample diverse and coherent piano performances, and can be used for expressive performance rendering with fine-grained style control.

Our main contributions are:

1. We extend transformers for expressive piano performance rendering with hierarchical style encoding and control at the global, bar, beat, and onset levels;
2. We design a tokenized encoding for aligned score and performance music that proposes an efficient local tempo computation function;
3. We introduce performance direction classifiers to provide musical language-driven performance control by modifying the learned style latent spaces.

[1] Source code and demo are available at: https://github.com/ilya16/scoreperformer

2. RELATED WORK

Expressive Music Performance: Recent expressive music performance rendering models mainly utilize deep learning methods [3, 6]. Jeong et al. [8] and Maezawa et al. [9] use conditional variational autoencoders for performance style encoding and recurrent neural networks for expressive performance rendering. Rhyu et al. [11] allow performance style to be intuitively "sketched" by a set of learned latent representations. We propose to use transformers with self-attention mechanisms [12] to infer patterns in music performance and model its style through hierarchical style encoding heads.

Symbolic Music Generation: Symbolic music generation with deep learning [4] is dominated by transformers for learning long-term sequential musical patterns [13, 15–17, 23] and variational autoencoders for unsupervised style encoding and control [19, 23–26]. The models offer unconditional or priming melody-based music generation [14], global control of performance style [14, 25] or fine-grained control of music through learned high-level features [23, 26] and descriptions [19]. Our model is close to the melody-conditioned transformer autoencoder [14], but introduces modifications for the task of score-based performance rendering with style control.

Symbolic Music Encoding: The simplest way to encode symbolic music is a MIDI-like encoding with note-on, note-off, and time-shift events [13, 18]. REMI [15], REMI+ [19], and Compound Word [16] replace position shifts with absolute bar, position, and beat tempo tokens. OctupleMIDI [20] shortens sequence lengths by stacking note attributes into tuples of 8 tokens. For expressive music performance rendering, it is common to mix real, categorical and pianoroll-based score and performance features parsed from MusicXML and MIDI files [8, 9, 11, 27]. Transformers work well with tokenized data [12, 28, 29]. Inspired by OctupleMIDI, we design a tuple-like token encoding that naturally fits aligned score-performance data.

3. DATA ENCODING

3.1 Score and Performance Data Matching

Expressive music performance rendering models require datasets of aligned score and performance music [5,30]. In this work, we consider piano music performances in MIDI format and use the following data preparation pipeline. First, we compute alignments using Nakamura's alignment tool [31]. The alignments may contain errors, such as alignment holes or close performance notes aligned with distant and unrelated score notes. Following the literature [8, 32], we revise the alignments and filter out notes that deviate from the local performance tempo. After the cleanup, we omit performances with less than 80% aligned notes. Finally, to achieve a perfect match, we remove extra performed notes and interpolate missing notes using the local performance tempos and dynamics, since taking only matched notes and discarding score notes can result in the removal of important chord and bar information.

3.2 SPMuple Encoding

We introduce the Score-Performance Music tuple (SPMuple), a tokenized representation for aligned symbolic score and performance music. It encodes performed notes using tuples of 8 score and 4 performance tokens.

Score Tokens: a set of features extracted from the score MIDI. **Pitch** is a MIDI pitch number in the range 21 to 108. **Duration** is a score note value, encoded by 128 tokens with high and low resolution tokens for short and long durations, respectively [20, 33]. **Bar** is an index of the musical bar to which the note refers, ranging from 0 to the maximum bar in the data. **Position** is the position of the note in the bar, one of 128 tokens with 64th note resolution. **TimeSignature** is the time signature of the beat containing the note, a set of 22 tokens for 2nd, 4th, and 8th note beat lengths, with a maximum bar length of 2 whole notes for 2nd note, and 1.5 for 4th and 8th note. **OnsetShift** is the positional interval between the current and previous note onsets (chords). **NotesInOnset** and **PositionInOnset** are the number of notes and the index of the note in the onset, ranging from 1 to 12, notes are ordered by pitch.

Performance Tokens: a set of performance features extracted from the performance MIDI and processed using the aligned score note features. **Velocity** is a MIDI velocity from 1 to 127. **Tempo** is the performance tempo at the bar, beat or onset level, encoded by a geometric sequence of 121 tokens for beats per minute tempos from 15 to 480. **RelOnsetDeviation** models the exact timing of the note, encoded as the ratio of the absolute note-onset position deviation to the inter-onset interval scaled by the local onset tempo using 161 tokens for values in the range -2 to 2. **RelPerformedDuration** is an articulation of the performed note, computed as the ratio of the performed duration to the score duration, scaled by the local onset tempo, and encoded by 121 tokens for logarithmically distributed values between 0.1 and 3.

The score and performance token sequences are sorted by score note start position, pitch and duration.

3.3 Local Tempo

Inset-onset tempos are noisy and have very high variance, while beat and bar tempos are smoother but still fluctuate at beat/bar boundaries, which can lead to degraded musical experience [34–36]. We design a smooth alternative, local onset tempos, weighted with respect to previous onsets in the local onset time window.

Let $\{\mathrm{IOI}_i^s\}$ and $\{\mathrm{IOI}_i^p\}$ be the sets of score and performance inter-onset intervals between the onset o and N preceding onsets o_i in the time window W. The weights w_i^o for inter-onset tempos $\frac{\mathrm{IOI}_i^s}{\mathrm{IOI}_i^p}$ are computed as:

$$w_i^o = 1 - \frac{\mathrm{IOI}_i^p}{\max_j\{\mathrm{IOI}_j^p\} + 10^{-2}} \tag{1}$$

The weights give more attention to the closest preceding onsets, but still consider the more distant onsets to smooth the local tempo. Based on the decoding quality, we set the time window length W to 8s as the optimal one. In

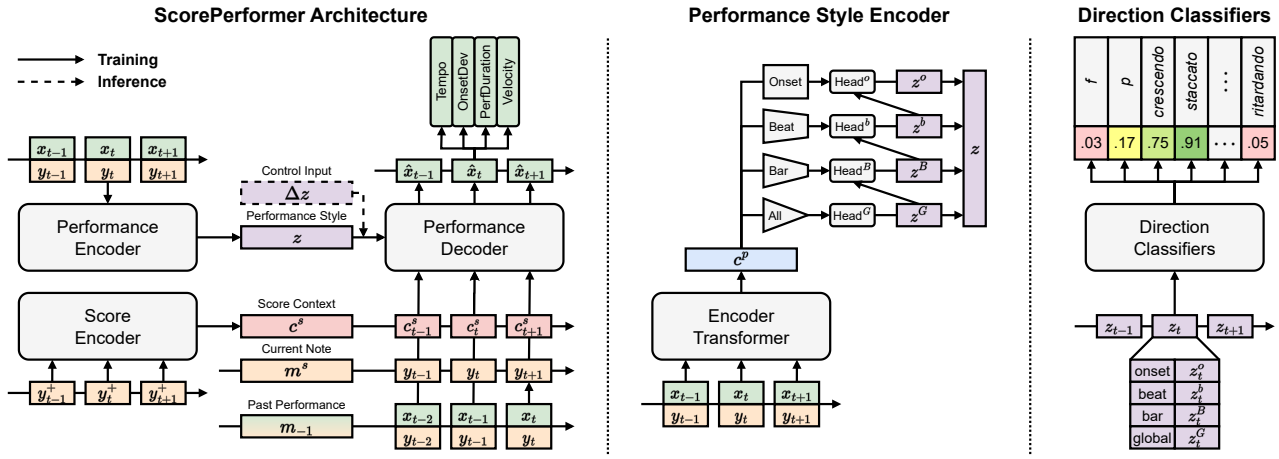


Figure 1. The overall architecture of ScorePerformer, hierarchical style encoding heads and direction classifiers.

addition to the window W, we filter out the nearest onsets with $\mathrm{IOI}_i^p < 0.5$ to reduce the effect of immediate tempo changes, and take at least $N_{\min} = 8$ past onsets with any IOI^p to have enough points for smoothing ($N \geq N_{\min}$).

4. MODEL

With a focus on hierarchical performance style control and efficient training on tokenized sequences, we present **ScorePerformer**, an encoder-decoder model that combines transformers [12] and maximum mean discrepancy variational autoencoders (MMD-VAE) [21,22] for controllable expressive rendering of piano performances for written scores. The model is illustrated in Figure 1.

4.1 Model Architecture

Score Encoder is an encoder transformer that computes a contextual representation of the written music. It maps a score note sequence $y^+ \in \mathbb{N}^{N\times 10}$ (score tokens y + score tempos and velocities) to note embeddings $c^s \in \mathbb{R}^{L\times D}$.

Performance Encoder is an encoder transformer that computes performance style representations at different levels of the musical hierarchy. It takes a sequence of music tuples of score and performance tokens $m = [y, x]$, $y \in \mathbb{N}^{N\times 8}, x \in \mathbb{N}^{N\times 4}$, and outputs performance context embeddings $c^p \in \mathbb{R}^{N\times D}$. The embeddings are grouped and averaged over the entire sequence, bars, beats, and onsets, and iteratively passed through conditional linear layers to compute global, bar, beat, and onset latents z^G, z^B, z^b, and z^o. With the idea of learning missing lower-level details hierarchically, at each step t the latent z_t^* depends on the context c_t^p and all higher-level latents containing the note, e.g. $z_t^b = f_\phi^b(c_t^p, z_t^G, z_t^B)$. All note latents are stacked to produce note-level style embeddings $z \in \mathbb{R}^{N\times D_z}$.

The latent spaces are fit into the Gaussian distribution using a maximum mean discrepancy objective:

$$\mathcal{L}_{\mathrm{MMD}}(p\|q) = \mathbb{E}_{p(z),p(z')}[k(z,z')] + \mathbb{E}_{q(z),q(z')}[k(z,z')] - 2\mathbb{E}_{p(z),q(z')}[k(z,z')], \quad (2)$$

where $k(z,z') = e^{-\frac{\|z-z'\|^2}{2\sigma^2}}$ is a Gaussian kernel.

We use MMD-VAE [21, 22] to solve issues with posterior collapse and latent space holes [37] common to conventional variational autoencoders [38], especially, when trained on sequential data [39].

Performance Decoder is a decoder transformer that renders performance by sequentially predicting performance tokens x_t for score note tokens y_t. The input token sequence combines two sequences: 1) a sequence $m^s = y$ with the current score notes to be rendered; 2) a sequence $m_{-1} = [y_{-1}, x_{-1}]$ shifted one step into the past score y_{-1} and rendered performance tokens x_{-1} describing the past performance history. To reuse the SPMuple token embedder, the first sequence is extended with the masked performance tokens. The two sequence embeddings are concatenated with the score context c^s and passed to the transformer layers together with the style embeddings z.

We use style-adaptive layer normalization (SALN) [40] and pass style embeddings z to the decoder's layer normalization layers, rather than concatenating the style and input token embeddings, to increase the focus on the performance style at each transformer layer.

The performance decoder minimizes the negative log-likelihood for the sequence of performance tokens x:

$$\mathcal{L}_{\mathrm{perf}} = -\sum_{t=1}^{N} \log p_\theta(x_t | x_{<t}, y_{\leq t}, c_{\leq t}, z_{\leq t}) \quad (3)$$

4.2 Transformer Modifications

Discrete+Continuous Tokens: Discrete musical tokens do not explicitly encode the absolute and relative information about note attributes, e.g. that pitches C2, C3, and C4 differ by an octave, or that velocity 80 is louder than 60. We mix discrete and continuous tokens by summing learned discrete token embeddings with delta embeddings provided by a learned nonlinear mapping of the real values associated with tokens to the token embedding dimensions.

Relative Attention: We use the learned ALiBi relative positional bias [41] in the decoder and the learned bidirectional symmetric bias [42] in the encoder for efficient interpolation to sequence lengths not seen during training.

Other: We use single key-value attention heads [43] to speed up decoding, SwiGLU activation [44, 45] in feedforward layers, reuse token embedding weights between the encoders and decoder since they share token vocabularies, and tie input and output embeddings in the decoder [45].

4.3 Performance Direction Classifiers

We provide an intuitive interpretation of the learned style embedding space by training performance direction classifiers on the learned note style embeddings. We extract performance direction markings from MusicXML files and associate score notes with performance direction labels where they are present. We train classifiers for **dynamics** (degrees of *piano* and *forte*), **dynamic changes** (*crescendo* and *diminuendo*), **tempo** (*adagio*, *largo*, etc.), **tempo changes** (*accelerando*, *ritardando*, etc.) and **note articulation** binary classes (*staccato*, *fermata*, etc.).

Classifiers take as input the combined note-level performance style embeddings $z = [z^G, z^B, z^b, z^o]$ and output the probabilities of directions being performed in a given performance context. The module minimizes the sum of cross entropy losses for K classifiers with C_k classes each:

$$\mathcal{L}_{\text{clf}} = \sum_{k=1}^{K} \mathcal{L}_{\text{clf}}^{k} = -\frac{1}{N} \sum_{k=1}^{K} \sum_{t=1}^{N} \sum_{c=1}^{C_k} d_{t,c}^{k} \log(\hat{d}_{t,c}^{k}), \quad (4)$$

where $d_{t,c}^{k}$ and $\hat{d}_{t,c}^{k}$ are true and predicted labels for direction c of the classifier k at step t.

Given the smoothness of the learned latent space, the differences between embeddings with high and low classification scores for a given marking may provide a direction in the latent space to move the generation toward the marking. We compute and use mean per-marking delta embeddings to control performance rendering. Since markings are related to defined musical concepts, we can map natural language commands, such as *"play more piano here"* or *"switch to largo"*, to quantitative model control inputs.

4.4 Training and Inference

The total loss minimized by the model during training is:

$$\mathcal{L} = \mathcal{L}_{\text{perf}} + \mathcal{L}_{\text{MMD}} + \mathcal{L}_{\text{clf}} \quad (5)$$

To avoid overfitting of the decoder to lower-level performance embeddings during training, we drop bar, beat, and onset embeddings with probabilities of 0.1, 0.2, and 0.4, respectively. The embeddings are dropped inclusively, i.e. if the bar latent is dropped out, all beat and onset latents are also dropped. Additionally, the classifiers are trained on detached style embeddings z, as we found the model to overfit the unbalanced direction markings labels.

During inference, the sampled or modified reference performance embeddings can be used to control the rendering of the music performance. Based on the learned style spaces, the control can range from high-level global to low-level onset. The extracted performance direction delta embeddings can be used to provide intuitive, command-driven performance manipulation. The model supports real-time inference on the CPU for use in interactive applications.

5. EXPERIMENTS

Datasets: For all experiments, we use the ASAP dataset of matched piano scores and performances [46], preprocessed as described in Section 3.1. The prepared dataset represents 212 musical compositions by 15 composers with a total of 937 performances, 79 hours of performed music. The data is divided into training and evaluation sets with an approximate ratio of 9:1 for the number of performances in the entire dataset and for each composer.

Implementation: The SPMuple data encoding is implemented using `miditok`'s [33] MIDI tokenizer interface. The encoders and decoders in all experiments have a hidden dimension of 256, 4 layers, and 4 attention heads, except for the score encoder, which has 2 layers. The token embedding dimension is set to 128 for each token type, the projected embedding dimension for input embeddings is set to 256. The global, bar, beat, and onset latent dimensions are set to 32, 20, 8, and 4, respectively.

Training: The maximum sequence length during training is set to 256 tokens. To regularize the model and artificially increase the variety of data, we augment the data with sampled pitch shifts (up to ± 3 semitones) and velocity shifts (up to ± 12 MIDI values). In addition, we randomly replace real performances with deadpan performances with a probability of 25% to allow the model to learn the style of both expressive and inexpressive music. We use the ADAM optimizer [47] with an initial learning rate of $2 \cdot 10^{-4}$, decaying by 0.995 after each epoch. Models are trained for 70,000 iterations with batch size 128.

Evaluation: We conduct three sets of experiments: 1) evaluation of the designed data encoding and different local tempo calculation functions; 2) comparison of different latent style hierarchies and their impact on performance rendering; 3) an ablation study on the model architecture design. For the metrics, we use Pearson correlation [9, 11, 48] and mean absolute error for performance features: inter-onset intervals (IOI), absolute onset deviations (OD), performed note durations (PD), and velocity (Vel). We generate 3 samples for each performance in the evaluation set and compute and average the metrics between the ground truth and the generated performances, decoded to MIDI. The errors are measured in seconds, except for velocities, which are measured in MIDI velocity values. After the objective evaluation, we analyze the generation and control capabilities of the designed ScorePerformer model.

6. EVALUATION

6.1 Encoding and Local Tempos

The tokenized representation of performance is not lossless, since some information is lost during feature quantization. We evaluate the decoding quality and performance of ScorePerformer on sequences encoded using SPMuple with different local tempo functions.

Table 1 shows the evaluation results. The local window onset tempo function (Section 3.3) shows the least degradation in decoding quality for inter-onset intervals and onset deviations. It captures local tempo changes and note

	Decoded						**Generated,** $\Delta z = 0$							
	Error ↓			Correlation ↑			Error ↓				Correlation ↑			
Tempo	IOI	OD	PD	IOI	OD	PD	IOI	OD	PD	Vel	IOI	OD	PD	Vel
Bar	0.092	0.002	0.026	0.770	0.953	0.954	0.140	0.012	0.063	**2.354**	0.650	0.361	0.837	0.940
Beat	0.084	0.002	0.027	0.836	0.971	0.958	0.116	0.009	0.066	2.627	0.727	0.406	0.854	0.932
Onset	**0.019**	**0.001**	**0.006**	0.921	0.977	**0.982**	0.124	0.011	0.056	2.856	0.709	0.339	0.890	0.932
Window	0.028	**0.001**	0.011	**0.963**	**0.985**	0.979	**0.090**	**0.008**	**0.048**	2.583	**0.901**	**0.538**	**0.907**	**0.943**

Table 1. Encoding evaluation on decoded performances and performances generated with unaltered style embeddings from the performance encoder. IOI – inter-onset interval, OD – onset deviation, PD – performed duration, Vel – velocity.

G	B	b	o	z	IOI	OD	PD	Vel
32	**20**	**8**	**4**	**64**	**0.901**	**0.538**	**0.907**	**0.943**
32	20	12	✗	64	0.464	0.194	0.739	0.861
32	32	✗	✗	64	0.417	0.067	0.722	0.812
64	✗	✗	✗	64	0.327	0.066	0.658	0.576
✗	32	✗	✗	32	0.410	0.069	0.702	0.792
✗	✗	12	✗	12	0.384	0.066	0.711	0.767
✗	✗	✗	4	4	0.590	0.063	0.735	0.748
32	20	8	✗	60	0.410	0.065	0.764	0.847
32	20	✗	4	56	0.842	0.224	0.881	0.857
32	✗	8	4	44	0.863	0.386	0.886	0.913
✗	20	8	4	32	0.890	0.485	0.904	0.939

Table 2. Correlation with ground truth performances for samples generated by models trained with different combinations of latent hierarchies. G – global, B – bar, b – beat, o – onset, and z – total latent dimensions.

timing more efficiently than bar, beat and onset tempos. These findings are supported by the generation results. The model trained with local window tempo tokens renders samples with smaller errors and closer to the ground truth than the models trained with bar, beat, or onset tempo tokens. In particular, it shows more consistency in modeling local tempo changes and note timing. For future work, the encoding could be further improved by incorporating pedals, an essential element of piano performance [49].

6.2 Style Embedding Hierarchies

Table 2 shows the impact of different learned style embedding hierarchy combinations in ScorePerformer on the quality of performance rendering. Replacing lower-level latents with higher-level ones, using only a single level, or omitting any level of the hierarchy leads to a decrease in quality for all musical features. The lower-level onset latents account for most of the variation in performance features, while the higher-level latents provide the missing performance timing, articulation, and dynamics information at the beat, bar, and global levels. The results suggest that a hierarchical style representation is advantageous for modeling global and local changes in music performance. The search for an optimal configuration of latent dimensions is beyond the scope of this study.

	IOI	OD	PD	Vel
ScorePerformer	**0.901**	**0.538**	**0.907**	0.943
w/o Score Encoder	0.885	0.526	0.889	**0.951**
w/o input seq. m^s	0.844	0.422	0.895	0.925
w/o SALN	0.871	0.469	0.920	0.930
w/o in-out emb. tie	**0.901**	0.459	0.873	**0.951**
w/o Continuous Tokens	0.576	0.116	0.747	0.561

Table 3. Evaluation of model configurations using the correlation between ground truth and generated performances.

6.3 Ablation Study

The ablation study on the ScorePerformer model is summarized in Table 3. Removing any of the proposed design choices degrades the quality for all features in almost all cases. The score encoder adds a local future score context to the decoder and contributes to a slight quality improvement. The same is true for the additional decoder input sequence m^s, which explicitly highlights the currently rendered score notes. Without style-adaptive layer normalization or input-output embedding weight sharing, the correlation for timing features decreases. The most noticeable quality degradation occurs after using only discrete tokens without continuous input tokens, demonstrating the positive impact of value-aware inputs on model predictions.

6.4 Performance Embeddings Analysis

We explore the learned performance style spaces using the trained performance direction marking classifiers. We take the style embeddings z for note onsets in the dataset and project them into two dimensions using principal component analysis [50]. Figure 3 shows the projected embeddings labeled by the selected dynamics, tempo, and articulation markings classifiers and their ground truth labels. We can see the gradient moving from the light colors (high probabilities) to the darker colors (low probabilities). Despite the class imbalances and low representation of some labels in the dataset, the positive classifier predictions match the areas of the ground truth labels shown in the right plots for each marking. This suggests that the vectors for moving the performance toward the markings exist in the original latent style space and can be used to attempt to control the performance rendering through the model.

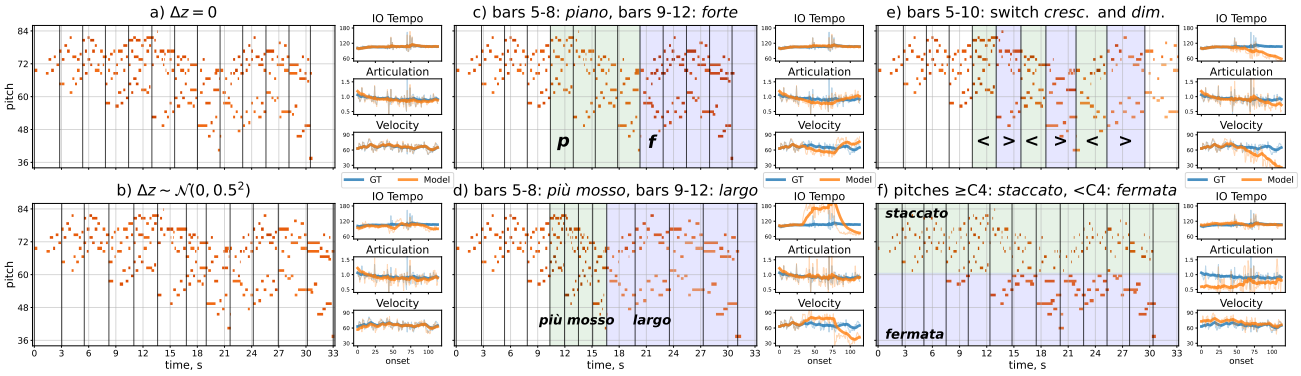


Figure 2. Pianorolls and performance features (inter-onset tempo, articulation, and velocity) for the first 12 musical bars of Bach's "Prelude and Fugue No.19", rendered by ScorePerformer with unconditional or conditional style control. The title of each plot indicates the form of the control input. Colored areas highlight the regions with the applied control.

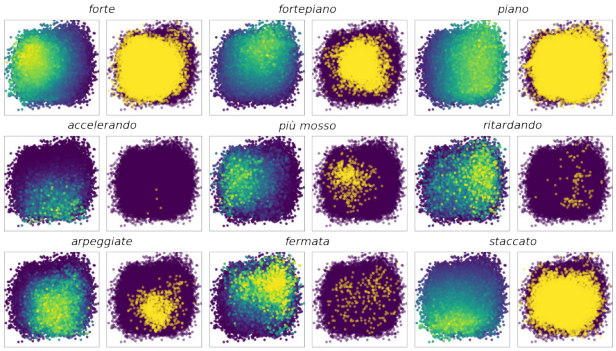


Figure 3. Projected style embeddings classified by chosen direction marking classifiers. The left and right plots for each marking highlight predicted and ground truth labels.

The direction classifiers can also be used to analyze performance practices. For example, take all performance contexts with a given notated performance direction marking and sort them by the classification scores using the associated direction classifier. Further analysis of the score contexts can provide insight into the reasons why musicians follow or interpret differently certain markings.

6.5 Performance Rendering Control

For performance rendering control, we add control embeddings Δz to the encoded style embeddings and pass them to the decoder. We analyze both uncontrolled generation with sampled control embeddings and direction-based control using the computed delta latents for markings.

Figure 2 shows examples of music performance rendering for a composition from the evaluation set. The sample (a) shows the successful reconstruction of the performance variations from the encoded style embeddings. When generated using sampled delta latents (b), the added noise is transferred to higher variations in tempo than in articulation and dynamics. In our observations, small amounts of delta noise can result in both pleasant and diverse samples.

From the evaluation of the performance direction based control, we can see that in most cases the model follows the musical meaning of the marking. Example (c) shows that *piano* and *forte* delta embeddings lead to the expected decrease and increase in dynamics. The *più mosso* (more movement, faster) and *largo* (slowly and broadly) in the example (d) lead to the expected changes in tempo, articulation and dynamics. An interesting behaviour can be found in example (e), where the model values one marking over the other. During the alternation of *crescendo* and *diminuendo*, the model follows *diminuendo* more and falls on the path of slow and quiet performance. The last example (f) shows that the control can also be applied effectively to individual notes. As the definitions suggest, the *staccato* on higher pitched notes makes them more abrupt, and the *fermata* on other notes holds the notes a bit longer.

Despite the positive examples of piano performance rendering control, the model has some limitations. The proposed marking delta embeddings encode the highest learned deviations between performance styles and lead to immediate changes in performance, which can sound unnatural. One solution is to scale or interpolate the control inputs for smoother performance changes. Another issue to be addressed is disentangling the learned latent space across direction classes for a more controllable generation. Finally, the study was limited by low performance variation for some markings and compositions in the dataset. We believe that the proposed approach has a high potential for both analytical and musical creativity applications that could be fulfilled with orders of magnitude larger datasets.

7. CONCLUSION

We presented ScorePerformer, an encoder-decoder transformer with hierarchical MMD-VAE style encoding heads for fine-grained controllable expressive rendering of piano music performances. We also introduced performance direction classifiers, trained on performance style embeddings, to map notated direction markings and natural language inputs to model control inputs. Evaluation showed that the model captures performance style variations and follows control intents. Future work will focus on improving the diversity of training data to enable large-scale analysis, and may include in-depth subjective evaluation of the proposed and existing performance rendering models.

8. ACKNOWLEDGEMENTS

We thank Dmitry Yarotsky for his valuable comments during the model development and experimentation. We thank the anonymous reviewers for their critical feedback, which allowed us to improve the paper. The experiments were performed on the Zhores computing cluster [51].

9. REFERENCES

[1] C. Palmer, "Music performance," *Annual review of psychology*, vol. 48, no. 1, pp. 115–138, 1997.

[2] J. Rink, *Musical Performance: A Guide to Understanding*. Cambridge University Press, 2002.

[3] 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.

[4] C. Hernandez-Olivan, J. Hernandez-Olivan, and J. R. Beltran, "A Survey on Artificial Intelligence for Music Generation: Agents, Domains and Perspectives," *arXiv preprint arXiv:2210.13944*, 2022.

[5] A. Kirke and E. R. Miranda, *Guide to Computing for Expressive Music Performance*. Springer, 2013.

[6] C. E. Cancino-Chacón, M. Grachten, W. Goebl, and G. Widmer, "Computational Models of Expressive Music Performance: A Comprehensive and Critical Review," *Frontiers in Digital Humanities*, vol. 5, p. 25, 2018.

[7] 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 36th International Conference on Machine Learning*. PMLR, 2019, pp. 3060–3070.

[8] 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*, 2019, pp. 908–915.

[9] A. Maezawa, K. Yamamoto, and T. Fujishima, "Rendering Music Performance With Interpretation Variations Using Conditional Variational RNN," in *Proceedings of the 20th International Society for Music Information Retrieval Conference*, 2019, pp. 855–861.

[10] H. H. Tan, Y.-J. Luo, and D. Herremans, "Generative modelling for controllable audio synthesis of expressive piano performance," *arXiv preprint arXiv:2006.09833*, 2020.

[11] S. Rhyu, S. Kim, and K. Lee, "Sketching the Expression: Flexible Rendering of Expressive Piano Performance with Self-Supervised Learning," *arXiv preprint arXiv:2208.14867*, 2022.

[12] 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*, vol. 30. Curran Associates, Inc., 2017, pp. 5998–6008.

[13] 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.

[14] K. Choi, C. Hawthorne, I. Simon, M. Dinculescu, and J. Engel, "Encoding Musical Style with Transformer Autoencoders," *arXiv preprint arXiv:1912.05537*, 2019.

[15] Y.-S. Huang and Y.-H. Yang, "Pop Music Transformer: Beat-based Modeling and Generation of Expressive Pop Piano Compositions," *arXiv preprint arXiv:2002.00212*, 2020.

[16] 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, 2021, pp. 178–186.

[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," *arXiv preprint arXiv:2210.10349*, 2022.

[18] 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, pp. 955–967, 2020.

[19] D. von Rütte, L. Biggio, Y. Kilcher, and T. Hoffman, "FIGARO: Generating Symbolic Music with Fine-Grained Artistic Control," *arXiv preprint arXiv:2201.10936*, 2022.

[20] 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.

[21] S. Zhao, J. Song, and S. Ermon, "InfoVAE: Information Maximizing Variational Autoencoders," *arXiv preprint arXiv:1706.02262*, 2017.

[22] A. Gretton, K. Borgwardt, M. Rasch, B. Schölkopf, and A. Smola, "A Kernel Method for the Two-Sample-Problem," in *Advances in Neural Information Processing Systems*, vol. 19. MIT Press, 2006, pp. 513—-520.

[23] S.-L. Wu and Y.-H. Yang, "MuseMorphose: Full-Song and Fine-Grained Piano Music Style Transfer with One Transformer VAE," *arXiv preprint arXiv:2105.04090*, 2021.

[24] A. Roberts, J. Engel, C. Raffel, C. Hawthorne, and D. Eck, "A Hierarchical Latent Vector Model for Learning Long-Term Structure in Music," in *Proceedings of the International Conference on Machine Learning*. PMLR, 2018, pp. 4364–4373.

[25] G. Brunner, A. Konrad, Y. Wang, and R. Wattenhofer, "MIDI-VAE: Modeling Dynamics and Instrumentation of Music with Applications to Style Transfer," *arXiv preprint arXiv:1809.07600*, 2018.

[26] 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.

[27] D. Jeong, T. Kwon, Y. Kim, and J. Nam, "Score and performance features for rendering expressive music performances," in *Music Encoding Conference*. Music Encoding Initiative Vienna, Austria, 2019, pp. 1–6.

[28] 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," *arXiv preprint arXiv:2010.11929*, 2020.

[29] Z. Borsos, R. Marinier, D. Vincent, E. Kharitonov, O. Pietquin, M. Sharifi, O. Teboul, D. Grangier, M. Tagliasacchi, and N. Zeghidour, "AudioLM: a Language Modeling Approach to Audio Generation," *arXiv preprint arXiv:2209.03143*, 2022.

[30] C. E. Cancino-Chacón, "Computational Modeling of Expressive Music Performance with Linear and Nonlinear Basis Function Models," Ph.D. dissertation, Johannes Kepler University Linz, Austria, December 2018.

[31] E. Nakamura, K. Yoshii, and H. Katayose, "Performance Error Detection and Post-Processing for Fast and Accurate Symbolic Music Alignment," in *International Society for Music Information Retrieval Conference*, 2017.

[32] G. G. Xia, "Expressive collaborative music performance via machine learning," Ph.D. dissertation, Carnegie Mellon University, August 2016.

[33] N. Fradet, J.-P. Briot, F. Chhel, A. El Fallah-Seghrouchni, and N. Gutowski, "MidiTok: A Python package for MIDI file tokenization," in *22nd International Society for Music Information Retrieval Conference*, 2021.

[34] S. Dixon, W. Goebl, and E. Cambouropoulos, "Perceptual Smoothness of Tempo in Expressively Performed Music," *Music Perception*, vol. 23, no. 3, pp. 195–214, 2006.

[35] B. H. Repp, "On Determining the Basic Tempo of an Expressive Music Performance," *Psychology of Music*, vol. 22, no. 2, pp. 157–167, 1994.

[36] H. Schreiber, F. Zalkow, and M. Müller, "Modeling and Estimating Local Tempo: A Case Study on Chopin's Mazurkas," in *Proceedings of the 21st International Society for Music Information Retrieval Conference*, 2020, pp. 773–779.

[37] J. Lucas, G. Tucker, R. B. Grosse, and M. Norouzi, "Understanding Posterior Collapse in Generative Latent Variable Models," in *Deep Generative Models for Highly Structured Data, ICLR 2019 Workshop*, 2019.

[38] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[39] S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Józefowicz, and S. Bengio, "Generating Sentences from a Continuous Space," in *Conference on Computational Natural Language Learning*, 2015.

[40] D. Min, D. B. Lee, E. Yang, and S. J. Hwang, "Meta-StyleSpeech: Multi-Speaker Adaptive Text-to-Speech Generation," in *International Conference on Machine Learning*. PMLR, 2021, pp. 7748–7759.

[41] 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.

[42] M. Lee, K. Han, and M. C. Shin, "LittleBird: Efficient Faster & Longer Transformer for Question Answering," *arXiv preprint arXiv:2210.11870*, 2022.

[43] N. Shazeer, "Fast Transformer Decoding: One Write-Head is All You Need," *arXiv preprint arXiv:1911.02150*, 2019.

[44] ——, "GLU Variants Improve Transformer," *arXiv preprint arXiv:2002.05202*, 2020.

[45] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann *et al.*, "PaLM: Scaling Language Modeling with Pathways," *arXiv preprint arXiv:2204.02311*, 2022.

[46] 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 21st International Society for Music Information Retrieval Conference*, 2020, pp. 534–541.

[47] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[48] 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, pp. 887–909, 2017.

[49] S. P. Rosenblum, "Pedaling the Piano: A Brief Survey from the Eighteenth Century to the Present," *Performance Practice Review*, vol. 6, no. 2, p. 8, 1993.

[50] M. E. Tipping and C. M. Bishop, "Probabilistic Principal Component Analysis," *Journal of the Royal Statistical Society Series B: Statistical Methodology*, vol. 61, no. 3, pp. 611–622, 1999.

[51] I. Zacharov, R. Arslanov, M. Gunin, D. Stefonishin, A. Bykov, S. Pavlov, O. Panarin, A. Maliutin, S. Rykovanov, and M. Fedorov, ""Zhores"—Petaflops supercomputer for data-driven modeling, machine learning and artificial intelligence installed in Skolkovo Institute of Science and Technology," *Open Engineering*, vol. 9, no. 1, pp. 512–520, 2019.

ROMAN NUMERAL ANALYSIS WITH GRAPH NEURAL NETWORKS: ONSET-WISE PREDICTIONS FROM NOTE-WISE FEATURES

Emmanouil Karystinaios[1] **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

Roman Numeral analysis is the important task of identifying chords and their functional context in pieces of tonal music. This paper presents a new approach to automatic Roman Numeral analysis in symbolic music. While existing techniques rely on an intermediate lossy representation of the score, we propose a new method based on Graph Neural Networks (GNNs) that enable the direct description and processing of each individual note in the score. The proposed architecture can leverage notewise features and interdependencies between notes but yield onset-wise representation by virtue of our novel edge contraction algorithm. Our results demonstrate that *ChordGNN* outperforms existing state-of-the-art models, achieving higher accuracy in Roman Numeral analysis on the reference datasets. In addition, we investigate variants of our model using proposed techniques such as NADE, and post-processing of the chord predictions. The full source code for this work is available at https://github.com/manoskary/chordgnn

1. INTRODUCTION

Automatic Chord Recognition is one of the core problems in Music Information Retrieval. The task consists of identifying the harmonies or chords present in a musical piece. Various methods have been proposed to address this task using either an audio or symbolic representation of the music [1]. In the symbolic domain, most approaches focus on the related and arguably more complex problem of Automatic Roman Numeral Analysis, which is a functional harmony analysis problem that has its roots in musicological research of Western classical music.

Roman Numeral Analysis is a notational system used in music theory to analyze chord progressions and identify the relationship between chords in a given key. In this system, each chord in a piece of music is assigned a Roman numeral based on its position within the key's scale. For example, in the key of C major, the I chord is C major, the IV chord is F major, and the V chord is G major. Roman Numerals are an important tool for understanding and analyzing the harmonic structure of music, and they are a valuable resource for musicians, composers, and arrangers alike.

In Music Information Retrieval, a lot of work has been done to automate Roman Numeral analysis. However, current approaches still face significant challenges. Some of these are related to the large chord symbol vocabulary. A common way to address this problem is to divide a Roman Numeral into several components (e.g., key, degree, inversion) and transform the analysis into a multitask learning scenario. However, multitask approaches themselves face challenges with interdependencies among tasks. Lastly, Roman Numeral analysis faces a score representation problem related to existing models such as CNNs whose inputs must be in fixed-sized chunks. Recent state-of-the-art approaches follow an audio-inspired strategy, dividing a musical score into fixed-length time frames ("windows") which are then processed by a Convolutional Recurrent Neural Network (CRNN). However, such a representation is unnatural for scores and has the added practical disadvantage of being time-limited (for example regarding notes extending beyond the current window) and, due to the fixed-length (in terms of score time) constraint, capturing varying amounts of musically relevant context.

In this paper, we propose a new method for automatic Roman Numeral analysis based on Graph Neural Networks that can leverage note-wise information to address the score representation issue. Our model, *ChordGNN*, builds on top of existing multitask approaches but introduces several novel aspects, including a graph convolutional architecture with an edge contraction pooling layer that combines convolution at the note level but yields the learned representation at the onset level.

Our proposed method, *ChordGNN*, is evaluated on a large dataset of Western classical music, and the experimental results demonstrate that it outperforms existing state-of-the-art methods, in terms of the commonly used Chord Symbol Recall measure. To address the interdependencies among tasks we investigate the effect of post-processing and other proposed techniques such as NADE and gradient normalization. Finally, we look at a qualitative musical example and compare our model's predictions with other state-of-the-art models.

© 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, "Roman Numeral Analysis with Graph Neural Networks: Onset-wise Predictions from Note-wise Features", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

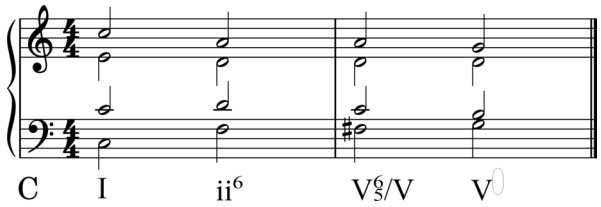


Figure 1. A Roman Numeral analysis for two bars for four-part harmony in C major. Capital letters stand for major quality and lowercase for minor quality. The third chord has a dominant seven as its primary degree and the dominant of C major as its secondary degree. The V_5^6 indicates a major with a seven quality in second inversion. The bass (lowest chord note) of that chord is F sharp, the root is D, and the local key is C major.

2. RELATED WORK

There is a big body of literature covering the topic of Automatic Chord Recognition applied in the audio domain; however, in our work, we focus on the problem of automatic Roman Numeral Analysis in the symbolic domain. It consists of labeling the chords and harmonic progressions in a piece of music using Roman Numerals, where each numeral represents a chord built on a particular scale degree. Numerous approaches have tried to automate Roman Numeral analysis or infer harmonic relations between chords. Notable work includes statistical models such as *Melisma* [2], HMM-based models [3], and grammar-based approaches [4].

In recent years, research has shifted towards a deep learning and data-driven approach. Due to the large vocabulary of possible Roman Numerals, the problem has been divided into several component subtasks, thus resulting in a multitask learning setting [5]. As a multitask problem, a Roman Numeral is characterized by the following components: the primary and secondary degree (as illustrated in Figure 1), the local key at the time point of prediction, the root of the chord, the inversion of the chord, and the quality (such as major, minor, 7, etc.). Although the root can be derived from the other components, it was pointed out by [6] that redundancy is assisting Roman Numeral analysis systems to learn. An example of Roman Numerals and their components can be viewed in Figure 1. Recent state-of-the-art approaches decompose the numeral prediction task to the simultaneous prediction of those 6 components [5–9].

Most deep learning approaches to Roman Numeral analysis are inspired by work in audio classification, cutting a score into fixed-size chunks (in terms of some constant score time unit; e.g., a 32nd note) and using these as input to deep models. Using this quantized time frame representation, [6] introduced a CRNN architecture to predict Roman Numerals. Other work has continued to build on the latter by introducing more tasks to improve performance such as the *AugmentedNet* model [7], or introducing intra-dependent layers to inform in an orderly fashion the prediction of one task with the previously predicted task, such as the model introduced by [8]. Other architectures, such as the CSM-T model, have demonstrated good results by introducing modular networks which treat a score as a sequence of notes ordered first by onset and then by pitch [9].

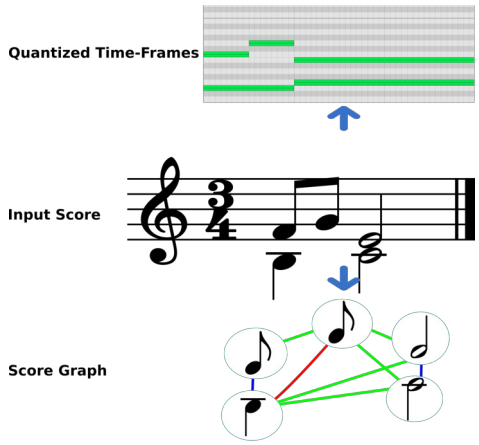


Figure 2. Different representations of the score excerpt shown in the middle. Top: quantized time frame representation, bottom: graph representation.

Should a musicologist perform music analysis on a piece of music, they would consider the individual notes existing in the score. Thus, a time frame representation would come across as unnatural for symbolic music and in particular for such an analysis task. In this paper, we present a method that no longer treats the score as a series of quantized frames but rather as a partially ordered set of notes connected by the relations between them, i.e., a graph. A visual comparison of the two representations is shown in Figure 2. Recently, modeling scores as graphs has also been demonstrated to be beneficial for problems such as expressive performance generation [10], cadence detection [11], voice separation [12], or boundary detection [13].

Automatic Roman Numeral analysis, as a multitask problem, is mostly tackled with hard parameter-sharing models. These models share part of the model across all tasks as an encoder, and then the common embeddings are branched to a classification model per task [6–8]. However, some approaches separate tasks from this paradigm to a more modular or soft parameter sharing approach [9].

In the field of multitask learning, a lot of research has been done on the problem of conflicting gradients during backpropagation in hard parameter-sharing models. Issues with multi-objective optimization have been early addressed by Zhang et al. [14] and recent solutions have been proposed for the multitask setting in the form of dynamic task prioritization [15], gradient normalization [16], rotation matrices [17], or even game-theoretic approaches [18]. In our work, we experimentally evaluate some of these techniques in the multitask setting to investigate whether Roman Numeral analysis subtasks conflict with each other (see Section 5.2).

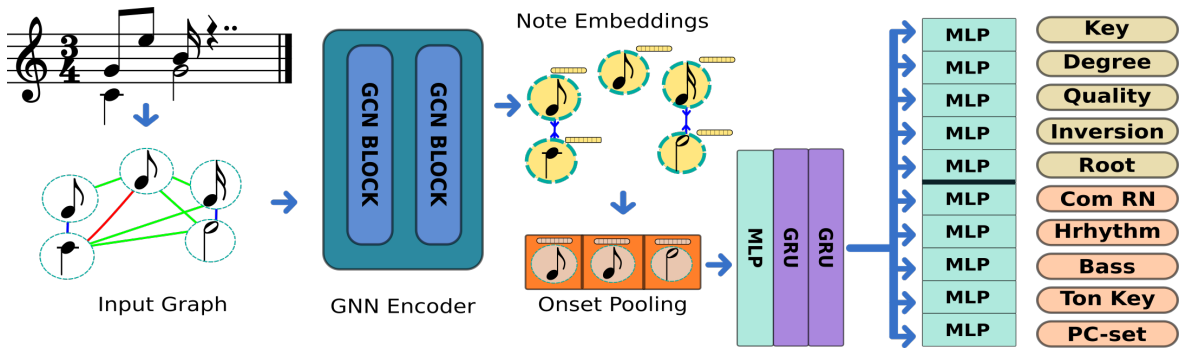


Figure 3. The proposed Architecture Chord-GNN

3. METHODOLOGY

3.1 Roman Numeral Analysis

We already discussed, in Section 2, how Roman Numeral analysis can be viewed as a multi-task problem. In this section, we describe in detail the additional tasks introduced by [7] that we also use for training and prediction. First, let us assume that the prediction can be broken down into specific time points, and each time point is attributed to a unique onset in the score.

The Roman Numeral prediction can be viewed as a simultaneous prediction of the local key, degree (primary and secondary), quality, inversion, and root. Each one of these tasks is a categorical, multiclass classification problem. However, [7] indicated that only three tasks would be sufficient for 98% of the Roman Numeral annotations in our dataset (detailed in Section 4.1). These three tasks comprise the prediction of a restricted vocabulary of common Roman Numeral symbols in combination with the local key and the inversion. We refer to Roman Numeral prediction involving the 5 tasks as *conventional RN*, and the combined prediction of key, inversion, and restricted RN vocabulary *alternative RN*, as RN_{alt}, in accordance with [7].

Several other tasks have been introduced that have been shown to improve the performance of related models [7]. These include the Harmonic Rhythm, which is used to infer the duration of a Roman Numeral at a given time point; the Tonicization task, a multiclass classification task that refers to a tonicized key implied by the Roman Numeral label and is complementary to the local key; the Pitch Class Sets task, which includes a vocabulary of different pitch class sets, and the Bass task, which aims to predict the lowest note in the Roman Numeral label.

3.2 Graph Representation of Scores

Our approach to automatic Roman Numeral analysis no longer treats the score as a sequence of quantized time frames but rather as a graph, which permits us to specify note-wise information such as pitch spelling, duration, and metrical position. We use graph convolution to model interdependencies between notes. We model our score generally following Karystinaios and Widmer [11], but we opt for a heterogeneous graph convolution approach, i.e., including different edge relations/types. Furthermore, we develop an edge contraction pooling layer that learns onset-wise representations from the note-wise embeddings and therefore yields a sequence.

After the edge contraction, we follow [6–8] by adding to the graph convolution a sequence model for the hard-sharing part of our model, and simple shallow multi-layer perceptron heads for each task. In essence, we replace the CNN encoder that works on quantized frames of the score in previous approaches, with a graph convolutional encoder followed by an edge contraction layer. Our proposed architecture is shown in Figure 3.

The input to the GNN encoder is an attributed graph $G = (V, E, X)$ where V and E denote its node and edge sets and X represents the node feature matrix, which contains the features of the notes in the score. For our model, we used pitch spelling, note duration, and metrical position features.

Given a musical piece, the graph-building process creates a set of edges E, with different relation types $\mathcal{R}$. A labeled edge (u, r, v) of type r between two notes u, v belongs to E if the following conditions are met:

- notes starting at the same time: $on(u) = on(v) \rightarrow r = \text{onset}$
- note starting while the other is sounding: $on(u) > on(v) \wedge on(u) \leq on(v) + dur(v) \rightarrow r = \text{during}$
- note starting when the other ends: $on(u) + dur(u) = on(v) \rightarrow r = \text{follow}$
- note starting after a time frame when no note is sounding: $on(u) + dur(u) < on(v) \wedge \nexists v' \in V,\ on(v') < on(v) \wedge on(v') > on(u) + dur(u) \rightarrow r = \text{silence}$

3.3 Model

In this section, we introduce and describe *ChordGNN*, a Graph Convolutional and Recurrent Neural Network. The structure of the network is visually outlined in Figure 3. *ChordGNN* uses heterogeneous graphSAGE [19] convolutional blocks defined as:

$$\begin{aligned} \mathbf{h}^{(l+1)}_{\mathcal{N}_r(v)} &= \text{mean}\left(\{\mathbf{h}^l_u, \forall u \in \mathcal{N}_r(v)\}\right) \\ \mathbf{h}^{(l+1)}_{v_r} &= \sigma\left(W \cdot \text{concat}(\mathbf{h}^l_v, \mathbf{h}^{l+1}_{\mathcal{N}_r(v)})\right) \\ \mathbf{h}^{(l+1)}_{v} &= \frac{1}{|\mathcal{R}|} \sum_{r \in \mathcal{R}} \mathbf{h}^{(l+1)}_{v_r} \end{aligned} \tag{1}$$

where $\mathbf{h}_v^{(0)} = \mathbf{x}_v$ and $\mathbf{x}_u$ is the input features for node u, $\mathcal{N}(u)$ are the neighbors of node u, and σ is a ReLU activation function. We name the output representations of all nodes after graphSAGE convolution $H = \{h_u^{(L)} \mid u \in V\}$ where L is the total number of convolutional layers.

Given the hidden representation H of all nodes, and onset edges $E_{\text{On}} = \{(u, v) \mid on(u) = on(v)\}$, the onset edge contraction pooling is described by the following equations: first, we update the hidden representation with a learned weight, $H' = HW^{(\text{cpool})}$. Subsequently we need to unify the representations for every node u, such that $\forall v \in \mathcal{N}_{\text{On}}(v),\ h_u^{(\text{cp})} = h_v^{(\text{cp})}$:

$$h_u^{(\text{cp})} = h_u + \sum_{v \in \mathcal{N}_{\text{On}}(v)} h_v \tag{2}$$

where, h_u and h_v belong to H'. Subsequently, we filter the vertices:

$$V' = \{v \in V \mid \forall u \in V,\ (v, u) \in E_{\text{On}} \implies u \notin V'\} \tag{3}$$

Therefore, $H^{(cp)} = \{h_u^{(cp)} \mid \forall u \in V'\}$ are the representations obtained. Sorting the representations by the onset on which they are attributed we obtain a sequence $S = [h_{u_1}^{(cp)}, h_{u_2}^{(cp)}, \ldots h_{u_k}^{(cp)}]$ such that $on(u_1) < on(u_2) < \cdots < on(u_k)$.

The sequence S is then passed through an MLP layer and 2 GRU layers. This concludes the hard-sharing part of our model. Thereafter, an MLP head is attached per task, as shown in Figure 3.

For training, we use the dynamically weighted loss introduced by [20]. The total loss $\mathcal{L}_{tot}$ of our network is calculated as a weighted sum of the individual losses for every task, where the weights are learned during training:

$$\mathcal{L}_{\text{tot}} = \sum_{t \in \mathcal{T}} \mathcal{L}_t * \frac{1}{2\gamma_t^2} + \log(1 + \gamma_t^2) \tag{4}$$

where $\mathcal{T}$ is the set of tasks; $\mathcal{L}_t$ is the cross-entropy loss relating to task t; the γ_t are learned scalars that give the weight for each task t; and the $\log$ expression is a regularization term [20].

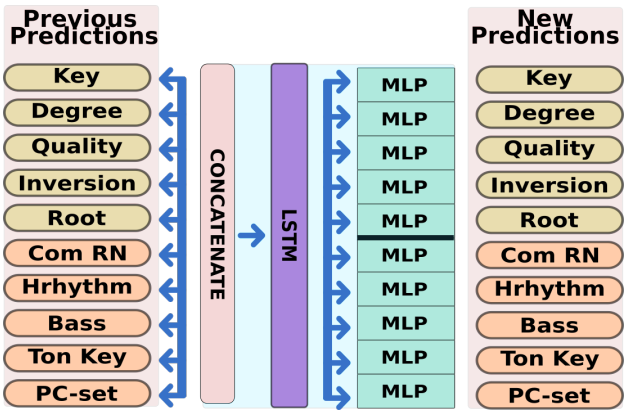


Figure 4. Post-processing of Roman Numeral predictions.

3.3.1 *Post-processing*

We enhance our model with a post-processing phase after the model has been trained. The post-processing phase combines the logits of all tasks' predictions by concatenating them and, then, feeds them to a single-layer bidirectional LTSM block. Then, again the embeddings of the sequential block are distributed to 11 one-layer MLPs, one for each task. The post-processing block is sketched in Figure 4.

4. EXPERIMENTS AND CORPORA

In the experiments, we compare our model, *ChordGNN*, with other recent models for automatic Roman Numeral analysis. We run experiments with our model in the exact same way as described in the paper [7], including the specific data splits, so that our results are directly comparable to the figures reported there. A detailed comparison of the results will be given in Table 1. Furthermore, we develop variants of our model using proposed techniques such as NADE [8], and post-processing of the chord predictions. We report a configuration study of our model on the use of gradient normalization techniques and NADE that should improve results on Multi-Task learning scenarios and avoid common Multi-Task Learning problems such as conflicting gradients. Lastly, we compare our model with the updated version *v1.9.1* of the state-of-the-art model Augmented-Net [21] and datasets.

4.1 Datasets

For training and evaluation, we combined six data sources into a single "Full" Dataset of Roman Numeral annotations in accordance with [7]: the Annotated Beethoven Corpus (ABC) [22]; the annotated Beethoven Piano Sonatas (BPS) dataset [5]; the Haydn String Quartets dataset (HaydnSun) [23]; the TAVERN dataset [24]; a part of the When-in-Rome (WiR) dataset [25, 26]; and the Well-Tempered-Clavier (WTC) dataset [25] which is also part of the WiR dataset.

Training and test splits for the full dataset were also provided by [7]. It is worth noting that the BPS subset splits were already predefined in [5]. In total, approximately 300 pieces were used for training, and 56 pieces were used for testing, proportionally taken from all the different data sources. We draw a distinction for the BPS test set, which includes 32 Sonata first movements and for which we ran an additional experiment. The full test set also includes the 7 Beethoven piano sonatas.

In addition to the above datasets, we include data augmentations identical to the ones described in [7]: texturization and transposition. The texturization is based on a dataset augmentation technique introduced by [27]. The transposition augmentation boils down to transposing a score to all the keys that lie within a range of key signatures that have up to 7 flats or sharps. It should be noted that the augmentations are only applied in the training split.

For our last experiment (to be reported on in Section 5.3 below), we add additional data that were recently introduced by [21]. The additional data include the annotated Mozart

	Model	Key	Degree	Quality	Inversion	Root	RN	RN (Onset)	RN_{alt}
BPS	Micchi (2020)	82.9	68.3	76.6	72.0	-	42.8	-	-
	CSM-T (2021)	69.4	-	-	-	75.4	45.9	-	-
	AugNet (2021)	**85.0**	**73.4**	**79.0**	73.4	**84.4**	45.4	-	49.3
	ChordGNN (Ours)	79.9	71.1	74.8	75.7	82.3	46.2	46.6	48.6
	ChordGNN+Post (Ours)	82.0	71.5	74.1	**76.5**	82.5	**49.1**	**49.4**	**50.4**
Full	AugNet (2021)	**82.9**	67.0	**79.7**	78.8	83.0	46.4	-	51.5
	ChordGNN (Ours)	80.9	70.1	78.4	78.8	84.8	48.9	48.4	50.4
	ChordGNN+Post (Ours)	81.3	**71.4**	78.4	**80.3**	**84.9**	**51.8**	**51.2**	**52.9**

Table 1. Model comparison on two different test sets, the Beethoven Piano Sonatas (BPS), and the full test set. RN stands for Roman Numeral, RN_{alt} for the alternative Roman Numeral computations discussed in Section 3.1. $RN(Onset)$ refers to onset-wise prediction accuracy, all other scores use the CSR score (see Section 5). Note that model *CSM-T* reports *Mode* instead of *Quality*.

Piano Sonatas (MPS) dataset [28] for which we also applied the aforementioned augmentations.

4.2 Configuration

For all our experiments, we train our network with the AdamW optimizer. We fix our architecture with a hidden size of 256, a learning rate of 0.0015, a weight decay of 0.005, and a dropout of 0.5 which is applied to each learning block of our architecture.

5. RESULTS

As an evaluation metric, we use Chord Symbol Recall (CSR) [29] where for each piece, the proportion of time is collected during which the estimated label matches the ground truth label. We apply the CSR at the 32nd note granularity level, in accordance with [6, 7, 9].

5.1 Quantitative Results

In the first experiment, which compares our *ChordGNN* to existing state-of-the-art approaches, we evaluate the full dataset, but also the annotated Beethoven Piano Sonatas (BPS) [5] subset, which many previous approaches had also used. The results are shown in Table 1. We present the CSR scores (where they are applicable) for Local Key, Degree, Quality, Inversion, Root, conventional Roman Numeral, and Alternative Roman Numeral (see Section 3). Furthermore, we include the onset-wise accuracy score for our models' conventional Roman Numeral predictions.

On the BPS subset, we compare our model *ChordGNN* with the Micchi (2020) model [6], the *CSM-T* (2021) model [9] and the *AugmentedNet* 2021 model [7]. Our results on Roman Numeral prediction surpass all previous approaches. Note that the *AugmentedNet* model exhibits higher prediction scores on the individual Key, Degree, Quality, and Root tasks, which are used jointly for the prediction of the Roman numeral. These results indicate that our model obtains more meaningfully interrelated predictions, with respect to the Roman numeral prediction, resulting in a higher accuracy score.

Moreover, we compare *ChordGNN* to *AugmentedNet* on the full test dataset. Our model surpasses *AugmentedNet* with and without post-processing in all fields apart from local key prediction and quality. Our model obtains up to 11.6% improvement in conventional Roman Numeral prediction.

In both experiments, post-processing has been shown to improve both RN and RN_{alt}. However, *ChordGNN* without post-processing already surpasses the other models.

Variant	RN	RN_{alt}
ChordGNN (Baseline)	46.1 ± 0.003	47.8 ± 0.007
ChordGNN + WLoss	$\mathbf{48.9} \pm 0.001$	$\mathbf{50.4} \pm 0.010$
ChordGNN + Rotograd	45.5 ± 0.003	47.1 ± 0.005
ChordGNN + R-GradN	45.2 ± 0.006	46.7 ± 0.005
ChordGNN + NADE	48.2 ± 0.005	49.9 ± 0.005

Table 2. Configuration Study: Chord Symbol Recall on Roman Numeral analysis on the full test set. RN stands for Roman Numeral, RN_{alt} refers to the alternative Roman Numeral computations discussed in section 3.1. WLoss stands for the dynamically weighted loss described in Section 3, and R-GradN stands for Rotograd with Gradient Normalization. Every experiment is repeated 5 times with the same ChordGNN model as Table 1 without post-processing.

5.2 Configuration Study

For a systematic study of multitask training, we investigated the effects of extension modules, gradient normalization techniques, and learnable weight loss. In detail, we test 5 configurations using as baseline the *ChordGNN* model (without post-processing) with standard CE loss and no weighing. Furthermore, we test our proposed architecture using the dynamically weighted loss described in Section 3.3 (same as the model in Table 1), Rotograd [17] and GradNorm [16] for Gradient Normalization, and NADE [8]. The models are run on the Full data set described above and averaged over five runs with random initialization. The results, summarized in Table 2, suggest that using the dynamically weighted loss yields better results compared to other methods such as the Baseline or Gradient Normalization techniques. Furthermore, the dynamically weighted loss is comparable to NADE but also more robust on Conventional Roman Numeral prediction on our datasets.

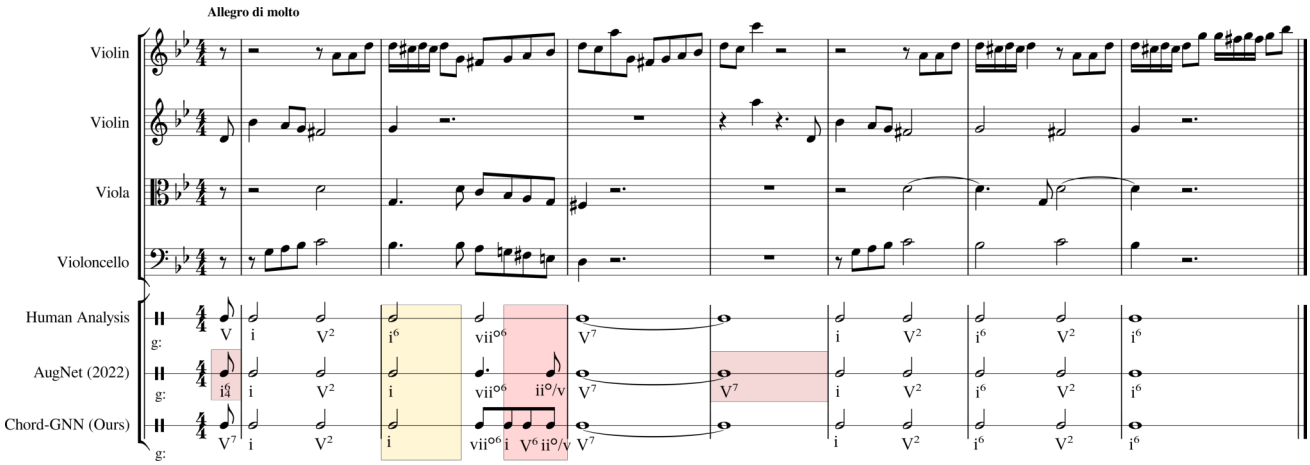


Figure 5. A comparison between the human annotation, AugmentedNet, and ChordGNN on a passage of Haydn's string quartet op.20 No.3 movement 4. The red (wrong) markings on Human Analysis and AugNet (2022) are from [21]

5.3 Latest developments

Our last experiment focuses on specific developments that have very recently been published in Nápoles López's Ph.D. thesis [21]. In the thesis, three additional tasks, related to predicting the components of a canonical representation of the current chord, as implied by the Roman Numeral, were proposed and the dataset was extended with the Annotated Mozart Piano Sonatas (MPS) corpus [28], as mentioned in Section 4.1 above.

To test the relevance of these updates, we trained an adapted version of our model, now with 11+3=14 individual tasks and including the Mozart data. It turns out that the updated model improves significantly in performance, achieving a 53.5 CSR score on conventional Roman Numeral (compare this to row "ChordGNN (Ours)" in Table 1). Furthermore, post-processing can improve the results by up to two additional percentage points. [1]

5.4 A Musical Example

In Figure 5, we look at a comparison between the human annotations, *AugmentedNet* and *Chord-GNN* predictions (The musical excerpt is taken from Nápoles López's thesis [21], and the predictions relate to the new models trained as described in the previous section.). Marked in red are false predictions, and marked in yellow are correct predictions of the model with wrong ground-truth annotations. Both models' predictions are very similar to the human analysis. However, our model correctly predicts the initial pickup measure annotation. In measure 2, the ground truth annotation marks a tonic in first inversion; however, the viola at that point is lower than the cello and therefore the chord is actually in root position. Both models obtain a correct prediction at that point. Subsequently, our model predicts a harmonic rhythm of eighth notes, which disagrees with the annotator's half-note marking. Analyzing the underlying harmony in that passage, we can justify our model's choices.

[1] Unfortunately, we cannot directly compare these numbers to [21], as their results are not reported in comparable terms.

The human annotation suggests that the entire second half of the 2nd measure represents a vii^o chord. However, it should not be in the first inversion, as the cello plays an F# as the lowest note (which is the root of vii^o). The AugNet analysis faces the same issue, in contrast with the predictions of ChordGNN. However, there are two conflicting interpretations of the segment. First, the vii^o on the third beat is seen as a passing chord between the surrounding tonic chords, leading to a dominant chord in the next measure. Alternatively, the vii^o could already be part of a prolonged dominant harmony (with passing chords on the offbeats) leading to the V^7. The ChordGNN solution accommodates both interpretations as it doesn't attempt to group chords at a higher level, treating each eighth note as an individual chord rather than a passing event. The other two solutions prefer the second option.

6. CONCLUSION

In this paper, we presented *ChordGNN*, a model for automatic Roman Numeral analysis in symbolic music, based on a note-level, graph-based score representation. We showed that *ChordGNN* improves on other state-of-the-art models, and that post-processing can further improve the accuracy of the predictions. A configuration study suggests that gradient normalization techniques or techniques for carrying prediction information across tasks are not particularly beneficial or necessary for such a model.

Follow-up work will focus on strengthening the robustness of our models by pre-training with self-supervised methods on large corpora. We believe that such pre-training can be beneficial for learning helpful intrinsic musical information. Such a step is crucial since more data improves predictions but Roman Numeral annotations are hard to find or produce. Moreover, we aim to enrich the number of tasks for joint prediction by including higher-level analytical targets such as cadence detection and phrase boundary detection. Finally, we aim to extend our method to the audio domain.

7. ACKNOWLEDGEMENTS

We gratefully acknowledge the musical analysis of the vii^o passage in Fig. 5 (Section 5.4) that was offered by an anonymous reviewer, and which we took the liberty of adopting for our text. 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).

8. REFERENCES

[1] J. Pauwels, K. O'Hanlon, E. Gómez, M. Sandler *et al.*, "20 years of Automatic Chord Recognition from Audio," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2019.

[2] D. Temperley, *The cognition of basic musical structures*. MIT press, 2004.

[3] C. Raphael and J. Stoddard, "Functional Harmonic Analysis Using Probabilistic Models," *Computer Music Journal*, vol. 28, no. 3, pp. 45–52, 2004.

[4] J. P. Magalhaes and W. B. de Haas, "Functional Modelling of Musical Harmony: an experience report," *ACM SIGPLAN Notices*, vol. 46, no. 9, pp. 156–162, 2011.

[5] T.-P. Chen, L. Su *et al.*, "Functional Harmony Recognition of Symbolic Music Data with Multi-task Recurrent Neural Networks." in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2018.

[6] 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 (TISMIR)*, vol. 3, no. 1, pp. 42–54, 2020.

[7] N. Nápoles 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.

[8] 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 (ISMIR)*, 2021.

[9] A. P. McLeod and M. A. Rohrmeier, "A modular system for the harmonic analysis of musical scores using a large vocabulary," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2021.

[10] D. Jeong, T. Kwon, Y. Kim, and J. Nam, "Graph Neural Network for Music Score Data and Modeling Expressive Piano Performance," in *Proceedings of International Conference on Machine Learning (ICML)*, 2019.

[11] E. Karystinaios and G. Widmer, "Cadence Detection in Symbolic Classical Music using Graph Neural Networks," *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2022.

[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 *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2023.

[13] C. Hernandez-Olivan, S. R. Llamas, and J. R. Beltran, "Symbolic Music Structure Analysis with Graph Representations and Changepoint Detection Methods," 2023.

[14] Z. Zhang, P. Luo, C. C. Loy, and X. Tang, "Facial Landmark Detection by Deep Multi-task Learning," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014.

[15] M. Guo, A. Haque, D.-A. Huang, S. Yeung, and L. Fei-Fei, "Dynamic Task Prioritization for Multitask Learning," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[16] Z. Chen, V. Badrinarayanan, C.-Y. Lee, and A. Rabinovich, "GradNorm: Gradient Normalization for Adaptive Loss Balancing in Deep Multitask Networks," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2018.

[17] A. Javaloy and I. Valera, "RotoGrad: Gradient Homogenization in Multitask Learning," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022.

[18] A. Navon, A. Shamsian, I. Achituve, H. Maron, K. Kawaguchi, G. Chechik, and E. Fetaya, "Multi-task Learning as a Bargaining Game," *Proceedings of the International Conference on Machine Learning (ICML)*, 2022.

[19] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *Advances in neural information processing systems*, vol. 30, 2017.

[20] L. Liebel and M. Körner, "Auxiliary tasks in multi-task learning," *arXiv preprint arXiv:1805.06334*, 2018.

[21] N. Nápoles López, "Automatic roman numeral analysis in symbolic music representations," Ph.D. dissertation, Schulich School of Music McGill University, December 2022.

[22] 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.

[23] N. Nápoles López, "Automatic Harmonic Analysis of Classical String Quartets from Symbolic Score," Ph.D. dissertation, Master's thesis, Universitat Pompeu Fabra, 2017.

[24] 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 (ISMIR)*, 2015.

[25] M. Gotham, D. Tymoczko, and M. S. Cuthbert, "The RomanText Format: A Flexible and Standard Method for Representing Roman Numeral Analyses." in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2019.

[26] M. R. H. Gotham and P. Jonas, "The Openscore Lieder Corpus," in *Proceedings of the Music Encoding Conference (MEC)*, 2021.

[27] N. Nápoles López and I. Fujinaga, "Harmonic Reductions as a Strategy for Creative Data Augmentation," in *Late-Breaking Demo at International Society for Music Information Retrieval Conference (ISMIR)*, 2020.

[28] J. Hentschel, M. Neuwirth, and M. Rohrmeier, "The Annotated Mozart Sonatas: Score, Harmony, and Cadence," *Transactions of the International Society for Music Information Retrieval (TISMIR)*, vol. 4, no. ARTICLE, pp. 67–80, 2021.

[29] C. Harte, "Towards automatic extraction of harmony information from music signals," Ph.D. dissertation, Queen Mary University of London, 2010.

SEMI-AUTOMATED MUSIC CATALOG CURATION USING AUDIO AND METADATA

Brian Regan
Spotify
brianr@spotify.com

Desislava Hristova
Spotify
desih@spotify.com

Mariano Beguerisse-Díaz
Spotify
marianob@spotify.com

ABSTRACT

We present a system to assist Subject Matter Experts (SMEs) to curate large online music catalogs. The system detects releases that are incorrectly attributed to an artist discography (misattribution), when the discography of a single artist is incorrectly separated (duplication), and predicts suitable relocations of misattributed releases. We use historical discography corrections to train and evaluate our system's component models. These models combine vector representations of audio with metadata-based features, which outperform models based on audio or metadata alone. We conduct three experiments with SMEs in which our system detects misattribution in artist discographies with precision greater than 77%, duplication with precision greater than 71%, and by combining the approaches, predicts a correct relocation for misattributed releases with precision up to 45%. These results demonstrate the potential of such proactive curation systems in saving valuable human time and effort by directing attention where it is most needed.

1. INTRODUCTION

Online music catalogs such as Spotify's contain millions of releases, and new ones are added daily by providers ranging from professionally-staffed music labels to DIY artists via aggregators. In such large catalogs, it is common that multiple artists share the same or similar names, or that content by one artist comes from different providers. For example, there are 14 distinct metal bands with the name *Burial* [1] . When a new release by a *Burial* makes it to the catalog, in the absence of a unique artist identifier, we must make a decision of where to place the content: Is it by the Italian doom metal band, the English death metal band, one of the other 12 bands named *Burial*, or an entirely new one? In general, *to which artist do we attribute a release when there are multiple artists with the same name?*

Music streaming services have multiple systems to ensure that releases are correctly placed on artist discographies. However, given the large volumes of content and the diversity of sources, it is inevitable that on rare occasions a release is incorrectly attributed (e.g. due to incomplete or incorrect metadata, extreme ambiguity, or human error). These errors can manifest in two different ways: 1) *Misattribution*: when a release is incorrectly attributed to an artist, so that their discography now contains releases from two separate real-world artists; 2) *Duplication*: when a release is not attributed to the correct existing discography but to a new one, so that a single artist's work is split across the two discographies. These errors negatively impact the experience of both artists and users on the platform.

The problem of Named Entity Disambiguation (NED) has been extensively researched to attribute scientific papers to homonym authors using metadata such as the author's fields of research, academic affiliations, and co-authors [1–3]. In Music Information Retrieval (MIR), NED is primarily tackled as artist identification or multi-class classification with known artist classes. Approaches to this problem rely primarily on audio feature representations [4–6]. These methods cannot be applied to catalogs with a large or unknown number of artists, and do not take advantage of all existing information.

Here we present a semi-automated proactive curation system to detect and correct attribution errors across large music catalogs. The system consists of two machine learning sub-systems: a system for detecting misattribution by splitting discographies with releases from multiple real-world artists into their constituent sub-discographies (Fig. 1a), and a deduplication system that takes pairs of discographies or sub-discographies and decides if they should be combined (Fig. 1b). Both sub-systems rely on metadata and the acoustic similarity between releases, using deep convolutional network embeddings of their mel-spectrograms [7]. We show that combining audio and metadata features improves average precision in misattribution and duplicate detection by 10% and 6% respectively.

"In the wild" experiments with music catalog curation Subject Matter Experts (SMEs) show that our system achieves over 77% precision on misattribution detection, over 71% precision on duplicate detection, and 45% precision on finding the correct relocation of misattributed releases. Together these results demonstrate the power of proactive catalog correction systems in assisting human-led curation efforts.

[1] https://www.metal-archives.com/bands/Burial

© B. Regan, D. Hristova, and M. Beguerisse-Díaz. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** B. Regan, D. Hristova, and M. Beguerisse-Díaz, "Semi-Automated Music Catalog Curation Using Audio and Metadata", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

2. RELATED WORK

Recent advances in audio feature representation using deep learning [8] have applications to recommendations [7], audio classification [4] and artist identification [4, 6, 9, 10]. These works typically focus on the audio and do not include additional information (the method in [9] uses genre in its negative sampling method, but the model takes only audio). Work in other Named Entity Disambiguation (NED) applications shows that combining learned feature representations and manually crafted diverse features outperforms using either in isolation [11, 12]. This suggests that combining multiple data types (e.g. content and metadata) can improve the performance of music NED systems.

Duplicate entity detection (also known as entity matching or entity resolution) across or within databases typically has a *blocking* step [13] optimised for recall to reduce the set of pairwise comparisons, followed by an *entity matching* step optimised for precision. If labelled pairs of entities are available, supervised machine learning approaches can be used for matching. These are typically based on various string-based similarity features, such as entity name similarity [14].

Although state-of-the-art NED research focuses on automation [1], a human-in-the-loop (HITL) paradigm is commonly used in practice. A HITL approach is useful for resolving highly ambiguous cases and correcting automated decisions. In [3] the authors describe a machine learning approach that optimises human effort spent on labelling for author disambiguation. In the Microsoft Academic Graph [2], the author disambiguation system uses crowdsourced data as supervision signals.

Crowdsourced and authoritative sources such as MusicBrainz [15], VIAF [16], Wikidata [17], or ISNI [18] are useful for artist name disambiguation, but their benefit is limited for artists in the long tail or for brand new releases without unique artist identifiers.

3. METHODOLOGY

Our system operates on music releases (i.e. albums) denoted as a, and on artist credits in them. The set of releases credited to an artist forms the artist's discography: $\mathcal{A} = \{a_i, a_2, \dots\}$. The objective of our system is twofold:

Correct discographies Every release within a discography should credit the same real-world artist; i.e. there is no misattribution in the discography.

Complete discographies A real-world artist's releases should not be split across multiple discographies; i.e. there should be no more than one discography per artist.

Figure 1 illustrates our approach to achieve these goals; we achieve correctness and completeness by relocating misattributed releases and resolving (i.e. merging) duplicate discographies. Note that there are cases where a single real person performs under distinct artist identities (e.g. Dan Snaith performs as Caribou and Daphni). These discographies should not be considered duplicates. In addition, some releases can belong to multiple discographies if they credit multiple distinct artists (e.g. collaborations and remixes); however, a discography should always contain releases under a common artist.

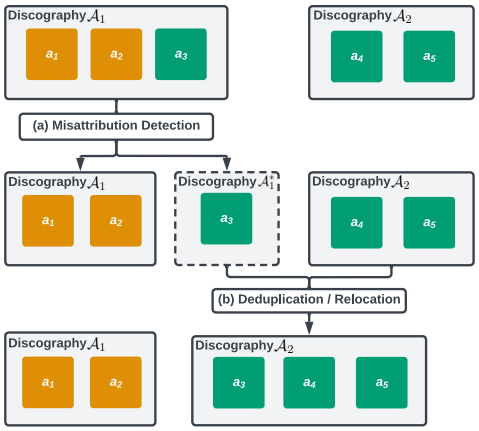


Figure 1: **System Overview:** (a) Misattribution detection is performed on each discography $\mathcal{A}$. The misattributed release a_3 is split out from $\mathcal{A}_1$ into *sub-discography* $\mathcal{A}_1^*$. (b) All (sub-)discographies are considered for deduplication; $\mathcal{A}_1^*$ is merged into $\mathcal{A}_2$, relocating the misattributed releases into the correct discography.

3.1 Misattribution Detection

The misattribution detection method, illustrated in Fig. 2, processes an artist's attributed discography $\mathcal{A}$ in two stages: First, we obtain a distance $\mathrm{dist}(a_i, a_j)$ between all pairs of releases $a_i, a_j \in \mathcal{A}$ using the combination of audio and metadata signals in Table 1. Second, we partition $\mathcal{A}$ using this distance by constructing a Minimum Spanning Tree (MST) [19] and imposing a threshold θ_{dist}. When we cut the MST edges where $\mathrm{dist}(a_i, a_j) > \theta_{\mathrm{dist}}$, the remaining connected components should contain releases from the same artist. These partitions are disjoint subsets: $\mathcal{A}_i \subseteq \mathcal{A}, i = 1 \dots m$, for which all releases belong to the same real-world artist. If the cardinality of the partition is $m > 1$, then there is at least one misattributed release in the discography (i.e. more than one artist's content is detected) and the discography should be split.

3.1.1 Pairwise Model

To obtain the pairwise distance between releases in a discography, we train a Random Forest ensemble classifier [20] $\mathrm{dist} : \mathcal{A} \times \mathcal{A} \rightarrow (0, 1]$, where high values indicate that the releases are likely to be from different artists.

Data. The training data consists of ∼45K release pairs from ∼28K artist discographies. This data, which we call the *Relocations* dataset, contains historical corrections of artist misattributions. The genres of the releases in this data are representative of Spotify's catalog. Each relocation is a move of an incorrectly-placed release from an artist's discography to the correct one. To construct the training

Figure 2: **Misattribution detection**: *(a):* An artist discography $\mathcal{A} = \{a_1, a_2, a_3\}$ in which release a_3 is misattributed. *(b):* The pairwise distance matrix D computed using our model. *(c):* A Minimum Spanning Tree (MST) is computed from the distances. *(d):* After applying a threshold θ_{dist} to the MST, the discography $\mathcal{A}$ is split into two partitions, which correspond to the two distinct real world artists present in the discography.

data, consider a release a_1^i that was moved from discography $\mathcal{A}_j$ to $\mathcal{A}_i$. We pair a_1^i with a release $a_1^j \in \mathcal{A}_j$ from the discography where it was incorrectly located: $(a_1^i,\, a_1^j)$, and give it the *"mismatched"* label. Then, we pair a_1^i with a release from the correct discography: $(a_1^i,\, a_2^i)$, $a_2^i \in \mathcal{A}_i$, and give it the *"not mismatched"* label.

Model Features. We use a combination of metadata and audio-based features, summarised in Table 1. Audio features include deep acoustic embeddings from a proprietary model trained in a fashion similar to [7], originally developed for music recommendations, and *speechiness* - a probability that a track contains spoken word as determined by another proprietary model [21]. An advantage of audio features is that they are available for every release. In general, we expect releases from the same artist to sound similar to each other. As mentioned in Sec. 2, previous works report good performance using audio-based methods alone [4,9,10]. However, releases from different artists can also sound similar (e.g. if they come from the same genre), and releases from the same artist can be musically different (e.g. an artist whose style evolved or spans many genres).

On the other hand, metadata features such as music labels, composers or lyricists can have high precision (e.g. releases from the same discography delivered by the same label are likely to be by the same artist), but in isolation metadata matches can be sparse, or have mistakes. Therefore, we supplement audio similarity with metadata based features to improve the performance of our classifier.

3.1.2 *Grouping releases in a discography*

Our distance allows comparisons between individual pairs of releases to decide whether they belong to distinct artists. For example, if $\text{dist}(a_i,\, a_j) > \theta_{\text{dist}}$ for a given $\theta_{\text{dist}} \in (0, 1]$, we could say that it is unlikely that the releases share an artist. However, this comparison ignores the context of the whole discography $\mathcal{A}$, and may fail when the sound of an artist has evolved in time, the artists changed collaborators or labels throughout their career. To mitigate these issues, we consider each comparison in the context of all the releases in $\mathcal{A}$.

Attribute	Functions
Music Label*	Exact Match*, Dice Score [2]
Music Licensor*	Exact Match
Music Source*	Exact Match
Release Name	Exact Match, Dice Score
Release Group* [3]	Exact Match
Release Artists	Overlap, Dice Overlap [4]
Release Track Names*	At Least 1 Exact Match, Min Dice Score
Release Track Artists	Max Overlap, Max Dice Overlap
Release Track Language*	At Least One Exact Match
Release Type†*	Categorical
Release Is Remix†	Categorical
At Least One Track Is Remix†*	Categorical
Track Audio Vectors*	Min/Max/Mean Cosine Similarity
Track Speechiness†	Min/Max/Mean

Table 1: **Pairwise Model Inputs**. The features above the line are metadata, and below are audio-based. Features with * were included in the model for the SME experiment. Track level attributes are aggregated to release level with the functions described. Attributes with † produce two features, one for each release. Random permutations of underlined feature values decreased test-set performance >95% of the time.

We construct the matrix $D \in \mathbb{R}^{m \times m}$ where $D_{ij} = \text{dist}(a_i,\, a_j)$, and use it to obtain a MST, which is a graph with node set $\mathcal{A}$, and edges with weight equal to the nodes' pairwise distance (see Fig. 2c). The MST connects releases that are "close" to each other, and provides a global summary of how the releases are organised in a latent space, while capturing the continuity of the data arising from evolution in the style and career of an artist. We can attribute two dissimilar releases to the same artist if there is a path of short hops along the MST that connects them. Put another way, if we cut very long hops (i.e. long edges) in the MST, we get connected components in which we can only go between nodes by a series of short hops. Our hypothesis is that these components (partitions of $\mathcal{A}$) are releases that

[2] The Dice score is the average of the Dice coefficient [22] for n-gram values of 1,2,3 and 4.

[3] Indicates whether the pair of releases have been identified by other systems as duplicates

[4] Number of pairs of artists with Dice score > 0.7

are likely to be from the same artist. Specifically, we need to find a threshold θ_{dist} and cut all edges in the MST that are larger. The remaining connected components preserve transitive relations even when the distance is not transitive: if $\text{dist}(a_i,\, a_j)$ is low and $\text{dist}(a_j,\, a_k)$ is low, $\text{dist}(a_i,\, a_k)$ can still be high, but one can traverse from a_i to a_k with short hops via a_j. This approach preserves the diversity of releases over the careers of artists. If no edge is larger than θ_{dist}, then the MST connects all releases with paths of short hops, and we assume that they are all correctly attributed to the same artist.

3.2 Discography Deduplication

The goal of deduplication is to merge existing discographies, or sub-discographies split out from misattribution detection, that belong to the same artist (e.g. release a_3 in Fig. 1). Deduplication consists of two steps: (1) generating candidates for deduplication through a blocking strategy, and (2) a prediction step that determines whether the pair of discographies belong to the same real-world artist.

3.2.1 Blocking

To reduce the comparisons between pairs of discographies while maintaining high recall, we want to create small *blocks* of discographies that could belong to the same artist. One way is to simply take homonym artist discographies as a block; however, errors which lead to misattribution and duplication in music catalogs are often associated with varied spellings or aliases of the same real-world artist. Therefore, we need a more robust blocking strategy.

We build an Elasticsearch [23] index of all artist names in the catalog which we use to match and rank deduplication candidates. The matching strategy combines three conditions: (1) n-grams with $n = 2, 3, 4$; (2) fuzzy string matching with edit distance $\leq$ 2; and (3) normalised string matching without spaces and stop-words. If one or more of these conditions match a *seed* discography artist name, Elasticsearch returns a list of all matching candidates ranked by their *elastic score* [24]. We evaluate this strategy on a dataset of source and target artist name pairs from the *Merges* dataset (described below), and obtain a recall@10 of 97%.

3.2.2 Duplicate detection model

We train a Random Forest classifier to compute the similarity $\text{sim}(\mathcal{A}_i, \mathcal{A}_j) \in (0,\, 1]$ between pairs of artist discographies within each block. A high similarity score means that the two discographies are likely to come from the same real-world artist and should be merged, while a low score indicates that they are from different artists and should remain separate.

Data. The training data consists of ∼224K discography pairs. This data, which we call the *Merges* dataset, contains historical corrections of duplicate artist discographies. We assign a positive label to each merged pair and generate up to 10 negative examples for each positive one using the blocking strategy. During training we balance the data by applying a weight to each sample to be inversely proportional to its class frequency.

Attribute	**Functions**
Elasticsearch relevance score	See [24]
Artist name similarity	2-gram Dice coefficient
Release Names	Jaccard similarity
Release Track Names	Jaccard similarity
Release Artists	Overlap between artist names of collaborators on releases
Release Track Artists	Overlap between artist names of collaborators on release tracks
Number of releases	$\lvert\mathcal{A}_i \bigcup \mathcal{A}_j\rvert$
Track Audio Vectors	Mean Cosine Similarity

Table 2: **Duplicate Discography Detection Model Inputs.** Features above the line are metadata, and below are audio-based. Random permutations of underlined feature values decreased test-set performance >95% of the time.

Model Features. As in the misattribution model, we combine engineered metadata features with acoustic embeddings (see Table 2). Duplicate entity detection systems typically rely heavily on string similarity, but there are some challenges. For example, consider merging the discography referencing the artist *Prince*, with the one referencing his alias *Prince of Funk*, while remaining distinct from another artist called *Princess*. Relying solely on string similarity would suggest that the discographies from *Prince* and *Princess* are more likely to belong to the same artist than the ones from *Prince* and *Prince of Funk*. In this scenario, including audio representations in the model can improve performance in the absence of other distinctive features.

4. EVALUATION

We evaluate our system's performance with a series of experiments: First, we examine the offline performance of each sub-system under different feature ablations, including audio and metadata signals alone, using the *Relocations* and *Merges* datasets. Second, we conduct three experiments with Subject Matter Experts (SMEs) showing the performance "in the wild" of the misattribution and deduplication models, and their unification for the relocation of misattributed releases, as described in Fig. 1.

4.1 Audio and Metadata Feature Ablations

We test the hypothesis that metadata and learned audio representations model catalog correction tasks (i.e. misattribution and duplicate detection) better together than separately. Figure 3 shows the performance of the two models in three configurations: audio features only, metadata features only and combined. The features for each model and the distinction between audio-based and metadata-based features can be found in Tables 1 and 2. For each set of features, we separately tuned the hyperparameters with 5-fold cross-validation.

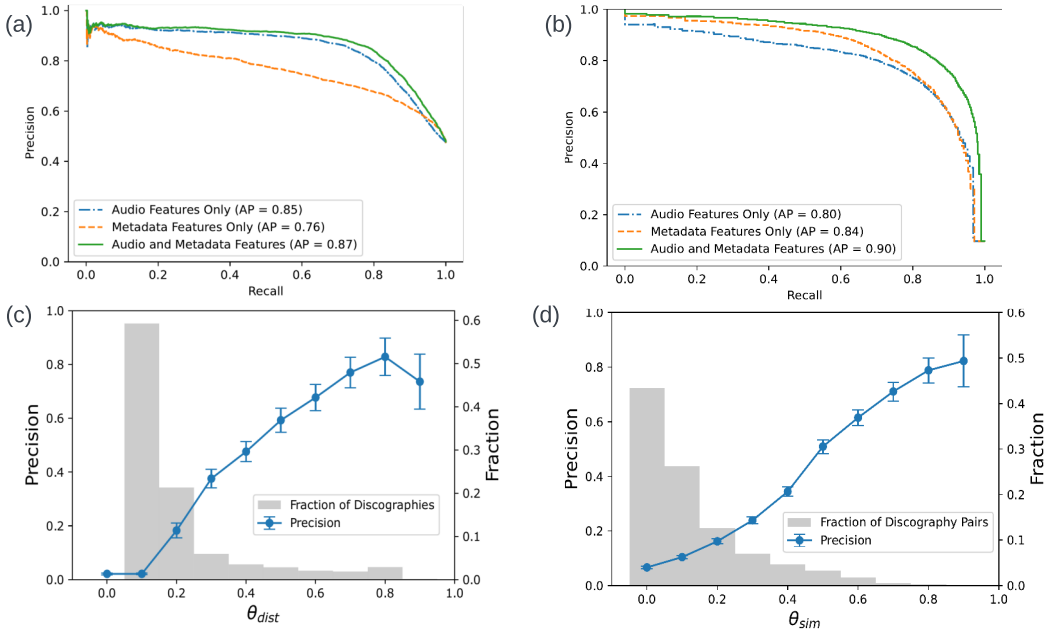


Figure 3: *(a) - (b)*: Precision-Recall curves in offline experiments with combinations of audio and metadata features for misattribution detection *(a)* and deduplication *(b)*. Average precision (AP) is reported in the legend for each set of features. *(c) - (d)*: Annotation experiment results for misattribution detection *(c)* and deduplication *(d)*. Precision is calculated for each threshold bucket and reweighed by the distribution of predictions shown on the second y axis.

Figure 3a shows that the pairwise misattribution model using audio-based features alone has good performance, but combining both audio and metadata produces the best performance. The full model has an average precision (AP) increase of 10.69% over the metadata-only model, and 1.95% AP over the audio-based model. These improvements come from a reduction in false positives (e.g. when the sound is not similar, but metadata similarities exist between two releases). For example, the test data contains the releases *SHOOT MY SHOT* and *Hurts Like Hell (feat. Offset)* from the American rapper *Offset*. The audio-only model predicts these releases come from different artists (their distance is 0.77). The full model gives the pair a distance of 0.1 because *"Kiari Kendrell Cephus"* (which is Offset's real name), appears in the credits of both releases as a writer and a composer/lyricist.

Figure 3b shows the performance of the duplicate detection task under the different ablations. Using metadata features alone outperforms audio features alone by 4% in AP. This is not surprising, as entity resolution tasks are usually heavily based on string similarity across aligned fields. Here too we can achieve good performance with metadata based features alone, but combining the features boosts AP by 6%. This boost is driven by cases where metadata features are insufficient. In the example of the *Prince* and *Prince of Funk* discographies, in the absence of shared collaborators or similarity on release titles we would get a false negative. However, the acoustic similarity between the two discographies is high, which allows us to correctly identify them as by the same real-world artist.

4.2 Experiments with SMEs

We conducted three experiments with SMEs to understand the performance of each task independently, and of the entire correction system (Fig. 1) in the context of its intended use, for a range of decision thresholds. We use precision as our evaluation metric since we want to reduce human effort spent reviewing and correcting the catalog.

4.2.1 Misattribution Detection

We ran the misattribution detection method from Sec. 3.1 using an early version of the pairwise model that was ready when the SMEs were available. The difference between the full model and this early version is that the latter uses only subset of the features of the full model (marked with * in Table 1). We selected a subset of artist discographies from the Spotify catalog, biased toward more popular artists, that reviewers are able to cross-reference externally. Then, we randomly sampled a pair of releases from each artist and calculated the value of the threshold θ_{dist} that would split the pair into two different partitions of the discography. This value is the largest edge weight along the path connecting the releases in the MST of the artist's discography. In the example in Fig 2c, the threshold between releases a_1 and a_3 would be $\theta_{\text{dist}} = 0.85$. We stratified our sample by these bucketed threshold values in 10 equally sized bins between 0 and 1, with a maximum of 100 pairs per bucket. The sampling produces ∼1K pairs, each of which was reviewed by a SME who classified it as *"by the same artist"* or *"by different artists"*. Figure 3c shows the precision for each value of θ_{dist} (blue line, left y-axis). For example, at a $\theta_{\text{dist}} > 0.7$, we can achieve 77%

precision. When θ_{dist} is small, many single-artist discographies are split into more than one group. This lowers precision but increases the fraction of artists that would have their discography partitioned into more than one group at each threshold (grey bars, right y-axis).

4.2.2 Duplicate Detection

To evaluate the duplicate detection model from Sec. 3.2, we generated a list of 140K seed artist discographies of popular artists from the catalog. Then, we generated 10 candidates for each seed artist using our blocking strategy to form artist-candidate pairs. For each pair we compute $\text{sim}(a_i, a_j)$, and bucket the scores in the same way as for the misattribution detection task above, sampling up to 100 per bin. For this task, 3 SMEs reviewed each sample and answered the question: *Do the two discographies belong to the same real-world artist?* We aggregated the annotations per sample to reflect the majority vote (i.e. at least 2 out of 3 of the annotators agree) and got 94% agreement. The remaining 6% of cases are ambiguous, and were excluded from the analysis. These cases are interesting and give insight into edge cases for future iterations of the model. For example, when the discographies were related but not technically by the same artist, e.g. the *Thelonious Monk Quintet* and the *Thelonious Monk Quartet*.

As in the misattribution task, as the threshold θ_{sim} in Fig. 3d increases so does precision, but with fewer candidate pairs (shown as grey bars, right y-axis). At a $\theta_{sim} >$ 0.7, we achieve 71% precision.

4.2.3 Predicted Relocation

Discography pairs that have been reviewed and determined to be duplicates can be merged in the catalog in a straightforward way. However, correcting misattributions is not so easy, and we still need to identify the correct discography in which they belong. Having validated both steps in our discography correction system, we can use the duplicate detection method to predict the correct discography (if any) for misattributed releases. To do this, we identify misattributions, using $\theta_{\text{dist}} > 0.7$ based on the previous experiments, and we treat the misattributed releases as a sub-discography. Then, we generate and score candidate duplicate discographies for these sub-discographies using the deduplication model.

We evaluate performance on $\sim$1K release-discography pairs. Since the model generates up to ten predictions per seed, we take the highest predicted placement as a candidate for annotation. We asked SMEs to review the release and its predicted relocation and answer the question: *Does the release belong with the discography?*

Figure 4 shows the precision as a function of the two steps in the correction system θ_{dist} and θ_{sim}. The highest precision is 45%, which is achieved when both the misattribution step and deduplication (relocation) step have a high θ (top right corner of Fig. 4, representing 17% of the sample). The relocation task is more difficult and less precise because it inherits the uncertainty and performance of misattribution and duplicate detection. Additionally, we expect that a large number of misattributed releases might not belong anywhere, and will become standalone discographies. This means that even if the system considered this relocation to be the best out of ten candidates, a relocation might not be possible at all. Even in this scenario, the human effort to detect and correct misattributed content is significantly reduced.

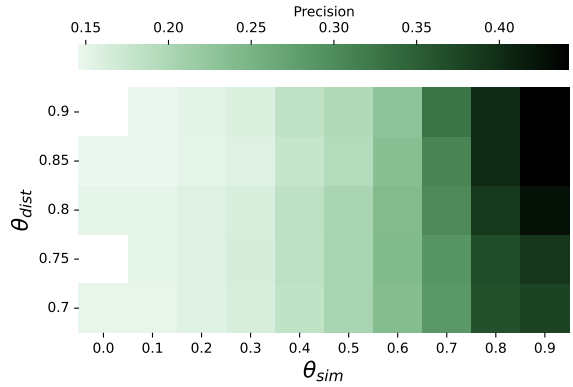


Figure 4: Precision of the combined system on the task of predicted relocation of misattributed releases for varying thresholds of the misattribution (θ_{dist}) and duplicate detection (θ_{sim}) methods.

5. DISCUSSION

We present a system designed for SMEs to maintain the correctness and completeness of artist discographies in a large online catalog. We demonstrate that leveraging both audio and metadata-based signals for misattribution detection and deduplication of discographies outperforms either in isolation. We validated each task separately, and the entire correction system across different thresholds, showing strong performance in three experiments with SMEs.

The power of this system is that it can scan a large catalog efficiently and direct the attention of human reviewers to where errors are most likely to be found, as well as suggest corrections for cases of misattribution and deduplication. This makes our system a key part of proactive catalog curation strategies. It is possible that some curation steps could be automated for high confidence predictions; however, due to the downstream impact of curation decisions (e.g. recommendations, search, user experience) the tolerance for incorrect relocations is low.

The current implementation of this system runs weekly, and the top-scoring candidates for misattibution, deduplication and predicted relocations are flagged for SMEs review. These reviews, in turn, become new labelled data on which the model can be re-trained and further improved.

Although discography errors are rare, it is important to minimise them as much as possible. Systems such as this are one tool among many that streaming platforms can use to ensure their catalog is correct, and to safeguard the experience of users and artists.

6. ACKNOWLEDGMENTS

We thank Nicola Montecchio and Glenn McDonald for discussions around the original idea for this work, Dimitrios Korkinof for valuable consultations throughout this project, Dana Puleo for assistance running experiments with SMEs, and Antonio Lima for comments on the manuscript.

7. REFERENCES

[1] A. A. Ferreira, M. A. Gonçalves, and A. H. Laender, "A brief survey of automatic methods for author name disambiguation," *Acm Sigmod Record*, vol. 41, no. 2, pp. 15–26, 2012.

[2] K. Wang, Z. Shen, C. Huang, C.-H. Wu, Y. Dong, and A. Kanakia, "Microsoft academic graph: When experts are not enough," *Quantitative Science Studies*, vol. 1, no. 1, pp. 396–413, 2020.

[3] Y. Qian, Y. Hu, J. Cui, Q. Zheng, and Z. Nie, "Combining machine learning and human judgment in author disambiguation," in *Proceedings of the 20th ACM international conference on Information and knowledge management*, 2011, pp. 1241–1246.

[4] H. Lee, P. Pham, Y. Largman, and A. Ng, "Unsupervised feature learning for audio classification using convolutional deep belief networks," *Advances in neural information processing systems*, vol. 22, 2009.

[5] A. Brinkman, D. Shanahan, and C. Sapp, "Musical stylometry, machine learning and attribution studies: A semi-supervised approach to the works of josquin," in *Proceedings of the Biennial International Conference on Music Perception and Cognition*, 2016, pp. 91–97.

[6] Z. Nasrullah and Y. Zhao, "Music artist classification with convolutional recurrent neural networks," in *International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2019, pp. 1–8.

[7] A. Van den Oord, S. Dieleman, and B. Schrauwen, "Deep content-based music recommendation," *Advances in neural information processing systems*, vol. 26, 2013.

[8] E. J. Humphrey, J. P. Bello, and Y. LeCun, "Moving beyond feature design: Deep architectures and automatic feature learning in music informatics." in *International Society for Music Information Retrieval Conference*. Citeseer, 2012, pp. 403–408.

[9] J. Royo-Letelier, R. Hennequin, V.-A. Tran, and M. Moussallam, "Disambiguating music artists at scale with audio metric learning," *International Society for Music Information Retrieval Conference*, 2018.

[10] J. Park, J. Lee, J. Park, J.-W. Ha, and J. Nam, "Representation learning of music using artist labels," *International Society for Music Information Retrieval Conference*, 2018.

[11] K. Kim, S. Rohatgi, and C. L. Giles, "Hybrid deep pairwise classification for author name disambiguation," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp. 2369–2372.

[12] Q. Zhou, W. Chen, W. Wang, J. Xu, and L. Zhao, "Multiple features driven author name disambiguation," in *2021 IEEE International Conference on Web Services (ICWS)*. IEEE, 2021, pp. 506–515.

[13] G. Papadakis, D. Skoutas, E. Thanos, and T. Palpanas, "A survey of blocking and filtering techniques for entity resolution," *arXiv preprint arXiv:1905.06167*, 2019.

[14] P. V. Konda, *Magellan: Toward building entity matching management systems*. The University of Wisconsin-Madison, 2018.

[15] "Musicbrainz," musicbrainz.org, accessed: 2023-04-14.

[16] "Virtual international authority file," viaf.org, accessed: 2023-07-03.

[17] D. Vrandečić and M. Krötzsch, "Wikidata: a free collaborative knowledgebase," *Communications of the ACM*, vol. 57, no. 10, pp. 78–85, 2014.

[18] "International standard name identifier," isni.org, accessed: 2023-07-03.

[19] J. B. Kruskal, "On the shortest spanning subtree of a graph and the traveling salesman problem," *Proceedings of the American Mathematical society*, vol. 7, no. 1, pp. 48–50, 1956.

[20] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[21] "Spotify for developers: Get track's audio features," developer.spotify.com/documentation/web-api/reference/get-audio-features, accessed: 2023-07-03.

[22] G. Kondrak, "N-gram similarity and distance," in *International symposium on string processing and information retrieval*. Springer, 2005, pp. 115–126.

[23] "Elasticsearch," https://www.elastic.co/, accessed: 2023-07-06.

[24] "Lucene's practical scoring function," www.elastic.co/guide/en/elasticsearch/guide/current/practical-scoring-function.html, accessed: 2023-04-14.

CROWD'S PERFORMANCE ON TEMPORAL ACTIVITY DETECTION OF MUSICAL INSTRUMENTS IN POLYPHONIC MUSIC

Ioannis Petros Samiotis
Delft University of Technology
i.p.samiotis@tudelft.nl

Christoph Lofi
Delft University of Technology
c.lofi@tudelft.nl

Alessandro Bozzon
Delft University of Technology
a.bozzon@tudelft.nl

ABSTRACT

Musical instrument recognition enables applications such as instrument-based music search and audio manipulation, which are highly sought-after processes in everyday music consumption and production. Despite continuous progresses, advances in automatic musical instrument recognition is hindered by the lack of large, diverse and publicly available annotated datasets. As studies have shown, there is potential to scale up music data annotation processes through crowdsourcing. However, it is still unclear the extent to which untrained crowdworkers can effectively detect when a musical instrument is active in an audio excerpt. In this study, we explore the performance of non-experts on online crowdsourcing platforms, to detect temporal activity of instruments on audio extracts of selected genres. We study the factors that can affect their performance, while we also analyse user characteristics that could predict their performance. Our results bring further insights into the general crowd's capabilities to detect instruments.

1. INTRODUCTION

Studies of the last decade have shown the success of data-driven algorithms to tackle complex classification tasks. Such algorithms require large annotated datasets to train and capture the nuances of multi-faceted problems, with crowdsourcing being successfully utilized to scale annotation processes to meet the ever higher demands [1–3]. While works such as [4] and [5] show that crowdsourcing can be a viable and powerful tool to distinguish and annotate music audio, it still remains underutilised as a tool in the domain, primarily due to the complexity of the annotation tasks [6] which are believed to demand extensive domain knowledge and training – arguably, musical elements such as tempo, chords and timbre can be demanding for an untrained human annotator to detect.

With this study, we aim at providing more evidence that complex music audio annotation tasks can be performed on crowdsourcing platforms. We focus on the task of musical instrument activity detection, and investigate non-experts' capability to recognise their activity and annotate the times in which they perform. Our study builds upon the findings of [4] where users were able to detect if an instrument was present in an audio excerpt or not. We extend this detection task to also cover the exact time-frames of instrument activity. This is a type of task where experts are commonly employed [7] to annotate data, due to several challenges such as multiple instruments playing simultaneously [8,9], or instruments of the same family exhibiting similar timbre [10, 11].

More specifically, we explore and analyse the capabilities of crowd workers to effectively detect temporal aspects of musical instrument activity in polyphonic audio (with focus on trio ensembles). We seek answer to the following questions:

- `RQ1`: To what extent non-experts can detect the onset and offset of a musical instrument's activity on polyphonic audio?
- `RQ2`: How their self-assessed perceptual abilities and musical knowledge relate to their performance?

Our study takes place on Prolific [1] . The audio excerpts were chosen from three different genres (namely *classical*, *jazz* and *rock*) to understand if different instruments and rhythms can affect the performance of crowd workers. We also utilize a set of pre-established and evaluated questionnaires to retrieve user attributes, that can potentially relate to their performance. We employ the "Musical Training" and "Perceptual Abilities" categories from Goldsmith's Music Sophistication Index (GMSI) [12], a questionnaire specifically designed to capture an individual's ability to engage with music. These specific categories were found previously to most significantly predict the workers' musical perceptual abilities [13].

Our results show that non-experts can demonstrate good perception of musical instruments' temporal activity for the chosen audio excerpts. Their self-assessed perceptual abilities reflect reasonably well their actual perception skill. These results open possibilities of further future studies on instrument activity annotation, and provide a positive outlook for systems relying on such annotations.

© I.P. Samiotis, C. Lofi, A. Bozzon. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** I.P. Samiotis, C. Lofi, A. Bozzon, " Crowd's Performance on Temporal Activity Detection of Musical Instruments in Polyphonic Music ", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

[1] `https://www.prolific.co`

2. RELATED WORK

The work in OpenMic 2018 [4] is one of the first attempts to annotate instrument presence for instrument recognition at scale, employing 2,500 unique annotators from CrowdFlower [2], using excerpts from Free Music Archive [3] and the AudioSet [14]. The researchers followed specific task design approaches to assist the crowd workers in their task, which they adapted after an initial study. The annotation process was limited to binary annotations, indicating the presence or absence of a musical instrument in an audio excerpt. Showcasing that crowd workers are able to provide strongly-labeled data, e.g. with temporal annotation, as in our study, can enable new opportunities for instrument activity detection and source separation.

Even though the study in [15] is not based on music audio, it demonstrates the crowd's ability to annotate temporal aspects of audio events. Our interface design is inspired by this study, as the crowd workers had to draw bounding boxes on spectrogram visualisations of audio excerpts. The sounds were synthesized using Scaper [16], for a greater control over *max-polyphony* and *gini-polyphony* (amount of sound overlap).

Our study is also motivated by recent findings regarding crowd workers music perception abilities [13]. Users of crowdsourcing platforms were shown to possess considerable skills to detect music aspects such as tempo and melody.

To the best of our knowledge, the current literature lacks works that study the performance of crowd workers on temporal activity detection of musical instruments in relationship with worker demographic or musical properties, which is the goal of this work.

3. EXPERIMENTAL DESIGN

We designed our experiment to study and understand if users on crowdsourcing platforms can perceive the temporal activity of a musical instrument in audio excerpts. We aim to focus on realistic use cases, thus testing the workers' capacity to perceive instruments in audio excerpts that are performed, recorded, mixed and mastered professionally. Therefore, we used existing recordings instead of synthesized audio which would have been less representative of real-life scenarios, but could have given us higher control on the musical aspects of the audio and instrumentation. To that end, we carefully selected the audio excerpts to control, as much as possible, musical aspects such as timbre and performance.

We employed previously established and evaluated questionnaires, to learn about workers' (a) "Perceptual Abilities" and "Music Training" through Goldsmith's Musical Sophistication Questionnaire (GMSI); (b) cognitive load through NASA's Task Load IndeX (NASA-TLX) survey [4]; (c) equipment quality [17] and (d) outside noise [18].

[2] https://visit.figure-eight.com/People-Powered-Data-Enrichment_T

[3] https://freemusicarchive.org

[4] https://humansystems.arc.nasa.gov/groups/tlx/

The task workflow started with simple demographic questions, followed by the GMSI questionnaire. The user was then introduced with the main task to annotate audio excerpts. The study concluded with a post-task survey regarding their cognitive load, equipment and a general feedback entry.

3.1 Selected Audio Excerpts

For the main annotation task, we made use of audio excerpts from trio ensembles of three major genres, *classical*, *jazz* and *rock*. We used audio excerpts of these particular three genres due to their wide discrepancy in instrumentation and rhythm. Even though in some occasions the instruments used in each genre can showcase timbre similarities (like double bass and bass guitar), in other cases the timbre can differ wildly (electric guitar compared to cello). To the best of our knowledge, there is no previous baseline of the crowd workers' perception of polyphonic music, so we decided to control for the maximum number of instruments that would play simultaneously in an excerpt, by selecting recordings of trio ensembles for each genre. Each audio excerpt had a length of 10 seconds, as used also in similar studies [4, 15]. The authors annotated the instrument activity per audio excerpt, which was later used to evaluate the crowd's annotations.

For the classical music excerpts, we made use of a specific type of a trio ensemble, namely *piano*, *clarinet* and *cello*. On the selected music clip, we selected an excerpt where both *clarinet* and *cello* have prominent parts, while *piano* is mostly following in the background. For our jazz excerpt, we used of the more standardized trio ensemble of *piano*, *double bass* and *drums*, where *double bass* and *drums* keep the rhythm and *piano* is performed in small melodic bursts. Lastly, for the category of rock, we made use of a music excerpt from "power trio" bands, which most frequently consist of *electric guitar*, *bass guitar* and *drums*. It follows the same performance pattern with the jazz excerpt on the *bass guitar* and *drums*, while the *electric guitar* enters near the middle of the excerpt with a sustained, distorted power chord.

We hypothesise that bass instruments will be more difficult to annotate in these genres, as bass-related sounds are more often "pushed back" during the mixing stage for such types of music. The different genres were selected to lessen the impact of possible enculturation bias. We believe that if only one genre was selected, participants who would be more familiar with it, would find it easier to spot the activity of instruments prominent in the genre. With the selected genres, we cover a variety of rhythms, instrumentations and performative aspects, which could impose a challenge to non-experts.

3.2 Task and Interface Design

To assess the music expertise of the crowd we employed parts of GMSI, namely: "Music Training" and "Perceptual Abilities". The choice of the categories was based on a study on music perception skills of crowd workers [13],

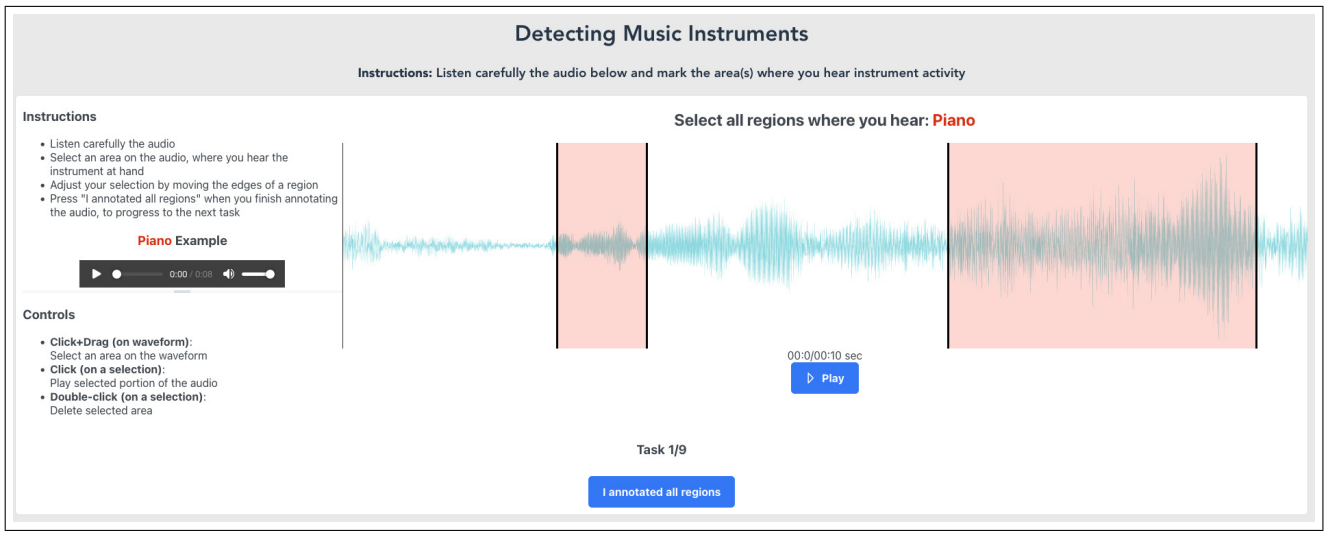


Figure 1: Main audio annotation task

where results in these two categories were found to most significantly predict their auditory capabilities.

The questions of both GMSI categories were aggregated to one questionnaire, with one attention question placed in-between the questionnaire's items. The users also had the ability to use a "Back" button to return to a previous question and alter their answer. We used the complete set of questions on both "Music Training" and "Perceptual Abilities", after consulting the online GSMI "configurator" [5] .

The users were greeted with an "Instructions" message before the main annotation task, which described the steps to complete each microtask and a warning regarding the volume (as seen in Figure 2). The main audio annotation task (see Figure 1) consisted of four main parts: (a) audio waveform and controls (center-right), (b) instructions and instrument example (upper left), (c) description of controls and (d) submission button with a simple progress indication. The instrument to be identified, was indicated on both (a) and (b) in red, to draw the attention of the users.

Based on the findings during the OpenMic 2018 work [4], the crowd workers were found to struggle to detect multiple instruments at once. To that end, we followed their task design of annotating one instrument at a time; we presented the participants with the audio excerpt and requested to annotate the regions where a chosen single instrument, was active during the recording.

The worker would be presented with an audio excerpt and was instructed to detect the activity of one of the instruments present in the excerpt. The same procedure would follow for each of the instruments per audio excerpt, presented in a random order across genres (e.g. piano from classical music excerpt, followed by the electric guitar from rock music excerpt).

In the audio annotation interface, the users could play and pause the audio excerpt while also draw bounding boxes on the audio waveform. The regions drawn on the waveform were adjustable on both ends and the user could

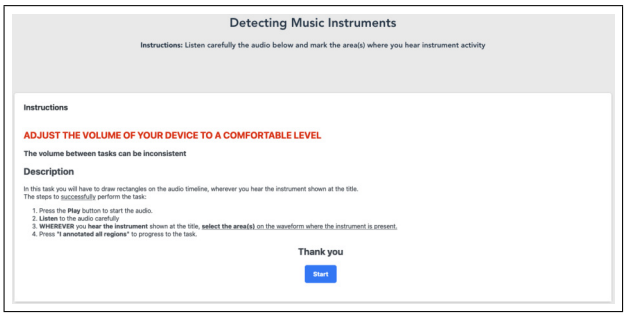


Figure 2: Task instructions and warning

easily dismiss them with a double-click. A single-click on a region would play only the selected part of the audio excerpt. A crowd worker could only progress to the next excerpt if they had drew at least one bounding box on the waveform.

For the design of the interface, we utilized `wavesurfer.js` [6] to draw the waveform and used the `regions` package to enable the bounding boxes interaction. Our choice of these tools was based on previous studies on audio annotation that utilized them successfully [15, 19].

Finally, as mentioned in [4], crowd workers could experience high cognitive load during instrument detection tasks, ultimately affecting their psyche. It was important for us to capture such a phenomenon, so we included the NASA-TLX questionnaire and a free text input to accommodate their feedback towards the study.

3.3 Evaluation methods

Our task design is based around one audio excerpt per genre (10 seconds), where maximum three instruments can play simultaneously. As described before, per task, a worker had to draw the regions where they detect the activity of the selected musical instrument.

[5] `https://shiny.gold-msi.org/gmsiconfigurator/`

[6] `https://wavesurfer-js.org`

To evaluate their performance, we followed the same methods established in [15, 20] and in the Detection and Classification of Acoustic Scenes and Events (DCASE) challenge [21]. We segmented each excerpt into 100ms-long frames which had binary values, depending on the presence or absence of the selected instrument. A frame is considered active when there is an overlap between the annotation region and any portion of the time interval of the frame. We believe that the frame's resolution of 100ms can help us to adequately assess the extent of crowd workers' precision when annotating the temporal activity of an instrument. Based on the ground truth values, we later calculated *Accuracy*, *Precision* and *Recall* of the workers' annotations. To evaluate their performance, we followed the same methods established in [15, 20] and in the Detection and Classification of Acoustic Scenes and Events (DCASE) challenge [21]. We segmented each excerpt ($N = 3$) into 100ms-long frames which had binary values, depending on the presence or absence of the selected instrument. A frame is considered active when there is an overlap between the annotation region and any portion of the time interval of the frame. We believe that the frame's resolution of 100ms can help us to adequately assess the extent of crowd workers' capabilities to detect the temporal activity of an instrument. Based on the ground truth values, we later calculated *Accuracy*, *Precision* and *Recall* of the workers' annotations.

4. RESULTS

The study took place on Prolific, employing 28 crowd workers. We used the built-in prescreening filters of Prolific, setting criteria for fluency in English – for instructions' comprehension and higher chance of affinity to western music – and minimum task approval rate to 90% – to maximise the chances for good-quality work. The reward was set to 4.5 GBP (5.62 USD) which was classified as "*Good*" by the platform. We preserved the results of the 14 workers (see their demographics on Table 1) that successfully passed the attention question. Filtering the results based on the attention questions.

	Variables	**Statistics**
Gender, n	Female	10
	Male	17
	Prefer not to say	1
Age (years)	Range	18-55
Occupation	Full-time	12
	Part-time	5
	Unemployed	11
Education	Associate degree	2
	Bachelor's degree	12
	High school/HED	4
	Master's degree	4
	Some college, no diploma	3
	Technical/trade/vocational training	3

Table 1: Participant demographics

4.1 Demographics and Equipment

The workers used mostly earphones, headphones and laptop speakers, while three reported using dedicated speakers. Most workers (15) reported the quality of their equipment as "Excellent", with the majority (22) reporting "Imperceptible" impairment. Finally, the majority (15) reported that conducted the study in near silence conditions, while one reported performing the tasks in an environment with high noise levels.

4.2 Detecting Musical Instruments

The crowd workers showed high performance detecting most instrument activities on all three audio clips (`RQ1`). Studying the results per genre, we see on Table 2 that "Clarinet" was the most easily identifiable instrument. In the given audio excerpt, "Clarinet" had a prominent and distinct timbre, compared to the rest of the instruments. This might have helped annotators to detect its activity correctly. "Piano" on the other hand was more difficult to detect its temporal activity, as it accompanied the rest of the instruments with a softer tone.

	Accuracy	Precision	Recall
Piano	70.6%	91.5%	66.5%
Clarinet	**84.5%**	**95.8%**	**82.9%**
Cello	62.6%	95.5%	59.6%

Table 2: Accuracy, Precision, Recall and F-score on Classical audio excerpt (the highest scores per metric are in bold)

"Cello" though appears to be the hardest instrument to detect in the audio excerpt, as both accuracy and recall are near 60%. The high precision combined with low accuracy, could indicate that most workers mistook the activity of another instrument, with that of a cello. The results are surprising, as "Cello" was equally prominent as the "Clarinet", playing at a lower register than the rest of the instruments.

In the case of "Jazz" we find the "Drums" to be the most recognizable instrument, while "Double Bass" yielded better results than "Cello" in the "Classical" excerpt (see Table 3. Recordings of "Double Bass" in jazz can vary from barely noticeable to accentuated, depending on the recording setting or the part of the song (being more prominent during solo performance). Despite being the prominent instrument alongside "Drums" for a large portion of the excerpt, the workers still had trouble identifying the regions where it was active.

	Accuracy	Precision	Recall
Piano	81.8%	70.9%	**87.7%**
Double Bass	64%	**100%**	64%
Drums	**84.4%**	**100%**	84.4%

Table 3: Accuracy, Precision, Recall and F-score on Jazz audio excerpt (the highest scores per metric are in bold)

It is very interesting to highlight how the performance on "Piano" which is present in both "Classical" and "Jazz" music clips, changes greatly between the two samples. A possible explanation could be on the rather more prominent role it plays in piano jazz trios, where in most cases carries the melodic part of a composition (which would explain also the high recall score). In this specific example, we see that on average the crowd workers accurately selected the small rhythmic bursts of piano play, although not as precisely. This shows that they could definitely detect its activity correctly, but could not indicate precisely its onset and offset regions.

	Accuracy	Precision	Recall
Electric Guitar	**91.7%**	96.5%	**91.6%**
Bass Guitar	82.4%	**100%**	82.4%
Drums	73%	**100%**	73%

Table 4: Accuracy, Precision, Recall and F-score on Rock audio excerpt (the highest scores per metric are in bold)

The participants performed better on average, in the "Rock" excerpt. We speculate that the sounds of "Electric Guitar" and "Bass Guitar" are more familiar to the demographics of the participating workers, who scored quite highly on accuracy and recall, on both instruments.

The sustained power chord of the "Electric Guitar" was easy to identify and correctly annotate its onset and offset. On the other hand, despite "Drums" and "Bass Guitar" being present during the entirety of the audio excerpt, crowd workers found "Drums" more difficult to recognize correctly, despite the results in the jazz excerpt. Difference in "Drums" between the two excerpts, show higher use of the snare drum in the jazz excerpt, while in the rock, the use of lower tone tom drums was more prominent.

4.3 Self-assessed Music Characteristics and Performance

On Table 5 we see the self-assessed "Perceptual Abilities" and self-reported "Musical Training" of the participants. The low "Musical Training" is consistent with the results of [13] but pretty low when compared to the participant pool of [12] (scoring near the bottom 30% of the population in the original study).

	Range	Median	Standard Deviation(1σ)
Perceptual Abilities	29-63	47.5	8.19
Musical Training	7-41	18.5	9.04

Table 5: Range, Median, Mean and Standard Deviation of Perceptual Abilities and Musical Training

The self-assessed "Perceptual Abilities" are also low compared to the sample of [12] but considerably higher than in [13]. The results in our study certainly showcase adequate perceptual skills, in regards with the task at hand.

We study the connection of their musical properties to their performance from a more qualitative perspective, due to the size of our participant pool. Their self-assessed "Perceptual Abilities" show that the users felt quite confident on the degree they can detect musical traits on sound, despite their lack of expertise as shown by their "Musical Training" average score (Table 5).

Comparing their assessment to their actual performance we further see that their "Musical Training" is not indicative of their capability to detect temporal activity of musical instruments. Their median score as shown on the table, is close to the low 25th percentile of the results in the original GMSI study [12], showing a general low formal musical training. While formal training could certainly be beneficial for such tasks, people are still exposed to different musical instruments through casually enjoying music, especially as it is widely and easily accessible through streaming services. We also believe that the task design with the inclusion of an audio example of a given instrument, assisted the workers in their task to identify instruments.

4.4 Cognitive Load and Feedback

The results on the NASA-TLX questionnaire, show that from the total of 14 crowd workers, 10 found the task's difficulty average, while 9 were very confident on their performance. All of the participants reported average to low mental and physical demand, with mental load being higher than the physical. 10 workers experienced very low temporal demand, with most finishing the study in near 10 minutes. The results though show that the workers' self-assessed performance varied greatly between individuals, with scores from "Very Low" to "Very High".

Finally, crowd workers expressed their opinions on the study through a free form text area. Through their feedback we found that they greatly enjoyed the study through comments such as: "*Study was very well thought out. Nothing else to add.*", "*It was fun, I would love to take part similar studies again*" and "*the study was interesting and I am finding the piano very interesting instrument after this study*". Some even gave their insights for future improvements in comments such as: "*Put more instruments in there*" and "*it was ok but i propose next time the sounds be played slowly for us to easily identify. thank you*".

5. DISCUSSION

Non-experts exhibited high precision with a rather high recall on most instruments, especially on the "Jazz" and "Rock" audio clips. Despite their low expertise as indicated through the "Musical Training" attribute, the results show that they were capable of perceiving the temporal activity of instruments. These abilities are in line with the findings from [13] but also people's innate understanding of music, as shown in studies [22–24].

The high precision scores combined with lower accuracy and recall scores though, could indicate that the participants underestimated the activity of the instruments in

the excerpts. This means that the users although detected correctly segments of an instrument's activity, they weren't able to identify the totality of temporal activity for the given instrument. By selecting more, smaller and precise regions, one would select only the most prominent "True Positive" frames in an excerpt, but fail to select all of them, as is apparent on the cases of "Cello" and "Double Bass". Additionally, in our evaluation, we used a quite short and strict frame resolution which could potentially affect their recall scores. However, further studies are needed with variable frame resolution to test its suitability for this type of annotation task.

While it is inevitable to experience issues of sampling bias when executing crowdsourcing studies (i.e. participants will always be a smaller set of the userbase, which by itself is highly specific and smaller than the general public), we justify the differences with [12] based on the form of incentive from the side of participants, to perform the study. In our case the incentive was strictly monetary, therefore we employed participants who could be less enthusiastic about music, compared to [12]. When comparing to [13] though, while the results are consistent regarding "Musical Training", the results on "Perceptual Abilities" were higher in our case, despite the use of the same crowdsourcing platform. Of course, the landscape of crowdsourcing platforms is constantly changing, but it could be a nice indication of adequately, musically perceptive crowd workers.

Finally, we believe that our interface design with the inclusion of short examples of the musical instruments on each task, must have assisted the crowd workers during annotation. We encourage further experimentation on interface design, to explore effective ways to assist workers during their audio annotation task.

Limitations. Being an exploratory study, we acknowledge that the number of participating crowd workers is lower than in traditional crowdsourcing studies. Nonetheless, we believe that the rigorous set up and the in-depth qualitative analysis of the obtained results allow us to provide valuable and robust insights, which could be used to design and deploy larger-scale studies in the future.

The music excerpts we used in our study focus on popular genres of music. As such, despite the diverse demographics of Prolific, the participants in our study were expected to be familiar with the instruments in our excerpts. We strongly encourage future studies to experiment with instruments of different traditions, as we believe that similar techniques could yield equally promising results for those instruments.

6. CONCLUSION

Our study focuses on exploring the ability of non-experts to identify the temporal activity of musical instruments in audio excerpts of western music. This is an important task during dataset production for instrument recognition, as it can provide strongly-labeled annotations which enable event detection classification tasks. Results show that untrained crowd workers can successfully detect the activity of instruments like *clarinet* and *electric guitar*, one at a time, given an example of the instrument. The overall cognitive load that workers experienced was average, while most of them expressed their enjoyment of the tasks through free-form feedback. The positive outcomes of this work encourage conducting further studies on the topic, with focus on a larger participant pool and a more extensive evaluation dataset that includes additional genres, instruments, and identification complexities.

7. REFERENCES

[1] S. Nowak and S. Rüger, "How reliable are annotations via crowdsourcing: a study about inter-annotator agreement for multi-label image annotation," in *Proceedings of the international conference on Multimedia information retrieval*, 2010, pp. 557–566.

[2] T. Yan, V. Kumar, and D. Ganesan, "Crowdsearch: exploiting crowds for accurate real-time image search on mobile phones," in *Proceedings of the 8th international conference on Mobile systems, applications, and services*, 2010, pp. 77–90.

[3] N. Sawant, J. Li, and J. Z. Wang, "Automatic image semantic interpretation using social action and tagging data," *Multimedia Tools and Applications*, vol. 51, no. 1, pp. 213–246, 2011.

[4] E. Humphrey, S. Durand, and B. McFee, "Openmic-2018: An open data-set for multiple instrument recognition." in *ISMIR*, 2018, pp. 438–444.

[5] H. Schreiber and M. Müller, "A crowdsourced experiment for tempo estimation of electronic dance music." in *ISMIR*, 2018, pp. 409–415.

[6] S. Grollmisch, E. Cano, C. Kehling, and M. Taenzer, "Analyzing the potential of pre-trained embeddings for audio classification tasks," in *2020 28th European Signal Processing Conference (EUSIPCO)*. IEEE, 2021, pp. 790–794.

[7] J. J. Bosch, J. Janer, F. Fuhrmann, and P. Herrera, "A comparison of sound segregation techniques for predominant instrument recognition in musical audio signals." in *ISMIR*. Citeseer, 2012, pp. 559–564.

[8] Y. Han, J. Kim, and K. Lee, "Deep convolutional neural networks for predominant instrument recognition in polyphonic music," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 1, pp. 208–221, 2016.

[9] T. Kitahara, M. Goto, K. Komatani, T. Ogata, and H. G. Okuno, "Instrument identification in polyphonic music: Feature weighting to minimize influence of sound overlaps," *EURASIP Journal on Advances in Signal Processing*, vol. 2007, pp. 1–15, 2006.

[10] D. L. Wessel, "Timbre space as a musical control structure," *Computer music journal*, pp. 45–52, 1979.

[11] J. Pons, O. Slizovskaia, R. Gong, E. Gómez, and X. Serra, "Timbre analysis of music audio signals with convolutional neural networks," in *2017 25th European Signal Processing Conference (EUSIPCO)*. IEEE, 2017, pp. 2744–2748.

[12] 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.

[13] I. P. Samiotis, S. Qiu, C. Lofi, J. Yang, U. Gadiraju, and A. Bozzon, "Exploring the music perception skills of crowd workers," in *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, vol. 9, 2021, pp. 108–119.

[14] 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.

[15] M. Cartwright, A. Seals, J. Salamon, A. Williams, S. Mikloska, D. MacConnell, E. Law, J. P. Bello, and O. Nov, "Seeing sound: Investigating the effects of visualizations and complexity on crowdsourced audio annotations," *Proceedings of the ACM on Human-Computer Interaction*, vol. 1, no. CSCW, pp. 1–21, 2017.

[16] J. Salamon, D. MacConnell, M. Cartwright, P. Li, and J. P. Bello, "Scaper: A library for soundscape synthesis and augmentation," in *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2017, pp. 344–348.

[17] I. Recommendation, "General methods for the subjective assessment of sound quality," *ITU-R BS*, pp. 1284–1, 2003.

[18] E. F. Beach, W. Williams, and M. Gilliver, "The objective-subjective assessment of noise: Young adults can estimate loudness of events and lifestyle noise," *International journal of audiology*, vol. 51, no. 6, pp. 444–449, 2012.

[19] M. Marolt, C. Bohak, A. Kavčič, and M. Pesek, "Automatic segmentation of ethnomusicological field recordings," *Applied Sciences*, vol. 9, no. 3, p. 439, 2019.

[20] A. Mesaros, T. Heittola, and T. Virtanen, "Metrics for polyphonic sound event detection," *Applied Sciences*, vol. 6, no. 6, p. 162, 2016.

[21] D. Stowell, D. Giannoulis, E. Benetos, M. Lagrange, and M. D. Plumbley, "Detection and classification of acoustic scenes and events," *IEEE Transactions on Multimedia*, vol. 17, no. 10, pp. 1733–1746, 2015.

[22] A. M. Liberman and I. G. Mattingly, "The motor theory of speech perception revised," *Cognition*, vol. 21, no. 1, pp. 1–36, 1985.

[23] S. Koelsch, K. Schulze, D. Sammler, T. Fritz, K. Müller, and O. Gruber, "Functional architecture of verbal and tonal working memory: an fmri study," *Human brain mapping*, vol. 30, no. 3, pp. 859–873, 2009.

[24] B. Gingras, H. Honing, I. Peretz, L. J. Trainor, and S. E. Fisher, "Defining the biological bases of individual differences in musicality," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 370, no. 1664, p. 20140092, 2015.

MoisesDB: A DATASET FOR SOURCE SEPARATION BEYOND 4-STEMS

Igor Pereira **Felipe Araújo** **Filip Korzeniowski** **Richard Vogl**
Moises Systems Inc., Salt Lake City, USA.
igor@moises.ai

ABSTRACT

In this paper, we introduce the MoisesDB dataset for musical source separation. It consists of 240 tracks from 45 artists, covering twelve musical genres. For each song, we provide its individual audio sources, organized in a two-level hierarchical taxonomy of stems. This will facilitate building and evaluating fine-grained source separation systems that go beyond the limitation of using four stems (drums, bass, other, and vocals) due to lack of data. To facilitate the adoption of this dataset, we publish an easy-to-use Python library to download, process and use MoisesDB. Alongside a thorough documentation and analysis of the dataset contents, this work provides baseline results for open-source separation models for varying separation granularities (four, five, and six stems), and discuss their results.

1. INTRODUCTION

Source separation is the task of splitting an audio signal into separate signals for each signal source. For music, the signal sources are the instruments that appear in the track, e.g.: guitar, bass, piano, drums, and vocals.

Music source separation is a relevant task within music information retrieval. While it can be used as a preprocessing step for other tasks (e.g. voice separation for f0 tracking), source separation enables diverse applications on arbitrary music tracks that would need manual creation of stems otherwise. For example, in the context of music education, the creation of play-along tracks for students, facilitating by-ear transcription of relevant instruments, or automatic creation of karaoke backing tracks. Such applications are relevant for industry, as demonstrated by initiatives like the demixing challenges [1] .

State-of-the-art source separation systems are usually built using neural-network-based machine learning systems, trained in a supervised way [1–3]. In order to train these systems, a large amount of training data is required. For supervised approaches, the training data is represented by pairs of *i*. a mixed audio track and *ii*. a set of so-called stems that, when combined, recreate the audio track. Stems are audio signals containing only one (or a group of related) sources, i.e. instruments. A pair of one mixed track and its corresponding stems constitutes one training example.

Dataset	Year	No. of Tracks	Stems / Multitracks
MedleyDB [4]	2014	122	Multitracks
MedyleyDB-V2 [5]	2016	196	Multitracks
DSD100 [6]	2015	100	4 Stems
MUSDB18 [7]	2017	150	4 Stems
MUSDB18-HQ [8]	2019	150	4 Stems
MoisesDB	2023	240	Multitracks

Table 1. Overview of publicly released datasets for music source separation. The datasets are grouped according to the set of tracks they contain. For example, DSD100 is a subset of MUSDB18. Additionally, 46 songs from MedleyDB are also used in MUSDB18.

Besides the large amount of manual work involved in any large-scale dataset creation, this kind of data is especially hard to come by for several reasons. Whenever dealing with music audio data, legal issues may arise by collecting and sharing a dataset. The copy and distribution rights for most music are held by music publishers and record labels and are enforced rigorously. Obtaining the audio recordings for the individual instruments (stems) along with the final mix may expose recording, mixing, and mastering techniques of the recording studios, responsible for producing a track, which is why recording studios may oppose the publishing of stems in order to keep their trade secrets. Finally, processing, exporting, and organizing stems from recording projects (often from a digital audio workstation) is a considerable task. Usually, these recording projects are created without considering the requirement of exporting instrument stems. All these factors hinder the creation and release of multitrack and stem datasets.

While there exist source separation datasets aimed at a specific task, like vocal separation [9, 10], these are only of limited relevance for the more general task of splitting audio tracks up into stems. A majority of the existing stem datasets [6, 8] use a limited taxonomy of four stems, namely: vocals, drums, bass, and other. While this has become a de-facto standard for works on source separation [1–3] due to the availability of data and comparability of results, this is a strong limitation of the resulting mod-

[1] https://www.aicrowd.com/challenges/music-demixing-challenge-ismir-2021/sound-demixing-challenge-2023

© I. Pereira, F. Araújo, F. Korzeniowski, and R. Vogl. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** 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.*, Milan, Italy, 2023.

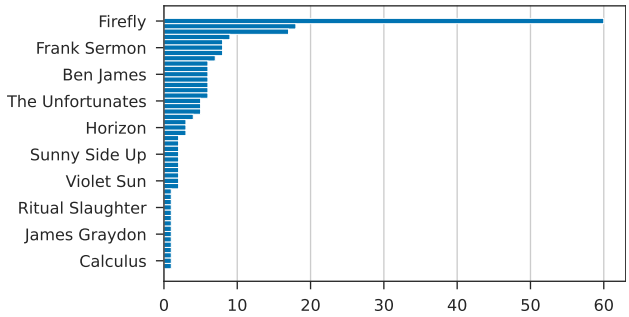


Figure 1. Artist distribution of MoisesDB.

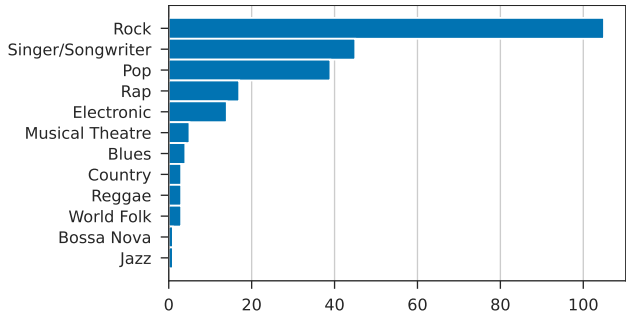


Figure 2. Genre distribution of MoisesDB.

els. For many practical applications, separation of other, widely used instruments may be relevant: e.g. guitars, keys, strings, etc.

Datasets featuring individually recorded tracks (multitrack, e.g. [5]), as well as other collections of multitrack recordings, like *Open Multitrack Testbed* [11], do exist. However, these are not prepared to be used for source separation, out of the box, and may come with license restrictions. Looking at recent source separation publications, we see that non-public data usually represents the bulk of training data (e.g. Bean dataset [12] in [1]; 800 tracks of undisclosed source in [3]). This hints that by only using publicly available data, it is not possible to train competitive source separation models. Thus, there is a need for more free data featuring a more detailed taxonomy, in order to be able to successfully train and test robust source separation models with the capability to separated more stems.

To improve the current situation, we introduce MoisesDB, a multitrack dataset featuring track annotations and a taxonomy to group individual tracks into stems. This dataset is offered free of charge for non-commercial research use only. It consists of 240 music tracks from different artists and genres with a total duration of over 14 hours. Along with the dataset, we provide baseline performance values for state-of-the-art source separation systems.

The remainder of this work is structured as follows: Section 2 covers related work and contrasts it with the dataset presented here. Section 3 discusses the details of MoisesDB. Section 4 introduces baseline performance evaluation statistics using freely available source separation models. Finally, Section 5 provides concluding remarks.

Stem	Track
Bass	Bass Guitar, Bass Synthesizer, Contrabass
Bowed Strings	Cello, Cello Section, Other Strings, String Section, Viola Section, Viola Solo
Drums	Cymbals, Drum Machine, Full Acoustic Drumkit, Hi-Hat, Kick Drum, Overheads, Snare Drum, Toms
Guitar	Acoustic Guitar, Clean Electric Guitar, Distorted Electric Guitar
Other	Fx
Other Keys	Organ, Electric Organ, Other Sounds, Synth Lead, Synth Pad
Other Plucked	Banjo/Mandolin/Ukulele/Harp
Percussion	A-Tonal Percussion, Pitched Percussion
Piano	Electric Piano, Grand Piano
Vocals	Background Vocals, Lead Female Singer, Lead Male Singer, Other
Wind	Brass, Flutes, Other Wind, Reeds

Table 2. MoisesDB stem-track taxonomy used to organize individual tracks into stems.

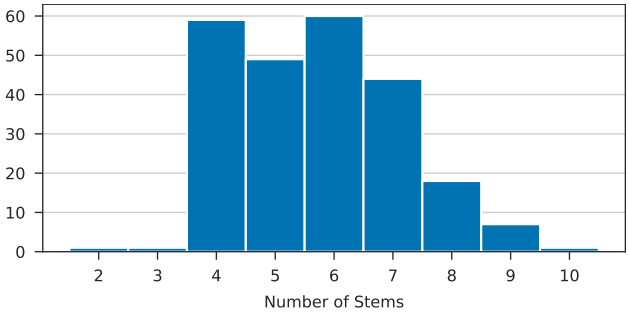


Figure 3. Number of stems per track in MoisesDB.

2. RELATED WORK

In the past, several multitrack and stem datasets have been published by the community (see Tab. 1). This section will discuss their properties and set the context for the dataset presented in this work. Since the main focus of this work is source separation into as many stems as possible, single stem focused datasets (e.g. voice separation datasets [9, 10]) will be mainly ignored.

In 2014, Bittner et al. released the *MedleyDB* dataset [4], which comprises 122 songs in multitrack format. It was extended by 74 songs (totalling 196 songs) in 2016, and published as *MedleyDB 2.0* [5]. The dataset provides audio files in a hierarchical structure, where the final mix is split into multiple stems, each containing numerous raw audio sources (multitracks). Besides the multitrack data, the MedleyDB dataset provides an extensive list of metadata, such as artist, track name, origin, genre, and producer, amongst others. Additionally it provides multiple annotations, such as instrument activation, melody, and pitch.

The annotations in MedleyDB make it useful for many MIR tasks, including the source separation of diverse instruments. However, the shortcoming of MedleyDB for

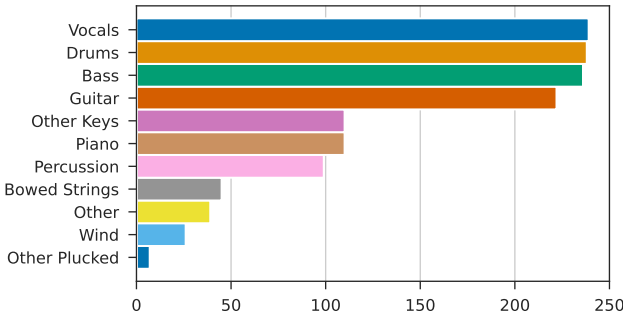


Figure 4. Distribution of stems in MoisesDB.

```
from moisesdb.dataset import MoisesDB

db = MoisesDB(data_path='./moises-db-data')
n_songs = len(db)
track = db[0]
# mix multitracks to stems
stems = track.stems
# stems = {
#   'vocals': np.ndarray (stem audio data),
#   'bass': np.ndarray (stem audio data),
#   ...}
mixture = track.audio # mixture: np.ndarray
track.save_stems('./stems/track_0') # save mixed stems
```

Listing 1: Usage of the MoisesDB Python package.

music source separation is the way it organizes tracks into stems. While it provides instrument information for each of them, and functional annotations for stems (such as "melody" or "bass"), stems are not meaningfully labelled, only numbered. As a result, stem 01 of one song may be the drum kit, while stem 01 of a another mix is the bassoon. Furthermore, instruments—and thus tracks—are grouped according to how they physically produce their sound, rather than their role in the mix of a song. For example, the "drum machine" falls into the same category as "electric piano", namely "electric→electronic". These shortcomings make it cumbersome to use for music source separation out of the box and significant work has to be done in order to use it for this task.

In 2016, Liutkus et al. released the *DSD100* [6] dataset as part of the 2016 signal separation evaluation campaign to develop and benchmark source separation models. It contains 100 songs and uses the the four-stems taxonomy (vocals, drums, bass, and other). Later, in 2017, Rafii et al. extended DSD100 to 150 songs by adding 46 pieces from MedleyDB, and including four previously unreleased recordings from commercial providers. This dataset became known as the *MUSDB18* [7] dataset, and was used for the the 2018 signal separation evaluation campaign. In 2019, Z. Rafii et al. released an uncompressed version of the MUSDB18 dataset, MUSDB18-HQ [8]. As its predecessor DSD100, this dataset provides four stems—vocals, drums, bass, and other—as well as linear mixes. MUSDB18 is widely used to train and benchmark source separation models, but the limited number of stems prevents researchers from building more granular source separation systems.

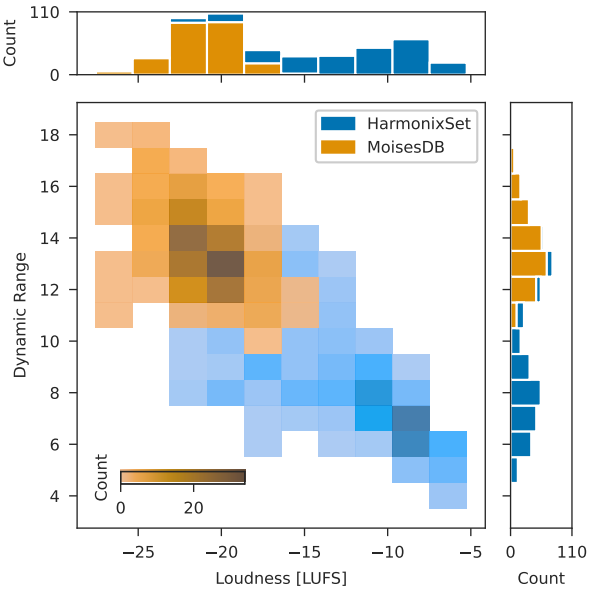


Figure 5. Loudness and Dynamic Range distribution of tracks in MoisesDB. For a comparison with commercially mixed and mastered songs, we sampled 240 tracks from the HarmonixSet [13].

In summary, data for training granular source separation systems is scarce: the 150 tracks from MUSDB18 are ready to use, but offer only four stems to separate; the 140 remaining tracks from MedleyDB (46 of the originally 196 are already part of MUSDB18) are not organized in a way that easily supports source separation research. This issue is also reflected in the fact that state-of-the-art source separation models often use larger, non-public datasets for training [1, 3], or have to resort to synthetic training data (e.g. [14, 15]). Other works find that MUSDB18's *"source groupings remain overly coarse for many real-world remixing applications."* [16]. To address these issues and to foster more research in music source separation, we created the MoisesDB dataset.

MoisesDB comprises the largest publicly available set of multitrack audio recordings—240 previously unreleased songs—organized in a taxonomy that reflects the needs of source separation systems (as detailed in Sec. 3.1). The large number of songs, the diverse types of stems and tracks, and their organization in a source-separation-focused taxonomy will allow researchers to build their own stems according to their own requirements, and thus develop more granular source separation systems.

3. DATASET

MoisesDB consists of 240 songs by 47 artists that span twelve high-level genres. Both artists and genres follow a power-law-like distribution, where the majority of songs belong to few genres and are performed by few artists—see Fig. 1 and 2. The total duration of the dataset is 14 hours, 24 minutes and 46 seconds, where the average recording is 3:36 seconds, with a standard deviation of 66 seconds.

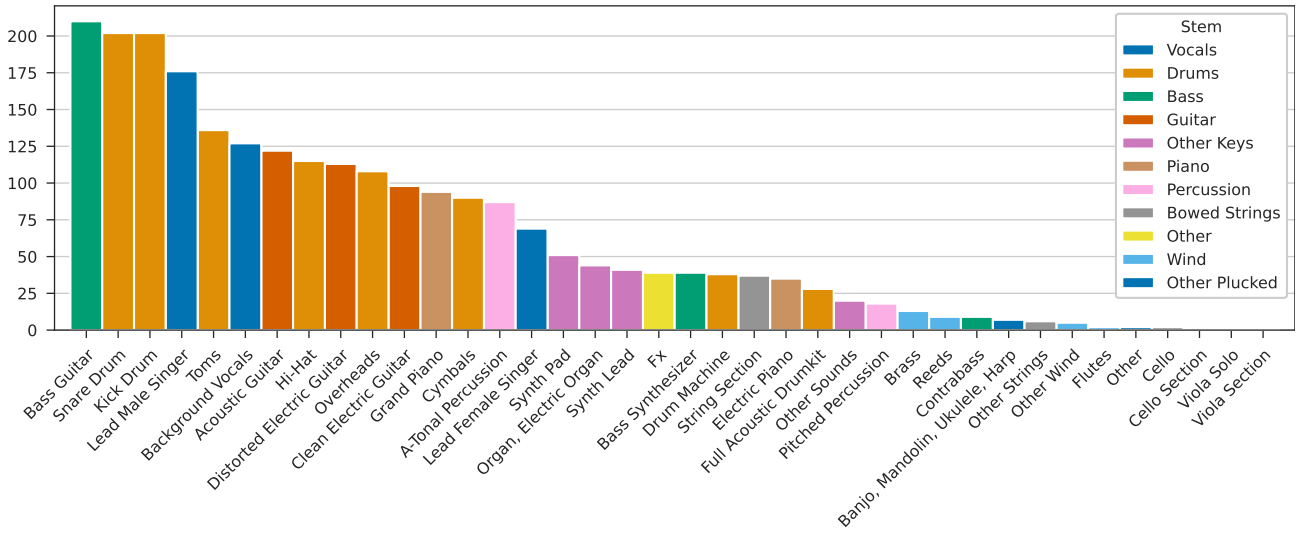


Figure 6. Distribution of tracks in MoisesDB.

3.1 Stem Taxonomy

Modern song recordings consist of multiple recorded *tracks*, which can be grouped and down-mixed into a smaller number of *stems*. For example, the "drums" stem might comprise tracks for the snare drum, the bass drum, hi-hat, cymbals, and so on. MoisesDB provides all individual tracks for each song, grouped into stems by the taxonomy shown in Table 2. This taxonomy reflects the recording & mixing process, and thus facilitates its reversal—music source separation—by grouping the raw tracks into semantically labeled stems. This also means that songs may consist of different numbers of stems, as shown in Fig. 3. MoisesDB thus facilitates many future research directions: source separation models for a larger number of stems, data augmentation through mixing stems on-the-fly from their tracks, or separation of individual tracks from a stem, to name a few.

Given the genres of the songs in MoisesDB, certain stems are more common in the dataset than others: "vocals", "drums", and "bass" appear on virtually every song, while "wind" is rare. Similarly, certain tracks appear much more frequently than others, both within stems ("bass guitar" vs. "contrabass") and between stems ("snare drum" vs. "cello"). Figs. 4 and 6 show the distributions of stems and sources, respectively.

We anticipate that this imbalance will present a challenge in training source separation models for underrepresented stems, as it is likely that certain tracks, such as "other plucked" tracks, will still be difficult to distinguish from "guitar" tracks if trained solely on MoisesDB. However, the available data provides an opportunity for researchers to better identify and characterize errors made by their models. For instance, instead of simply observing that the separated "other" stem bleeds into "guitar," MoisesDB enables researchers to pinpoint this issue to tracks where "other" includes plucked instruments.

3.2 Recording and Mastering

The songs in MoisesDB are professionally recorded in stereo. The individual tracks are combined additively to create stems, which are then mixed together to produce the final version of the song. Due to technical limitations during recording, minuscule bleeding from other stems may be present for some of the tracks. No compression, equalization, or other effects are used during the mixing process, and the songs are not subjected to mastering. As a result, the song mixes have a lower loudness and a higher dynamic range than professionally mastered commercial songs. This raises concerns about the distributional shift between un-mastered training data and commercial recordings. Indeed, Jeon and Lee [17] have found that training separation models using mastered mixes can improve separation quality. However, providing un-mastered mixes is common in existing datasets such as MUSDB18, and models such as HT-Demucs [3] generalize reasonably well to mastered recordings, even if trained on un-mastered data.

Figure 5 shows the loudness and dynamic range distributions for the dataset, where loudness is measured in LUFS (Loudness Units relative to Full Scale) [18], and Dynamic Range is computed based on the definitions of the "Pleasurize Music Foundation" as implemented in the "DR14 T.meter" software [2] .

3.3 Python Library

With MoisesDB comes a Python library that facilitates working with the dataset by parsing metadata and automatically building stems and mixes. Listing 1 shows an example usage of the library. The code shown there initializes the library, retrieves the number of tracks, creates the stems and the full mix, and saves the individual stems to a directory. For a detailed and up-to-date documentation, we refer the reader to the GitHub repository [3] .

[2] https://github.com/simon-r/dr14_t.meter
[3] https://github.com/moises-ai/moises-db

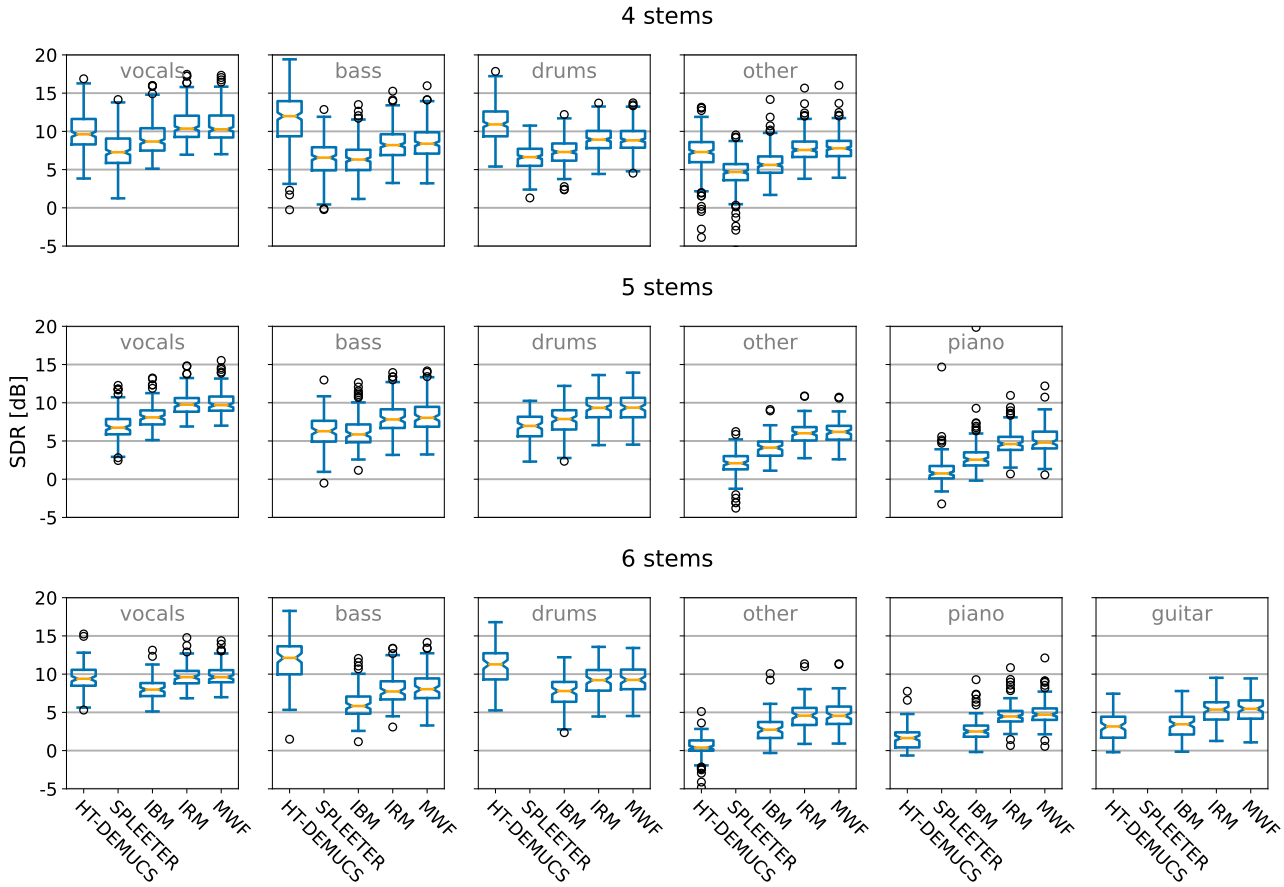


Figure 7. SDR values of each group of sources, for IBM, IRM, MWF, Demucs, and Spleeter source separation methods.

	HT-Demucs		Spleeter		IBM		IRM		MWF	
	Mean ± Std	Mdn	Mean ± Std	Mdn	Mean ± Std	Mdn	Mean ± Std	Mdn	Mean ± Std	Mdn
4 stems (N = 235)										
vocals	10.05 ± 2.48	9.62	7.61 ± 2.45	7.27	9.02 ± 2.13	8.67	**10.72 ± 2.03**	10.37	**10.72 ± 2.11**	10.27
bass	**11.64 ± 3.35**	11.99	6.46 ± 2.26	6.57	6.46 ± 2.08	6.31	8.43 ± 2.03	8.20	8.68 ± 2.07	8.38
drums	**10.94 ± 2.30**	10.91	6.65 ± 1.72	6.64	7.33 ± 1.77	7.30	8.98 ± 1.68	8.92	9.01 ± 1.67	8.83
other	7.00 ± 2.76	7.30	4.45 ± 2.26	4.69	5.77 ± 1.72	5.61	7.74 ± 1.65	7.57	**7.90 ± 1.65**	7.79
overall	**9.91 ± 3.27**	9.69	6.29 ± 2.47	6.24	7.14 ± 2.28	6.99	8.97 ± 2.16	8.81	9.08 ± 2.15	8.87
5 stems (N = 104)										
vocals			6.99 ± 1.97	6.74	8.29 ± 1.66	8.08	9.94 ± 1.59	9.75	**10.01 ± 1.71**	9.68
bass			6.26 ± 2.27	6.28	6.13 ± 2.15	5.86	8.02 ± 2.07	7.82	**8.32 ± 2.08**	8.03
drums			6.89 ± 1.88	6.97	7.67 ± 1.94	7.87	9.29 ± 1.84	9.34	**9.32 ± 1.84**	9.36
other			1.97 ± 1.76	2.09	4.04 ± 1.47	4.13	6.00 ± 1.44	6.01	**6.10 ± 1.48**	6.19
piano			1.17 ± 1.86	0.75	3.04 ± 2.37	2.55	4.99 ± 2.32	4.60	**5.30 ± 2.46**	4.79
overall			4.66 ± 3.20	5.02	5.12 ± 2.81	4.87	7.65 ± 2.66	7.60	**7.81 ± 2.66**	7.83
6 stems (N = 88)										
vocals	9.55 ± 1.87	9.39			8.09 ± 1.51	7.98	9.73 ± 1.46	9.61	**9.81 ± 1.49**	9.61
bass	**11.93 ± 2.87**	12.13			6.04 ± 1.98	5.83	7.92 ± 1.93	7.73	8.24 ± 1.96	8.03
drums	**11.02 ± 2.44**	11.28			7.58 ± 1.96	7.79	9.19 ± 1.86	9.21	9.23 ± 1.85	9.25
other	0.28 ± 1.84	0.39			2.85 ± 1.76	2.74	4.67 ± 1.76	4.57	**4.72 ± 1.82**	4.55
piano	1.60 ± 1.68	1.64			2.78 ± 1.61	2.49	4.71 ± 1.61	4.47	**4.97 ± 1.74**	4.70
guitar	3.07 ± 1.81	3.16			3.35 ± 1.54	3.44	5.28 ± 1.54	5.36	**5.41 ± 1.65**	5.46
overall	6.24 ± 5.17	6.05			5.12 ± 2.81	4.87	6.91 ± 2.70	6.69	**7.06 ± 2.73**	6.89

Table 3. Mean, standard deviation (Std), and median (Mdn) of the SDR in dB for each Model/Method and stem type. The varying number of available tracks is denoted by N. *Overall* indicates performance over all tracks regardless of stem group. Best results are marked in bold.

4. BENCHMARKING

In order to establish reference values for each track of the MoisesDB dataset, we computed the Source to Distortion Ratio (SDR) [19] scores for Ideal Binary Mask (IBM) [20], Ideal Ratio Mask (IRM) [21], and Multichannel Wiener Filter (MWF) [22] oracle separation methods. Additionally, we assessed SDR scores for two popular public available and open-source architectures: Hybrid Transformer Demucs (HT-DEMUCS) [3] and Spleeter [1]. The SDR scores were calculated for three different groups of sources: four, five, and six stems. Given the architecture of the open-source models, results for Spleeter are available for four and five stems, and for HT-DEMUCS for four and six stems.

The SDR measure [19] represents how much of the energy in a true source signal is preserved in an estimated source signal after applying a separation algorithm. The equation can be defined as

$$\mathrm{SDR} = 10 \log_{10} \frac{\sum_n |s(n)|^2 + \epsilon}{\sum_n |s(n) - \hat{s}(n)|^2 + \epsilon}, \qquad (1)$$

where $s(n)$ represents the true source signal at time n, $\hat{s}(n)$ represents the estimated source signal at time n, and the result is given in decibels (dB).

Table 3 shows the SDR values in dB for each group of stems (4, 5, and 6) evaluated in this benchmark. For a better comparison, we chose the stems available in the open-source models: vocals, bass, drums, other, guitar, and piano. We also pick tracks containing at least all the stems chosen for each group, which explains the distinct number of tracks in Table 3. Songs with more individual tracks than the ones specified for each group were merged into the "other" stem using a linear sum strategy.

Figure 7 depicts boxplots representing the distribution of the SDR metric for both oracle and separation methods, calculated for each group of tracks comprising 4, 5, and 6 stems. The groups of stems evaluated were vocal, bass, drums, other, piano, and guitar. Detailed results for every track and each stem are provided in the GitHub [3] repository.

The first fact that calls our attention can be seen in Figure 7, where the SDR results of IRM and MWF oracle methods did not show a significant difference for all groups of stems. The striking fact is the performance of HT-DEMUCS architecture, which outperforms the oracle methods for bass and drums stems, for the groups of 4 and 6 stems tracks, as we can see in Figures 7 A and C, respectively. Those results contrast with the slightly worse performance of HT-DEMUCS for other, piano, and guitar stems, compared with oracle methods, as seen in Figure 7 C.

5. CONCLUSION

In this work, we introduced MoisesDB, a multitrack dataset with a hierarchical taxonomy aimed at more-than-four-stems source separation. We set the context by analysing the current landscape of source separation datasets and presented a comparison with other relevant datasets along with a detailed analysis of MoisesDB. Specifically, we discussed the organizational taxonomy focused on source separation, the distribution over track duration, the distribution over genres, and the number of songs for each stem and source available in the dataset.

Moreover, we include performance results for two publicly available source separation methods: HT-Demucs, which has the best overall SDR score evaluated on the MUSDB18 test set, and Spleeter, which was one of the first source separation models released and adopted by the general public. We also added results for a few masking-based oracle methods: IBM, IRM, and MWF, which indicate the theoretical performance limits for mask-based source separation models. Additionally, we provide an easy-to-use Python library to access the data which allows fast integration with machine learning libraries.

Overall, this paper represents a detailed report on the MoisesDB dataset, which will hopefully prove to be a great resource for the source separation community in the future. This work aims at facilitating the development of better and extended source separation models as well as providing opportunities to be applied for other use cases, such as automatic mixing and generative accompaniment systems, among others.

6. REFERENCES

[1] 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.

[2] Q. Kong, Y. Cao, H. Liu, K. Choi, and Y. Wang, "Decoupling magnitude and phase estimation with deep ResUNet for music source separation," *arXiv:2109.05418 [cs, eess]*, Sep. 2021.

[3] S. Rouard, F. Massa, and A. Défossez, "Hybrid transformers for music source separation," *arXiv:2211.08553 [cs, eess]*, Nov. 2022.

[4] 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 International Society for Music Information Retrieval Conference (ISMIR)*, Taipei, Taiwan, Oct. 2014, pp. 155–160.

[5] R. Bittner, J. Wilkins, H. Yip, and J. P. Bello, "MedleyDB 2.0: New data and a system for sustainable data collection," in *Late Breaking Demo of the 16th International Society for Music Information Retrieval Conference (ISMIR)*, New York, USA, Aug. 2016.

[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 *Proceedings of the 13th International Conference on Latent Variable Analysis and Signal Separation (LVA/ICA)*, Grenoble, France, Feb. 2017, pp. 323–332.

[7] Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimilakis, and R. Bittner, "The MUSDB18 corpus for music separation," Dec. 2017. [Online]. Available: https://doi.org/10.5281/zenodo.1117372

[8] ——, "MUSDB18-HQ - an uncompressed version of MUSDB18," Dec. 2019. [Online]. Available: https://doi.org/10.5281/zenodo.3338373

[9] T.-S. Chan, T.-C. Yeh, Z.-C. Fan, H.-W. Chen, L. Su, Y.-H. Yang, and R. Jang, "Vocal activity informed singing voice separation with the ikala dataset," in *Proceedings of the 40th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, South Brisbane, QLD, Australia, Apr. 2015, pp. 718–722.

[10] 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.

[11] B. De Man, M. Mora-Mcginity, G. Fazekas, and J. D. Reiss, "The open multitrack testbed," in *Proceedings of the 137th Audio Engineering Society Convention*. Los Angeles, CA, USA: Audio Engineering Society, Oct. 2014.

[12] L. Prétet, R. Hennequin, J. Royo-Letelier, and A. Vaglio, "Singing voice separation: A study on training data," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Brighton, UK: IEEE, May 2019, pp. 506–510.

[13] O. Nieto, M. McCallum, M. E. P. 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 (ISMIR)*, pp. 565–572.

[14] Y. Özer and M. Müller, "Source separation of piano concertos with test-time adaptation," in *Proceedings of the 23rd International Society for Music Information Retrieval Conference (ISMIR)*, Bengaluru, India, Dec. 2022, pp. 493–500.

[15] 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, Oct. 2019, pp. 45–49.

[16] E. Manilow, G. Wichern, and J. Le Roux, "Hierarchical musical instrument separation." in *Proceedings of the 21st International Society for Music Information Retrieval Conference (ISMIR)*, Virtual Conference, Oct 2020, pp. 376–383.

[17] C.-B. Jeon and K. Lee, "Towards robust music source separation on loud commercial music," in *Proceedings of the 23rd International Society for Music Information Retrieval Conference (ISMIR)*, Bengaluru, India, Dec. 2022, pp. 575–582.

[18] ITU-R, "Algorithms to measure audio programme loudness and true-peak audio level," International Telecommunication Union, Recommendation BS.1770-4, Oct. 2015.

[19] 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, Jul. 2006.

[20] E. Vincent, H. Sawada, P. Bofill, S. Makino, and J. P. Rosca, "First stereo audio source separation evaluation campaign: Data, algorithms and results," in *Proceedings of the 7th International Conference on Independent Component Analysis and Signal Separation (ICA)*, London, UK, Sep. 2007, pp. 552–559.

[21] 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)*, South Brisbane, Australia, Apr. 2015, pp. 266–270.

[22] N. Q. K. Duong, E. Vincent, and R. Gribonval, "Underdetermined reverberant audio source separation using a full-rank spatial covariance model," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 7, pp. 1830–1840, Sep. 2010.

MUSIC AS FLOW: A FORMAL REPRESENTATION OF HIERARCHICAL PROCESSES IN MUSIC

Zeng Ren
EPFL
zeng.ren@epfl.ch

Wulfram Gerstner
EPFL
wulfram.gerstner@epfl.ch

Martin Rohrmeier
EPFL
martin.rohrmeier@epfl.ch

ABSTRACT

Modeling the temporal unfolding of musical events and its interpretation in terms of hierarchical relations is a common theme in music theory, cognition, and composition. To faithfully encode such relations, we need an elegant way to represent both the semantics of prolongation, where a single event is elaborated into multiple events, and process, where the connection from one event to another is elaborated into multiple connections. In existing works, trees are used to capture the former and graphs for the latter. Each such model has the potential to either encode relations between events (e.g., an event being a repetition of another), or relations between processes (e.g., two consecutive steps making up a larger skip), but not both together explicitly. To model meaningful relations between musical events and processes and combine the semantic expressiveness of trees and graphs, we propose a structured representation using algebraic datatype (ADT) with dependent type. We demonstrate its applications towards encoding functional interpretations of harmonic progressions, and large scale organizations of key regions. This paper offers two contributions. First, we provide a novel unifying hierarchical framework for musical processes and events. Second, we provide a structured data type encoding such interpretations, which could facilitate computational approaches in music theory and generation.

1. INTRODUCTION

When understanding music as a temporal art, there are at least two properties we need to model. The first is that musical events are ordered in a nontrivial way resembling goal-directedness; the essence of a piece is lost if we "recompose" a piece by performing random temporal permutations. The second phenomenon is the temporal hierarchy, which is a central theme in the understanding of Western tonal music, where we hear multiple entities as the manifestation of a single musical entity. Regarding this hierarchy, there are at least two kinds of such entities. The first kind is a stationary process, such as key region, and harmony; we can say a phrase that enforces a key contains a tonic region (like some presentation in a non-modulating sentence) and we can also describe a time span as an arpeggiation of a harmony. The second kind is a transitory process, such as modulation region, passing, and neighboring motion; a descending third progression contains two stepwise downward motions.

There are multiple attempts to represent the hierarchical structure of such entities. For stationary entities, trees of musical events have been used to model tonal harmony [1, 2], extended tonal harmony [3], jazz harmony [4], rhythm and meter [2, 5]. One limitation of using trees of musical events is that semantics such as passing tones could not be elegantly expressed because one is forced to select either the left or the right parent event for the subordinate event whereas we would like to express an intermediate event subordinate to the melodic motion itself [2].

For the transitory process, trees on event transitions are also sometimes used [6]. They could model the semantics for a passing tone by describing how a melodic motion is split into two motions, one going to the passing note and one leaving the passing note. However, as the fundamental entities are transitions, it can not express the idea that a single event being elaborated in the temporal dimension, such as unfoldings, complete neighbor chords/tones, repetitions, and rearticulations [7].

There are attempts using graphical notations to capture both stationary and transitory processes [7–9]. There are also models [10] that extend such hierarchical organizations beyond the temporal dimension with inner structures of events resembling concurrent processes.

One could potentially encode the hierarchical organization between these two kinds of processes implicitly using networks and graphs, or more expressively using hypergraphs where higher-order relations can be encoded as hyper-edge. One could formulate a rewrite grammar on such networks and hypergraphs to describe the elaboration of nodes and edges. However, we believe there should be a more direct, elegant, and specialized solution (in a similar spirit as [11]) to not only implement but also characterize such generative principles of hierarchical processes.

In summary, there is a lack of formal representation as well as a specially designed data structure that explicitly captures the intricate hierarchical organizations of both the stationary and transitory processes, a fundamental idea in reductive theories of tonal music. This paper offers two contributions. First, we provide a novel unifying hierar-

© Z. Ren, W. Gerstner, and M. Rohrmeier. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).
Attribution: Z. Ren, W. Gerstner, and M. Rohrmeier, "Music as flow: a formal representation of hierarchical processes in music", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

chical framework for stationary and transitory processes. Second, we provide a structured data type encoding such interpretations, which could facilitate computational approaches in music theory, musicology, and algorithmic music composition.

2. THE HIERARCHICAL ORGANIZATIONS OF GENERAL PROCESSES

To demonstrate how these two kinds of processes could be hierarchically organized in the temporal dimension, perhaps it is helpful to consider a scenario in everyday life: "On his way back to home, John went to the supermarket, where he got his favorite yogurt from the fridge. Although he could take a bus directly to his house, he decided to get off one stop earlier by the lake to enjoy a short walk." One hierarchical organization of this particular scenario is depicted in Fig.1. The overarching process is that John went back home from someplace. This transitory process (represented by the arrow connecting "someplace" denoted by X to "home") contains three component processes: a transitory process from "someplace" to "supermarket", a stationary process at "supermarket," and a transitory process from "supermarket" to "home." The stationary process at "supermarket" further contains a stationary process at the "entrance" of the supermarket, a transitory process from "entrance" to the "exit," and a stationary process at the "exit".

2.1 The syntactic constraint of the hierarchical organization of stationary and transitory processes

One pattern that we observe is the mutual recursive relationship between stationary and transitory processes. A stationary process can contain three components (stationary, transitory, stationary). Symmetrically, a transitory process can contain three components (transitory, stationary, transitory).

However, it is clear from the above example (Fig. 1) we can not arbitrarily subdivide a stationary process at X (denoted by $\widehat{X}$), or a transitory process from X to Y (denoted by $\xrightarrow{X \to Y}$) into arbitrary triples of processes, even if they conform to the (stationary, transitory, stationary) or (transitory, stationary, transitory) patterns. We may allow a transitory process $\xrightarrow{A \to B}$ to be elaborated into three components of the form

$$\xrightarrow{A \to X} \widehat{X} \xrightarrow{X \to B}$$

But we would not allow a decomposition like

$$\xrightarrow{C \to D} \widehat{X} \xrightarrow{E \to F}$$

because their states are not compatible.

We can summarize the constraints as the following: a stationary process $\widehat{X}$ may contain ($\widehat{X}$, $\xrightarrow{X \to X}$, $\widehat{X}$); likewise, a transitory process $\xrightarrow{X \to Y}$ may contain ($\xrightarrow{X \to Z}$, $\hat{Z}$, $\xrightarrow{Z \to Y}$). The "entrance" and "exit" in the previous example, although being technically different, are equivalent to "supermarket" from abstract level.

3. LINEAR PROCESSES

We start with characterizing linear processes representing a single hierarchical stream.

3.1 An axiomatic system

We refer to a stationary process as $Joint$, and define it as a predicate J_x indexed by a state $x : A$. We refer to a transitory process as $Link$ and define it as a predicate $L_{x,y}$ indexed by two states x, y of the same type. Then we propose the following four axioms to characterize the hierarchical interactions between of stationary and transitory processes.

$$\forall(x : A)\ \exists(j : J_x) \tag{1}$$

$$\forall(x, y : A)\ \exists(l : L_{x,y}) \tag{2}$$

$$\forall(j, j' : J_x)\ \forall(l : L_{x,x})\ \exists(j^* : J_x) \tag{3}$$

$$\forall(l : L_{x,z})\ \forall(j : J_z)\ \forall(l' : L_{z,y})\ \exists(l^* : L_{x,y}) \tag{4}$$

Axiom 1 states that we may form a stationary process for a given state. Axiom 2 states that we may form a transitory process by for a pair of states of the same kind. Axiom 3 states that we may form a stationary process at x from any triple of processes (j, l, j'), where j and j' are both of stationary processes at x and l is a loop starting and ending at x . Axiom 4 states that we may form a transitory process from any triple of processes (l, j, l'), where l and l' are transitory and j is stationary, provided that their states are compatible.

3.2 A syntax based on dependent type theory

Using two mutually inductive algebraic datatypes, $Joint$ and $Link$, we formalize the notion of hierarchical process in Backus–Naur form (Eq. 5,6,7,8). For a stationary process, the base case (Eq. 5) of a $Joint$ is a $Point$, which means an atomic stationary process, whereas the inductive case (Eq. 6) resembles a stationary process (on the current level) containing two stationary process and a transitory process. The base case of a $Link$ is a $Unit$ (Eq. 7), representing a indivisible change of state, whereas the inductive case (Eq. 8) represents a composite motion that contains two changes and one stationary process. Eq. (5,6,7,8) corresponnds to Axiom. (1,3,2,4) respectively.

$$Joint_{(x:a)} := \langle Point \rangle \quad (x : a) \tag{5}$$

$$\mid \langle Joint \rangle \quad Joint_x \quad Link_{x,x} \quad Joint_x \tag{6}$$

$$Link_{(x:a),(y:a)} := \langle Unit \rangle \quad (x : a) \quad (y : a) \tag{7}$$

$$\mid \langle Link \rangle \quad Link_{x,z} \quad Joint_z \quad Link_{z,y} \tag{8}$$

Dependent typing [12] allows us to define types depending on value. This algebraic data structure with dependent typing has an important application for a generative system. One can define a function using polymorphic recursion to sample a value of the given type. Do-

$$\widehat{X} \xrightarrow{X \to h} \widehat{home}$$

$$\widehat{X} \xrightarrow{X \to s} \widehat{supermarket} \xrightarrow{s \to h} \widehat{home}$$

$$\widehat{X} \xrightarrow{X \to s \sim en} \widehat{entrance} \xrightarrow{en \to ex} \widehat{exit} \xrightarrow{ex \sim s \to l} \widehat{lake} \xrightarrow{l \to h} \widehat{home}$$

$$\widehat{X} \xrightarrow{X \to en} \widehat{entrance} \xrightarrow{en \to f} \widehat{fridge} \xrightarrow{f \to ex} \widehat{exit} \xrightarrow{ex \to l} \widehat{lake} \xrightarrow{l \to h} \widehat{home}$$

Figure 1: A hierarchical interpretation of John's journey. Words on the transitions are abbreviated to save space. The symbol $s \sim en$ means the "entrance" is functionally equivalent to "supermarket" in the interpretation of this journey

.

ing so will guarantee the syntactic correctness of the output. For example, writing a harmonic transition between I and V means sampling a value of type $Link_{I,V}$; elaborating a melodic motion from $\hat{8}$ to $\hat{5}$ becomes sampling a value of type $Link_{\hat{8},\hat{5}}$. Note that between these two examples, the type of the state is different; the first is harmony(roman-numeral) while the second is scale degree. However, within each example, the types of the states are always the same by construction (Axiom 2, 3, 4).

3.3 A data structure in Haskell

Haskell is a functional programming language with an expressive type system [13]. Although the dependent type is not yet built into the language, there exists an encoding involving Algebraic Datatype and Singletons [14] to simulate the behavior of dependent type. The implementation of the linear process is shown below [1] .

```
data Joint (x::a)
    = Point (Sing x)
    | Joint (Joint x) (Link x x) (Joint x)

data Link (x::a) (y::a)
    = Unit (Sing x) (Sing y)
    | forall (z::a). Link (Link x z) (Joint z) (Link z y)
```

3.4 A graphical notation system

To visualize an interpretation of hierarchical processes, we use two types of slurs to connect states in a sequence. For a $Point$, the base case of $Joint$, no visual representation is needed as the state it expresses is sufficient. For a $Unit$, the base case of $Link$, a dashed slur is drawn connecting the starting state to the ending state. Nontrivial stationary processes (the inductive cases) are represented as continuous slurs whereas transitory processes are represented with dashed slur. For a non-trivial processes, the left/right anchor point of the slur is the same as the left/right anchor point of the slur of the process's first/last component.

Five simple examples of such notation are provided in Fig. 2. The easiest to understand is Ex. D, which is a stationary process $\widehat{I}$, decomposed into a $\widehat{I} \xrightarrow{I \to I} \widehat{I}$. This expresses not just repetition but also the loop from I to itself. This loop can be the parent for further elaborations such as passing and neighbor chords. Note that the continuous slur covers a point, a dashed slur, and a point. This visual "covering" is intend to convey the hierarchical relation of processes where parent processes contains its component processes. Ex. C shows a more complex situation; although the top two level is the same as Ex. D (a $Joint$ containing two base case $Joint$ and a $Link$), there are richer structures within the $Link$ in Ex. C. This dashed slur covers a dashed slur, a continuous slur, and a dashed slur, signaling a non-trivial transitory process that contains a transitory process $\xrightarrow{I \quad V}$, a stationary process at V, and a transitory process $\xrightarrow{V \quad I}$. This stationary process at V is also a non-trivial one, capturing the sense of prolongation within which passing notes can be generated connecting its chordal pitches, forming a harmonic the entity $\mathrm{V}^{7-6-5}_{5-4-3}$. Ex. B expresses an overarching stationary process at I containing a nontrivial initial stationary process, a nontrivial transitory process and a trivial stationary process. The initial stationary process resembles a neighbor-passing chord with in the harmony of I. Note that there is a difference in semantic in drawing the continuous slur over the first two Is vs the last two Is. The former means that the overarching motion is stationary first and then moving to the target I chord, whereas the latter means the motion moves first and then performs a stationary rest. This kind of fine distinction could be used to further express embodied musical concepts such as "momentum," "potential energy," and "forces" [15]. Ex. A presents two interpretations of the same chord progression, where the top and bottom analysis reflects the melody and the bass respectively.

4. APPLICATIONS IN MUSIC ANALYSIS

We will demonstrate the potential usage of the proposed model in reductive analysis in three different levels of complexity.

4.1 The harmonic sequence of a hybrid theme type

For a harmonic sequence $I - V - V - I - IV - V - I$ in an 8-bar phrase, we may interpret the first four chords as a tonic prolongation and the rest as an incomplete cadential progression, as in an antecedent + cadential hybrid theme type [16]. Such an interpretation is captured by the derivation process in Fig. 3.

[1] Specifying the singleton arguments can be sometimes redundant and cumbersome. Instead, one could use the "implicit-passed" singleton by replacing the argument "Sing x" by a constraint "SingI x". In this way, the "Point" and "Unit" constructor takes no explicit argument and the state information is thus encoded as a phantom type that can be pattern matched using type application such as "Point @IV".

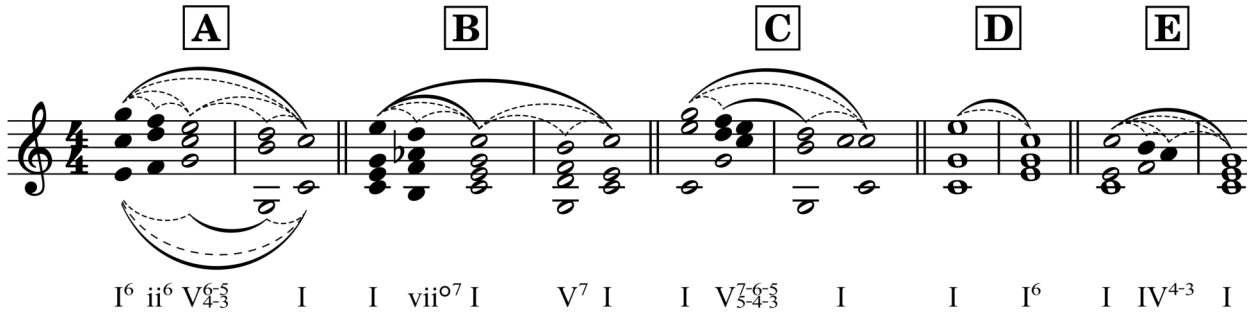


Figure 2: Examples of the graphic notation for Hierarchical processes where continuous slurs represent $Joint$ and dashed slurs represent $Link$.

$$\widehat{I}$$

$$\widehat{I} \xrightarrow{I \to I} \widehat{I}$$

$$\widehat{I} \xrightarrow{I \to I} \widehat{I} \xrightarrow{I \to V} \widehat{V} \xrightarrow{V \to I} \widehat{I}$$

$$\widehat{I} \xrightarrow{I \to V} \widehat{V} \xrightarrow{V \to I} \widehat{I} \xrightarrow{I \to IV} \widehat{IV} \xrightarrow{IV \to V} \widehat{V} \xrightarrow{V \to I} \widehat{I}$$

Figure 3: One derivation process for the harmonic sequence $I - V - V - I - IV - V - I$. The first three steps are elaboration of types whereas the last step is the instantiations of the types (in the framework of formal grammar, this corresponds to rules generating nonterminal and terminal symbols)

4.2 The key-level modulation analysis of a simple ternary form

Now we present an analysis of a section of Haydn, Piano Sonata in D, H.37, iii, using the proposed model (Fig. 4). The main focus of the analysis here is on key center and functional harmony. This overall section resembles a stationary process at the home key region. It contains a stationary process expressing the home key region, followed by a transitory process from to a home key region to another home key region. Within this transitory process, there is a stationary process at the dominant key region, as well as the two transitory processes functioning as key transition. The first connects the tonic chord in the home key $I\{I\}$ to the tonic chord in the dominant key $V\{I^6\}$ as the pivot chord for modulation. The second connects $I\{V\}$ to $I\{I\}$ as to signal the return to the home key. In a prolongational framework of pitch reduction, this link at mm.12 is the interrupted motion in a typical interruption, implying the restart of the fundamental line [8]. In a functional harmony framework, this same link is the dominant to tonic preparation relation, also implying the arrival of the tonic in the home key. Within the dominant region the first stationary process corresponds to the complete cadential progression signaling the stabilization of the dominant key, whereas the following transitory process $\xrightarrow[V\{I\} \to ||I\{V\}]{\hat{5} \to ||\hat{1}}$ corresponds to first attempted descent (interrupted) of the fundamental line accompanied by the "Ponte" modulation schema [17] that gradually change the underlying key of a chord.

4.3 A harmonic analysis of a Bach chorale

A more elaborated example of such harmonic organization can be found in Bach Chorale No 9 BWV 248 (Fig. 5). This analysis interprets the overarching structure of the piece as a stationary process in the home key tonic, containing an establishment of the home key, detour around the home key, and the re-stablization of the home key. This transitory process $\xrightarrow[I\{I\} \to I\{I\}]{\hat{1} \to \hat{8}}$ enforces its unity with an ascending linear progression of an octave, within which the music modulates to the relative minor (vi) via its dominant minor key (V/vi). The first stationary process around the home key is elaborated into a $I - V - I$ ternary-like structure on the key level.

5. CONCURRENT PROCESSES

Besides providing a formalization for processes in the temporal dimension, we also offer an extension to model the processes that are simultaneous, which enables us to model polyphonic texture.

5.1 The Formalism

To model concurrent processes, we add two **non-commutative** binary [2] operations to construct $Joint$ and $Link$ respectively:

$$(\infty) : Joint_x \to Joint_y \to Joint_{(x,y)} \tag{9}$$

$$(\Uparrow) : Link_{x,y} \to Link_{x',y'} \to Link_{(x,x'),(y,y')} \tag{10}$$

The first constructor ∞, called "**when**", expresses the concurrency of two stationary processes. The second constructor $\Uparrow$, called "**while**", expresses the concurrency of two transitory processes. Applying these operators using infix

[2] Although these operations are currently formulated as binary operations, they can be naturally extended to n-nary versions where the input is a vector of $Joint$ and $Link$ respectively.

notations, we have Eq. 11, 12:

$$\widehat{x} \infty \widehat{y} := \widehat{(x,y)} \tag{11}$$

$$\xrightarrow{x \to y} \Uparrow \xrightarrow{x' \to y'} := \xrightarrow{(x,x') \to (y,y')} \tag{12}$$

In addition we add an algebraic law (Eq. 13) on these two operations:

$$\begin{aligned} &(\widehat{x} \infty \widehat{y})(\xrightarrow{x \to x} \Uparrow \xrightarrow{y \to y})(\widehat{x} \infty \widehat{y}) \\ &= \\ &(\widehat{x} \xrightarrow{x \to x} \widehat{x}) \infty (\widehat{y} \xrightarrow{y \to y} \widehat{y}) \end{aligned} \tag{13}$$

To model temporal displacement of concurrent processes (like the suspension in fourth species counterpoint) we define a function $\uparrow_{\uparrow}$ called "**leads**" that is derivable from the basic operations in terms of Eq. 9,10 :

$$(\uparrow_{\uparrow}) : Link_{x,y} \to Link_{x',y'} \to Link_{(x,x'),(y,y')}$$

$$\xrightarrow{x \to y} \uparrow_{\uparrow} \xrightarrow{x' \to y'} = \xrightarrow{(x,x') \to (y,x')} \widehat{(y,x')} \xrightarrow{(y,x') \to (y,y')} \tag{14}$$

$$= (\xrightarrow{x \to y} \Uparrow \xrightarrow{x' \to x'})(\widehat{y} \infty \widehat{x'})(\xrightarrow{y \to y} \Uparrow \xrightarrow{x' \to y'}) \tag{15}$$

With Eq. 14, we formalize the relation about two processes where that one process **leads** the other. For example, in a typical suspension, this allows us to capture the meaning that the bass motion leads the melody motion, causing a consonance-dissonance-consonance pattern.

5.2 Concurrent processes in contrapuntal textures

Now we demonstrate that using the proposed formalism, we can model many complex hierarchical coordination of processes in both temporal (horizontal) and spatial (vertical) dimensions.

5.2.1 First species counterpoint

For first species counterpoint, processes are always vertically aligned in a one-to-one fashion.

$$\begin{aligned} &3 - 2 - 1 \\ &1 - 7 - 1 \end{aligned}$$

can be modeled as:

$$\widehat{(3,1)}\Big(\big(\xrightarrow{3 \to 2} \Uparrow \xrightarrow{1 \to 7}\big)\widehat{(2,7)}\big(\xrightarrow{2 \to 1} \Uparrow \xrightarrow{7 \to 1}\big)\Big)\widehat{(1,1)}$$

5.2.2 Second species counterpoint

For second species counterpoint, we encounter concurrent processes where one is more elaborated than the other. A segment of such texture

$$\begin{aligned} &5 - 4 - 3 \\ &7 \quad - \quad 1 \end{aligned}$$

can be modeled as

$$\widehat{(5,7)}\big((\xrightarrow{5 \to 4} \widehat{4} \xrightarrow{4 \to 3}) \Uparrow (\xrightarrow{7 \to 1})\big)\widehat{(3,1)}$$

Notice that the two-against-one coordination of the motion is reflected in the structure of the encoding and we do not need to "break" the $\hat{7}$ into two copies of $\hat{7}$ to convert it into first species texture.

5.2.3 Third species counterpoint

Third species counterpoint is a more elaborated version of the second species but the form of the representation is very similar.

5.2.4 Fourth species counterpoint

Fourth species counterpoint presents the opportunity of suspension. We can generalize such textures as overlaying transitory processes in an alternating manner, creating temporal displacement. Consider this 7-6 suspension ("=" represents "tied-over")

$$\begin{aligned} &3 = 3 - 2 = 2 - 1 \\ &5 - 4 = 4 - 3 = 3 \end{aligned}$$

It can be modeled using Eq. 14 as the following:

$$\widehat{(3,5)}\Big(\big(\xrightarrow{5 \to 4} \uparrow_{\uparrow} \xrightarrow{3 \to 2}\big)\widehat{(2,4)}\big(\xrightarrow{4 \to 3} \uparrow_{\uparrow} \xrightarrow{2 \to 1}\big)\Big)\widehat{(1,3)}$$

6. DISCUSSION

The contribution of this paper is to offer a characterization and representation on the hierarchical organization of both stationary and transitory musical processes as well as how they can be concurrently structured. Linear processes are modeled using two mutually inductive types $Joint$ and $Link$. Concurrent processes are modeled on top of linear processes by adding two binary operations for $Joint$ and $Link$ respectively. In addition, an algebraic law is imposed on these two operators to express an isomorphism between the horizontal view and the vertical view. We introduced a graphical notation for linear processes and presented several harmonic analysis using the notation to demonstrate the music analytical application of the characterization of linear processes. To demonstrate the analytical application of the concurrent processes, we presented their corresponding encoding for contrapuntal textures in species counterpoint.

This general formalism can be flexible to adapt to specific music theoretical constraints. One might encode specialized elaboration rules by equipping the $Link$ constructor with domain specific constraints on the type level and reuse the function itself. In Eq. 8, such constraint could be a predicate on the type variables $p_{x,z,y}$ [3] .

[3] Upper neighbor elaboration, for example, corresponds to the predicate $p_{x,z,y} = (x = y, z =\uparrow x)$. Likewise, downward passing elaboration corresponds to the predicate $p_{x,z,y} = (z =\downarrow x, y =\downarrow z)$.

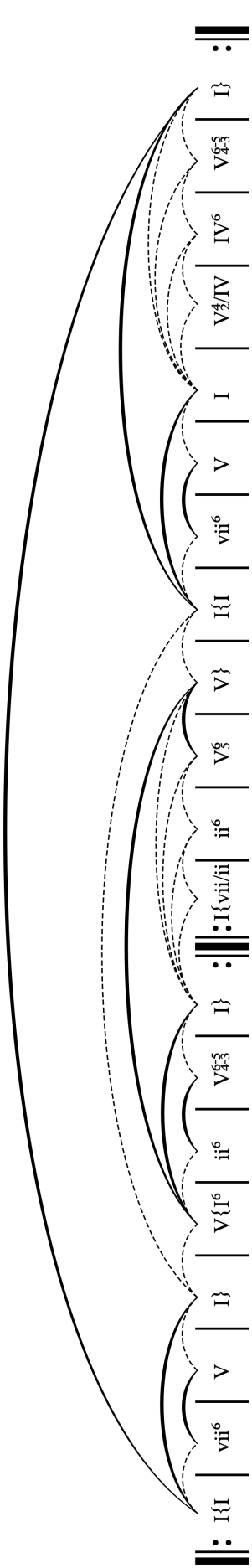


Figure 4: A prolongational analysis of Haydn, Piano Sonata in D, H.37, iii, mm. 1-20 using the notation of hierarchical process. Key regions are indicated by roman numerals followed by curly brackets.

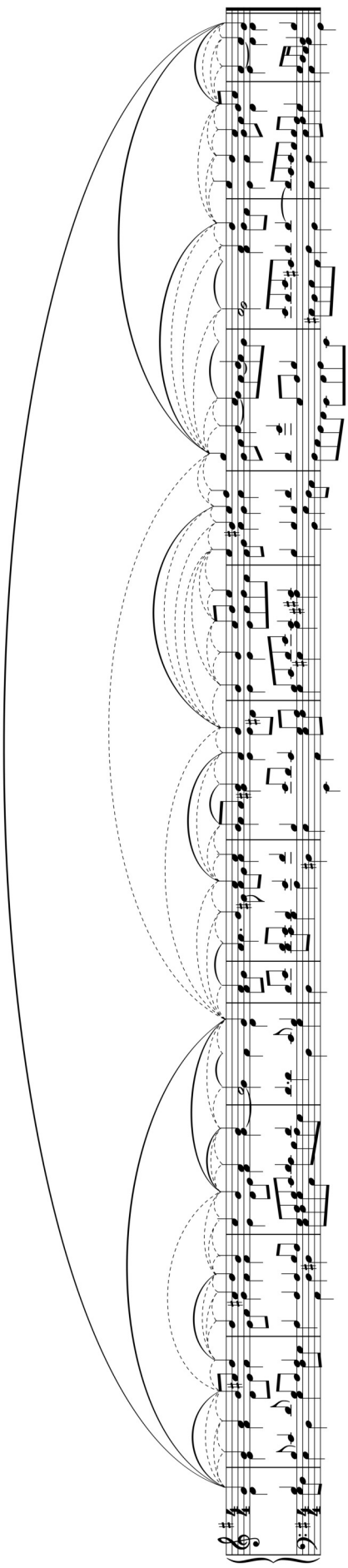


Figure 5: An analysis of Bach Chorale No 9 BWV 248.

7. ACKNOWLEDGEMENT

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 Gabriele Cecchetti and Xinyi Guan, for their helpful comments on earlier versions of this paper. Furthermore, we thank Yannis Rammos and Christoph Finkensiep for fruitful discussions on the music interpretations and implementation of the model in this paper.

8. REFERENCES

[1] M. Rohrmeier, "Towards a generative syntax of tonal harmony," *Journal of Mathematics and Music*, vol. 5, no. 1, pp. 35–53, 2011.

[2] F. Lerdahl and R. S. Jackendoff, *A Generative Theory of Tonal Music, reissue, with a new preface*. MIT press, 1996.

[3] M. Rohrmeier and F. C. Moss, "A formal model of extended tonal harmony," in *Proceedings of the 22nd International Society for Music Information Retrieval Conference*, no. CONF, 2021, pp. 569–578.

[4] 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.

[5] ——, "Towards a formalization of musical rhythm." in *ISMIR*, 2020, pp. 621–629.

[6] 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*, 2007, pp. 83–94.

[7] P. Westergaard, *An introduction to tonal theory*. Norton New York, 1975.

[8] H. Schenker, *Free Composition: Volume III of new musical theories and fantasies*. Pendragon Press, 2001, vol. 1.

[9] J. Yust, *Organized time: rhythm, tonality, and form*. Oxford University Press, 2018.

[10] C. Finkensiep, "The structure of free polyphony," p. 321, 2023. [Online]. Available: http://infoscience.epfl.ch/record/300206

[11] A. Mokhov, "Algebraic graphs with class (functional pearl)," *SIGPLAN Not.*, vol. 52, no. 10, p. 2–13, sep 2017. [Online]. Available: https://doi.org/10.1145/3156695.3122956

[12] M. Hofmann and M. Hofmann, "Syntax and semantics of dependent types," *Extensional Constructs in Intensional Type Theory*, pp. 13–54, 1997.

[13] S. Marlow *et al.*, "Haskell 2010 language report," *Available online http://www. haskell. org/(May 2011)*, 2010.

[14] R. A. Eisenberg and S. Weirich, "Dependently typed programming with singletons," *ACM SIGPLAN Notices*, vol. 47, no. 12, pp. 117–130, 2012.

[15] S. Larson, *Musical forces: Motion, metaphor, and meaning in music*. Indiana University Press, 2012.

[16] W. E. Caplin, *Analyzing classical form: An approach for the classroom*. Oxford University Press, USA, 2013.

[17] R. Gjerdingen, *Music in the Galant Style*. OUP USA, 2007.

ONLINE SYMBOLIC MUSIC ALIGNMENT WITH OFFLINE REINFORCEMENT LEARNING

Silvan David Peter
Institute of Computational Perception, Johannes Kepler University Linz, Austria
silvan.peter@jku.at

ABSTRACT

Symbolic Music Alignment is the process of matching performed MIDI notes to corresponding score notes. In this paper, we introduce a reinforcement learning (RL)-based *online* symbolic music alignment technique. The RL agent — an attention-based neural network — iteratively estimates the current score position from local score and performance contexts. For this symbolic alignment task, environment states can be sampled exhaustively and the reward is dense, rendering a formulation as a simplified offline RL problem straightforward. We evaluate the trained agent in three ways. First, in its capacity to identify correct score positions for sampled test contexts; second, as the core technique of a complete algorithm for symbolic online note-wise alignment; and finally, as a real-time symbolic score follower. We further investigate the pitch-based score and performance representations used as the agent's inputs. To this end, we develop a second model, a two-step Dynamic Time Warping (DTW)-based *offline* alignment algorithm leveraging the same input representation. The proposed model outperforms a state-of-the-art reference model of offline symbolic music alignment.

1. INTRODUCTION

Music alignment refers to matching at least two different versions of the same musical material. In this paper, we address symbolic music alignment, for our purposes defined as models that match individual notes of a performance recorded as MIDI file to individual notes of a score encoded as MusicXML file.

Alignment procedures can be separated into online or offline classes. If the alignment procedure is carried out with access to the full versions of the musical material, we refer to it as offline alignment. Conversely, if one version is only known up to the point currently to be matched, we refer to it as online.

We introduce a reinforcement learning (RL)-based *online* symbolic music alignment technique. It aligns symbolically encoded music or, more specifically, MIDI performances to their corresponding MusicXML scores by matching individual notes of each version. The RL agent – a small attention-based neural network – is trained to iteratively predict the current score position from limited score and past performance contexts. The current performance note and estimated score position are then processed to compute a symbolic note-wise matching.

RL terminology introduces another online versus offline differentiation. RL is termed online if the agent learns from data created by the agent's interaction with its environment during training. In our case, we use offline RL, that is, the agent is trained using a dataset of exhaustively sampled environment states and associated rewards, effectively turning agent training into a supervised learning problem. Once trained in an offline fashion, the agent can be used in online alignment.

The agent processes a purely *pitch-based representation* and timing information is only incorporated in a post-processing step. Before addressing the online problem, we investigate the same separation of pitch and time processing in an offline setting: we develop a two-step (first pitch, then time) Dynamic Time Warping (DTW) offline model and evaluate it against the state of the art in note-wise alignment in Section 3. The subsequent Section 4 addresses the RL-based online model reusing the input setup.

The rest of this paper is thus structured as follows: Section 2 introduces related work. Section 3 discusses offline symbolic music alignment. We develop as well as evaluate an offline symbolic music alignment algorithm based on two different applications of DTW, first on pitch information, then on onset times. Starting from these results, section 4 introduces a formulation of online alignment as reinforcement learning problem. In particular, we train an agent's value function in an offline setting. In section 5 we evaluate the trained agent in three ways: as a standalone score onset identification model, as an online symbolic alignment model (where the aim is the production of correct note-wise alignments), and in a score following scenario (where the aim is the precise temporal tracking the current score position). Finally, Section 6 concludes the paper with a critical appraisal of our models as well as recommendations for future research.

2. RELATED WORK

Symbolic music alignment has been a popular research area for many years. We begin our review of related work with online symbolic music alignment, then we progress

© S. Peter. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** S. Peter, "Online Symbolic Music Alignment with Offline Reinforcement Learning", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.*, Milan, Italy, 2023.

to offline symbolic music alignment, general music alignment, and finally applications of reinforcement learning.

Most often, online models have been presented in the context of score following, where the principal aim is to identify the current score position. Dannenberg [1] and Vercoe [2] pioneered this area of research in the mid 1980s. Recent works commonly use Dynamic Bayesian Networks to track the performance [3–6]. Recently, A. Anonymous compared both Hidden Markov Models (HMM) and On-Line Time Warping (OLTW) techniques. We use their OLTW model as comparison baseline for our online alignment technique. This model processes inputs represented as piano rolls, as is common for OLTW and DTW applications to symbolic music alignment in general.

The offline setting has seen more recent work [3, 5, 7–11]. Symbolic music alignment methods often perform very well, with error rates rarely exceeding 10%. Consequently, much recent work focused on the rare, indeterminate, or asynchronous events that make the errors difficult to identify and fix. Ornaments are one source of such events [9]. Another is left-right hand asynchrony in piano performance as discussed in Nakamura et al. [3]. Arbitrary skips and repeats further present a very difficult challenge for most algorithms, especially when runtime considerations are important [10]. This series of articles by Nakamura et al. [3, 9, 10] culminated in one of the most widely used automatic score-performance alignment tools and the current state of the art (SOTA) [11]. We use this model as reference for the evaluation of our offline algorithm.

Although beyond the scope of this article, no introduction of music alignment is complete without the mention of the large body of work concerning alignment of non-symbolic music formats, in particular audio. Wang [12], Arzt [13], and chapter three in Müller [14] present introductory discussions of audio alignment. As in our offline approach, applications of (non-standard) DTW are central to audio alignment [15–18]. Audio score following is commonly computed using On-Line Time Warping and variants of Hidden Markov Models [19–23].

To the best of our knowledge, Dorfer et al. [24] (later expanded upon by Henkel et al. [25]) are the only prior application of RL in a music alignment task, namely online audio to sheet music image alignment. For a general introduction to RL, we refer the reader to Sutton and Barto [26], for a discussion of the merits and disadvantages of offline RL to Levine et al. [27].

3. OFFLINE SYMBOLIC MUSIC ALIGNMENT

In this section, we introduce an offline symbolic music alignment based on two different DTW steps as well as an intermediate cleanup step. We close the section with an evaluation of our model against a state-of-the-art reference.

Symbolic music alignment produces *note alignments*, i.e., it matches individual notes of a performance recorded as MIDI file to individual notes of a score encoded as MusicXML file. Three types of note alignments exist: a match is tuple of a performance note and a score note, a deletion is a score note not played, and an insertion is a performed note not notated.

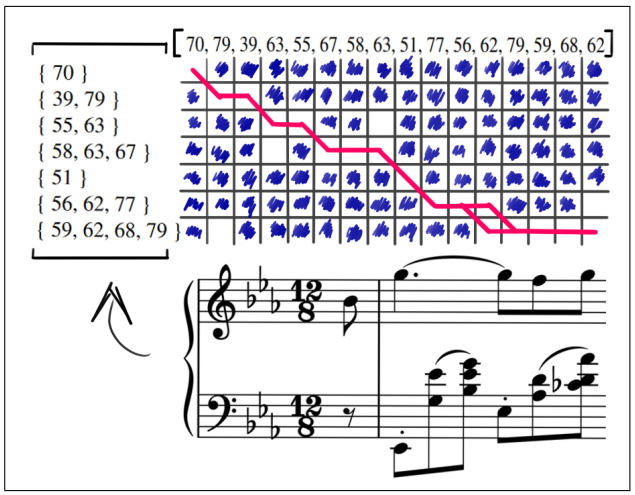


Figure 1. First half measure of Chopin Op. 9 No. 2 (bottom score), encoded as pitch set sequence (left) and warped to its performance, encoded as sequence of pitches as played (top). The matrix shows the corresponding pairwise distance (shaded is distance of 1, see equation 1), red lines indicate equivalent optimal warping paths.

Our proposed offline algorithm consists of the following steps: First, the performance and score are aligned using DTW on a purely pitch-based representation (Section 3.1). Then, remaining gaps are filled by complete sequences of a single pitch. Finally, individual notes are aligned using an application of DTW on their onset times (Section 3.2).

3.1 Pitch Sequence Warping

In this approach, we align sequences of performance notes, encoded as integer pitches $p_t \in \mathbb{I} := \{1...88\}$, with sequences of score onset notes, encoded as sets of integer pitches $s_t \in \mathcal{P}(\mathbb{I}) \setminus \{\varnothing\}$, with $\mathcal{P}(\mathbb{I})$ denoting the power set of the set $\mathbb{I}$. Since these sequence elements are of different types — integers and sets of integers — no standard local distance metric can be used. Instead we opt for a non-symmetric inclusion metric, with some abuse of the term.

$$m(p_t, s_t) = \begin{cases} 0 & \text{if } p_t \in s_t \\ 1 & \text{else} \end{cases} \tag{1}$$

Having defined the metric in Equation 1, two standard DTW paths are computed, one forward and one backward, i.e., using inverted sequences. Figure 1 shows the encoding as well as examplary DTW paths computed from cumulative pairwise distances. While the optimal DTW distance is unique, the DTW paths are not necessarily so. In our case, such ambiguity is often introduced by repeated pitches in neighboring score onsets, see e.g., the two adjacent (left, stacked vertically) notes of pitch D4/62 in Figure 1. To pinpoint non-robust path segments, we use the backward DTW path. Wherever the forward and backward paths disagree, they effectively bracket ambiguous parts from both sides, and we exclude all bracketed notes from the path.

These excluded segments are then processed using a simple heuristic: The notes in bracketed segments are sep-

arated by pitch. If two pitch-wise sequences with matching number of notes (in the performance and the score) are found, they are aligned and the result is added to the path. If no matching sequence is found, the path is linearly interpolated. We finally compute a mapping from score time to performance time from this merged and cleaned path.

3.2 Onset Sequence Warping

The next goal is to derive note-wise alignments from the approximate score to performance mapping computed in Section 3.1. To this end, the performance and score are split into pitch-wise sequences for each pitch occurring in the union of score and performance pitches. The approximate score-time-to-performance-time mapping computed in the previous step is used to project all score onsets (in beats) to performance time points (in seconds). The last step aligns the performance onset sequence with the score onset sequence, now mapped to the same space.

This alignment is computed by a DTW path between the onset sequences, this time using a simple L_1 metric, and for each non-unique alignment, keeping the tuple with the lowest distance. A threshold of maximal distance is further built-in (and set to 5 seconds) to avoid spurious alignment of unrelated deletions and insertions.

3.3 Offline Model Evaluation

To test the full offline model (first pitch DTW 3.1, then onset DTW 3.2), we compute alignments on four datasets of high-quality note-wise alignment piano music. The datasets are the Vienna 4x22 Dataset [28], four excerpts performed by 22 performers each; the Zeilinger Dataset [29], nine piano sonatas by Ludwig van Beethoven performed by Clemens Zeilinger; the Magaloff Dataset [30], the near complete solo piano works by Frederic Chopin performed by Nikita Magaloff; and the Batik Dataset [31], twelve piano sonatas by Wolfgang Amadeus Mozart performed by Roland Batik. We compare our model against the reference by Nakamura et al. [11], post-processed to produce the same output format we employ.

To compare produced alignments to ground truth ones, we have to define a metric. Recall that note alignments consist of three types: matches (tuples of performance and score notes), deletions (unplayed score notes), and insertions (unnotated performed notes). We use an F-score metric for matches: A predicted match is counted as a true positive (TP) only if the same notes are matched in the ground truth alignment. A false positive (FP) is a predicted note label that isn't in the ground truth, a false negative (FN) is a ground truth note label that isn't predicted. The F-score is defined as the harmonic mean of precision (TP / (TP + FP)) and recall (TP / (TP + FN)).

Table 3.3 shows dataset-wise and globally averaged F-scores for matches. Our proposed model outperforms the reference on each dataset. A two-sided sign test on performance-wise rankings shows significantly ($\alpha = 0.01$) higher performance for our proposed model on all datasets except the Vienna 4x22 Dataset. On the Vienna 4x22 dataset, the models reach the same F-score of 1.0 for 38 performances and our proposed model has higher F-scores for the remaining 50 performances.

Dataset	DTW Offline	Nakamura
Magaloff	98.4 ± 0.9 %	97.8 ± 1.4 %
Zeilinger	99.3 ± 0.9 %	98.8 ± 1.2 %
Batik	99.4 ± 0.7 %	98.5 ± 2.1 %
Vienna 4x22	99.8 ± 0.4 %	99.5 ± 0.5 %
Combined	99.0 ± 1.0 %	98.5 ± 1.5 %

Table 1. Dataset-wise averaged F-scores and standard deviations of each model.

4. ONLINE ALIGNMENT AGENT

Having established the effectiveness of the separation into pitch-based and time-based input representations in the offline setting, we now introduce a formulation of RL-based online alignment. We continue with the model and training setup used to approximate the agent's value function.

Reinforcement learning is formalized as Markov Decision Process (MDP). An MDP consists of the following components: a state space $\mathcal{S}$, an action space $\mathcal{A}$, an index set $\mathcal{T}$, a reward function $\mathcal{R}$, transition probabilities $\mathcal{P}$, and a discount factor γ.

An agent is placed in an environment and perceives this environment and itself as being in a possible state $S_t \in \mathcal{S}(t \in \mathcal{T})$. The agent now takes an action $A_t \in \mathcal{A}$ and receives a reward R_{t+1} as well as a new state $S_{t+1} \in \mathcal{S}$. Repeating this process iteratively yields a sequence of states, actions, and rewards, called an episode: $S_t, A_t, R_{t+1}, S_{t+1}, A_{t+1}, R_{t+2}, S_{t+2}, A_{t+2}, ...$ It is now the agent's task to infer actions from states that maximize long-term reward. Before we look at our formulation of this optimization problem, we discuss the online alignment's state and action space in more detail.

4.1 Alignment as Reinforcement Learning

The state information S_t comprises both the current score context as well as the most recent past performance. Specifically, the score context is represented as a window of the pitch set sequence introduced in Section 3.1. The window centers the last predicted score onset position and spans seven score onsets to the past as well as eight score onsets to the future for a total windowed sequence of 16 pitch sets. The performance context is only derived from past performance notes to enable real-time application. It consists of the eight most recent notes in the performance pitch sequence. Whenever less score or performance context is available, e.g., at the very beginning or end of a piece, the windows are shortened accordingly.

At each state S_t the agent aims to match the most recent performance note to its most likely score onset. There are 16 actions at S_t; select one score onset as matching onset position. Having decided on a next score onset, the agent receives a reward R_{t+1} which is set to one if the score onset is correctly aligned, zero otherwise. The environment transition probabilities $\mathcal{P}$ determine a new state S_{t+1}: the

performance context of the new state is determined by an actual performance, i.e., the agent is presented a new state based on the estimated next position in the score and a new incoming performance note.

4.2 Simplified Deep Q-learning

The agent's behavior is captured by its policy $\pi(A|S)$, the distribution of actions taken by the agent in state S. Although it is possible to optimize the policy directly, we instead adapt a value-function-based formulation, or more specifically, deep Q-learning [32, 33]. Q-learning aims to optimize a state-action value function $Q : \mathcal{S} \times \mathcal{A} \longrightarrow \mathbb{R}$, an estimate of the expected cumulative discounted future reward, also called return, given a state and an action. In deep Q-learning the value function $Q(S, A, \theta)$ contains trainable parameters θ which are fitted to the experienced reward distribution. A typical optimization loss l looks like this:

$$l = (R_{t+1} + \gamma \max_{A_{t+1}} Q(S_{t+1}, A_{t+1}, \theta) - Q(S_t, A_t, \theta))^2 \tag{2}$$

where the discount factor $\gamma \in [0, 1]$ determines the weighting of future rewards. For the alignment case we can make several simplifications. We opt for a completely myopic agent, i.e., $\gamma = 0$. The argument for this is that the optimal action to take for each incoming performance note is determined by the correct score onset which can in turn be specified by the immediate reward. >For a discussion of the implications of this modeling choice see section 6. Setting $\gamma = 0$ removes the value function at subsequent states from the loss. Using that $\mathcal{R} = \{0, 1\}$, we make a second reformulation and replace this squared error loss by a binary classification: For each state-action tuple (S, A) the agent predicts the probabilities of the reward being 1 or 0, optimized with a cross-entropy loss. Note that this formulation still optimizes a state-action value function Q and not a policy $\pi(A|S)$, i.e., the probabilities of rewards are computed for each possible action (that is, per score onset) and do not sum to one over all actions. There are several ways of deriving a policy from a value function; two are discussed in section 4.5.

4.3 Value Function Model

To approximate $Q(S, A, \theta)$, we use an attention-based Transformer Neural Network. The input of the network consists of a sequence of tokens encoding the performance, a delimiter token, the score, and an ending token. We encode 88 pitches of the piano keyboard, adding extra tokens for "no_pitch", "delimiter", and "end" in a 64-dimensional embedding space. The performance pitches are straightforward to embed, however, the score onset pitch sets require more processing. For our data, more than 99% of score onsets can be represented with pitch sets with no more than seven different pitches, we thus limit our pitch sets to this length (with a subset of seven taken randomly at onsets with more pitches). Pitch sets with fewer than seven pitches are filled up with a pitch corresponding to the

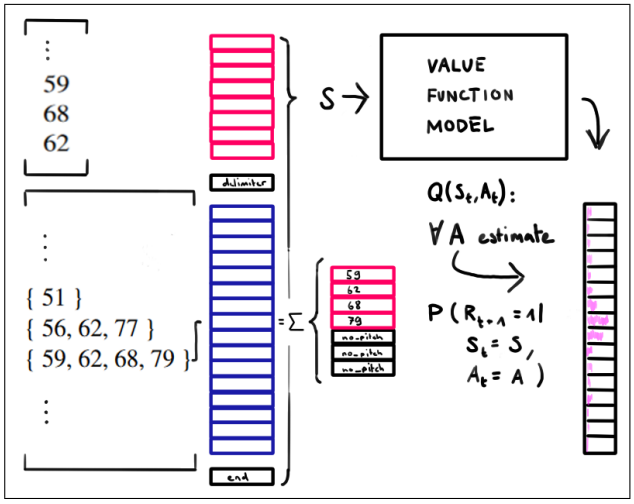


Figure 2. Setup of the value function model: states are encoded as contiguous token sequence of past performance (red) and current score (blue) contexts. Pitch set embeddings are summed over individual pitch embeddings. The model is set up as token classifier as each score onset in the context corresponds to a possible action (= "select this onset as next score onset") and is classified according to its expected reward class. The vector on the right shows the reward probability for each action (pink).

"no_pitch" token. To create score onset embedding with no more than 64 dimensions independent of the number of pitches at any onset, the pitch set tokens are summed up. Figure 2 shows the setup of the value function model. The model is set up with eight heads, and six layers, layer normalization, and a single feedforward head for binary classification, making for a total of 157250 parameters.

4.4 Training

The training is set up as token classification problem; i.e., for each token in the sequence, the probability of receiving a reward is estimated. The aligned piano datasets from Section 3.3 are used again. For our offline RL setting, a dataset of states is created before training. We extract local score and performance contexts from the aligned data, shifting the performance window such that true next score onsets fall from leftmost (current position minus seven) to rightmost (current position plus eight) to cover the possible states exhaustively. During training, batches of states are sampled randomly, not in sequence. To aid generalization, we further augment the data by random pitch shifting of all notes in the state within +/- one octave. We use an ADAM optimizer with a learning rate with warm-up followed by square root decay. The batch size is set to 8192, the models are trained for 50 epochs.

4.5 Online Models

Using a trained value function model, we derive two complete models. First, a simple score follower model ("Greedy Agent Model") that outputs only greedily estimated score onsets for incoming performance notes. Sec-

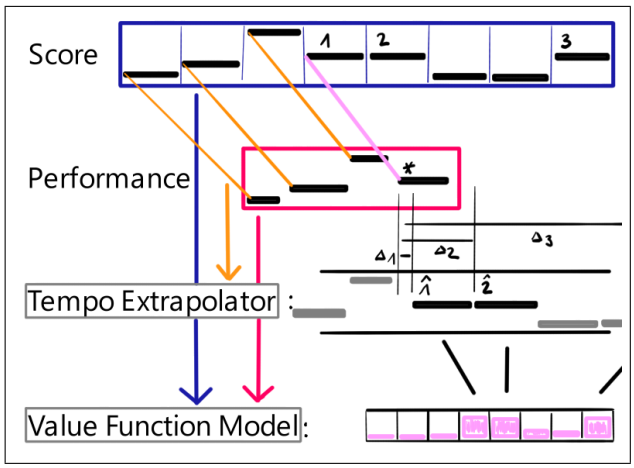


Figure 3. Schematic overview of the Online Alignment Model with a monophonic piece and 8 onset context. Score (blue, top) and performance (red) contexts are inputs to the Value Function Model which outputs value estimates (pink, bottom). The top three onsets (1,2,3) are passed to a tempo extrapolator, along with existing alignments (yellow). The tempo extrapolator predicts three onsets $(\hat{1}, \hat{2})$ for the the candidate onsets. The one with lowest distance (Δ_1) to the newest performance note (*) is aligned (pink).

ond, a note-level alignment model ("Online Alignment Model") that produces both note alignments as well as an estimate of the next score onset for the score following setting. The Greedy Agent Model consists of an agent following a greedy policy based on the trained value function model, i.e., an agent picking the action A with maximal estimated reward $Q(S, A)$.

For the Online Alignment Model a few additional steps are taken. See Figure 3 for an overview of the alignment loop. In this model, an action is selected from the top three value estimates for a given score and performance context. To pick one of these three actions, onset time information is incorporated. A simple local tempo estimator approximates an expected performed onset time for each of the three possible score onsets using linear extrapolation of beat periods computed from previously aligned notes. This process takes on the role of the second onset-wise DTW step in the offline model (see Section 3.2): to match notes that are closest together according to an approximate score-to-performance mapping.

There are two further heuristics worth mentioning. If the current performed pitch is not available at any of the three highest ranking score positions, the performed note is counted as an insertion, and the current score position is unchanged. Furthermore, we decrease the number of calls made to the agent in a real-time setting by directly aligning pitches that are trivially missing at the current score onset.

5. ONLINE EVALUATION

In the following, we evaluate the agent and the proposed online alignment model. In section 5.1, we address a greedy agent's capacity to identify correct score positions for sampled test contexts. In section 5.2, the Online Alignment Model is evaluated with respect to correct note-wise alignment. Finally, we use both the Greedy Agent Model as well as the Online Alignment Model as real-time symbolic score followers in an experiment in section 5.3.

5.1 Agent Evaluation

Top0	Top1	Top2
94.5 ± 0.8 %	96.6 ± 0.5 %	97.6 ± 0.4 %

Table 2. Average topK score onset hit rate and standard deviation across the five test folds.

For direct value function evaluation, we assume a greedy policy for each testing state S. That is, the agent picks the action A with the highest estimated value $Q(S, A)$. We evaluate this action (= chosen score onset) via the distance from the ground truth score onset. Specifically, we compute three metrics: the number of states where this action corresponds exactly to the true score onset ("Top0"), the number of times this action picks a score onset in the neighborhood of ± one score onset of the true location ("Top1"), and the number of times this action picks a score onset in the neighborhood of ± two score onsets of the true location ("Top2"), each normalized by the total number of test states. We use five-fold cross-validation on the same combined datasets used in section 3.3, and report mean and standard deviation values across testing folds. The fold splitting is carried out piece-wise with roughly the same number of score onsets in each fold.

Table 2 shows the results. Greedy action selects the correct score onset with more than 94 % probability on unseen pieces. Furthermore, for more than half of the remaining errant actions, the greedy action is not further than two onsets from the correct one.

5.2 Online Note-wise Alignment

Piece	OAM	DTW Offline	Nakamura
B. Op. 53 3rd. m.	99.0 %	99.4 %	98.2 %
C. Op. 9 No. 1	97.6 %	98.4 %	98.8 %
C. Op. 9 No. 2	97.4 %	99.1 %	97.6 %
C. Op. 10 No. 11	90.3 %	96.3 %	94.3 %
C. Op. 60	95.1 %	97.9 %	94.7 %

Table 3. Piece-wise F-scores of each model. OAM = Online Alignment Model, DTW Offline = model of section 3.3, Nakamura = reference SOTA model [11].

To evaluate the Online Alignment Model's performance, we perform alignments for five selected performances: Nocturnes Op. 9 No. 1 and 2, Etude Op. 10 No. 11, Nocturne Op. 15 No. 2, the Barcarole Op. 60 by F. Chopin, and the third movement of the Sonata Op. 53 (Waldstein) by L. v. Beethoven. The value function model used in this section was trained on all data except these five pieces for 100 epochs, the rest of the training and model setting remains the same. The same metrics of section 3.3

Model	Async	≤ 25ms	≤ 50ms	≤ 100ms
OLTW	60.6 ms	38.0 %	63.3 %	86.7 %
GAM	36.0 ms	89.0 %	91.4 %	94.6 %
OAM	15.7 ms	91.4 %	93.8 %	96.6 %

Table 4. Asynchrony of the models in score follower setting. Column "Async" presents the median asynchrony. Columns 3, 4, 5 present the percentage of onset estimates with lower asynchrony than 25ms, 50ms and 100ms, respectively.

apply, namely the F-score of retrieved matched note tuples. For comparison we also add piece-wise F-scores of our proposed offline model as well as the model by Nakamura et al. Table 3 shows the F-scores of note alignments. As expected, the proposed online alignment performs worse than offline methods, albeit with small difference. Notably, all models show the lowest performance on Chopin's Op. 10 No. 11.

5.3 Score Following

In the score following setting, the core metric is the accurate prediction of the current position. We thus compute asynchrony values in milliseconds which give the absolute time between any onset in a performance and the onset in the same performance that corresponds to the estimated score onset. The data used for this experiment consists of the same five pieces used in the previous section 5.2 with the same value function model training. Three models are compared in this setting: The Online Alignment Model (OAM) previously evaluated in terms of note alignment F-scores in Table 3, the Greedy Agent Model (GAM), and an On-Line Time Warping (OLTW) Model. This latter OLTW model performed best in a recent music score following comparison by Cancino-Chacón et al. [34] and is added as a reference. However, this model does not predict note alignments, hence a comparison in terms of note alignment F-score as in Section 5.2 is not possible.

Both the GAM and the OAM outperform the reference model in all metrics in Table 4. Most of the lower performance of the GAM is due to the fact that for Chopin's Op. 10 No. 11, this agent loses track of the performance close to the end when a full measure is deleted. All subsequent performance notes are estimated very wrongly. The online alignment model on the other hand follows all test performances robustly until the end.

6. DISCUSSION AND CONCLUSION

In this paper, we introduce two models, an offline alignment model based on dual DTW steps, and an online alignment model based on an RL agent trained in an offline fashion. Both models perform competitively; with the offline model surpassing the relevant state of the art.

In the setup of the RL training we made some simplifications that warrant further discussion. Specifically, we set the discount factor γ to zero and train using a dataset of sampled states. In section 4.2, we claim that the optimal action for each step is determined by the correct score onset. While this is true for the states in the dataset and if optimality is defined by accuracy in note-wise alignment, it might not be for out-of-distribution states or if the focus of the agent is shifted to robustness, i.e., following the entire performance even at the cost of some misaligned notes.

For offline RL, a crucial issue is distributional drift [27]; i.e., the fact that the agent learns from states that follow a different distribution that the states it would encounter in an online setting. Even though we can sample the state space exhaustively for the training set, out-of-distribution states are expected in test sets consisting of previously unseen pieces and performances. Furthermore, the relative frequency of training samples does not necessarily correspond to the states an online agent is likely to see during training, where all target locations have the same frequency. Specifically, for an agent that already learned to predict the score onset with some accuracy, the targets at the limits of the context are going to be less frequent than the center ones. In other words, a non-myopic online agent is likely to behave more conservatively, avoiding large skips as they do not occur that frequently in actual performances.

On the other hand, the offline RL formulation successfully leverages prior knowledge about the task and — more importantly — stabilizes the gradient, rendering the training of a complex value function approximator feasible. Future work includes shifting this trade-off back towards online RL, for example with online RL training after initial offline training.

The RL agent learns to align purely on pitch information. Including onset or even duration information is likely to increase the accuracy of following at the cost of requiring a more expressive model which in turn affects inference speed — a hard bottleneck for real-time application. In fact, running the value estimation for every incoming performance note (such as the score follower "GAM" in Table 4) uses up to a minute of computation time for the 7273 notes in the performance of Beethoven's Op. 53 Mvt. 3 (Roughly 10 ms per note). A further increase is liable to affect real-time score following in fast passages.

In terms of post-processing steps, both our offline and online models are comparatively crude, making little use of score information such as ornaments. As Nakamura et al. [11] correctly remark, their post-processing step is in principle able to improve upon any prior more error-prone alignment. Further research is needed to know whether the offline model can be improved in this way.

To conclude, we developed and evaluated two models of symbolic music alignment which both outperform relevant prior work. To the best of our knowledge, this RL-based online alignment model is one of the first applications of not only trainable but effectively *trained* models to symbolic music alignment.

7. REPRODUCIBILITY

Our alignment models are available at: `https://github.com/sildater/parangonar`

8. 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?").

9. REFERENCES

[1] 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.

[2] B. Vercoe, "The Synthetic Performer in the Context of Live Performance," in *Proceedings of the International Computer Music Conference (ICMC)*, 1984, pp. 199–200.

[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] C. Raphael and Y. Gu, "Orchestral Accompaniment for a Reproducing Piano," in *International Conference on Mathematics and Computing*, 2009.

[5] 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.

[6] A. Maezawa, H. G. Okuno, T. Ogata, and M. Goto, "Polyphonic audio-to-score alignment based on bayesian latent harmonic allocation hidden markov model," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2011, pp. 185–188.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] S. Wang, "Computational Methods for the Alignment and Score-Informed Transcription of Piano Music," Ph.D. dissertation, Queen Mary University of London, London, UK, 2017.

[13] A. Arzt, "Flexible and robust music tracking," Ph.D. dissertation, Johannes Kepler University Linz, Linz, Austria, 2016.

[14] M. Müller, *Fundamentals of Music Processing – Audio, Analysis, Algorithms, Applications*. Springer, 2015.

[15] T. Prätzlich, J. Driedger, and M. Müller, "Memory-restricted multiscale dynamic time warping," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 569–573.

[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] C. J. Tralie and E. Dempsey, "Exact, parallelizable dynamic time warping alignment with linear memory," in *21st International Society for Music Information Retrieval Conference (ISMIR)*, 2020.

[18] S. Ewert, M. Muller, and P. Grosche, "High resolution audio synchronization using chroma onset features," in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2009.

[19] S. Dixon, "An On-Line Time Warping Algorithm for Tracking Musical Performances," in *International Joint Conference on Artificial Intelligence*, 2005, pp. 1727–1728.

[20] A. Cont, "ANTESCOFO: Anticipatory Synchronization and Control of Interactive Parameters in Computer Music." in *Proceedings of the International Computer Music Conference (ICMC)*, 2008, pp. 33–40.

[21] C. Raphael, "Music Plus One and Machine Learning," in *International Conference on International Conference on Machine Learning*, 2010, pp. 21–28.

[22] 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.

[23] A. Arzt and G. Widmer, "Real-Time Music Tracking Using Multiple Performances as a Reference," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2015.

[24] 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, Sep. 2018, pp. 784–791. [Online]. Available: https://doi.org/10.5281/zenodo.1492535

[25] 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*, vol. 2, no. 1, 2019.

[26] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[27] S. Levine, A. Kumar, G. Tucker, and J. Fu, "Offline reinforcement learning: Tutorial, review, and perspectives on open problems," *arXiv preprint arXiv:2005.01643*, 2020.

[28] W. Goebl. (1999) The Vienna 4x22 Piano Corpus. [Online]. Available: http://repo.mdw.ac.at/projects/IWK/the_vienna_4x22_piano_corpus/index.html

[29] 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.

[30] S. Flossmann, W. Goebl, 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.

[31] G. Widmer and A. Tobudic, "Playing mozart by analogy: Learning multi-level timing and dynamics strategies," *Journal of New Music Research*, vol. 32, no. 3, pp. 259–268, 2003.

[32] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, pp. 279–292, 1992.

[33] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[34] 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*, 2023.

INVERSYNTH II: SOUND MATCHING VIA SELF-SUPERVISED SYNTHESIZER-PROXY AND INFERENCE-TIME FINETUNING

Oren Barkan[1] **Shlomi Shvartzman**[2] **Noy Uzrad**[2]
Moshe Laufer[2] **Almog Elharar**[2] **Noam Koenigstein**[2]

[1]The Open Univesity of Israel
[2]Tel Aviv University, Israel

ABSTRACT

Synthesizers are widely used electronic musical instruments. Given an input sound, inferring the underlying synthesizer's parameters to reproduce it is a difficult task known as *sound-matching*. In this work, we tackle the problem of automatic sound matching, which is otherwise performed manually by professional audio experts. The novelty of our work stems from the introduction of a novel differentiable *synthesizer-proxy* that enables gradient-based optimization by comparing the input and reproduced audio signals. Additionally, we introduce a novel self-supervised finetuning mechanism that further refines the prediction at inference time. Both contributions lead to state-of-the-art results, outperforming previous methods across various metrics. Our code is available at: `https://github.com/inversynth/InverSynth2`.

1. INTRODUCTION AND RELATED WORK

Sound synthesis has been an active research field since the end of the previous century [1]. Given a query audio input, the task of crafting a specific sound is known as *sound matching*. Synthesizer sound matching, also known as *inverse synthesis*, involves carefully tuning parameters from an exponentially large number of possible configurations- a task mostly reserved for professional audio experts. This paper presents a novel algorithmic approach for *automated sound matching*.

Algorithmic approaches for inverse synthesis can be loosely categorized into *search-based* methods and *modeling-based* methods [2]. Search-based methods often utilize genetic algorithms (GA) which are based on principles of Darwinian evolution to determine the optimal synthesizer configurations. For instance [3] initiated a set of randomly sampled configurations and used GA optimization to reconstruct the original audio signal. Other search-based methods include Particle Swarm Optimization (PSO) [4] and Hill-Climbing [5]. Search-based methods can employ different objectives such as mel-frequency cepstral coefficients (MFCCs) or a combination of multiple objectives [6, 7]. However, optimizing configurations through search-based methods can be both resource-intensive and time-consuming for every sound sample. Consequently, the rise of deep learning techniques has led to a shift from search-based methods to model-based ones, which avoid the previously mentioned drawbacks. However, search-based methods still possess a unique advantage: they can establish a loss term that directly contrasts the reconstructed audio signal with the input signal.

The aforementioned advantage is absent in most modeling-based methods, as they usually cannot propagate gradients through an external, commercial synthesizer. As a result, they depend on setting an optimization goal focused on reconstructing the parameters rather than the reproduced signal. In general, modeling-based methods employ deep learning in order to predict a synthesizer's configuration based on the input audio signal. For example, [8] employed long short-term memory (LSTM) networks for predicting the parameters in FM synthesizers. InverSynth (IS) [9] employed strided convolution neural networks (CNNs) to estimate a synthesizer's parameters as a multi-objective classification problem. When compared to the LSTMs approach of [8], IS provides improved accuracy with the ability to scale for longer audio sequences. Another direction involves employing variational inference [10] and normalizing flows [11, 12] to automatically tune an open-source replica of the Yamaha DX7 synthesizer [13]. Finally, [14–16] introduce a different versions of audio synthesizer models for sound matching.

A completely different direction for sound matching and synthesis is through neural synthesizers [17–22]. For example, in [19] the authors train Generative Adversarial Networks to synthesize sounds that simulate natural audio samples. However, these directions are inherently different from the current line of work, as they do not deal with the problem of tuning existing musical synthesizers. Instead, these works suggest alternatives to familiar synthesizers, which may be useful for future applications but are less relevant to mainstream musicians that use existing commercial synthesizers.

In this paper, we present InverSynth II (IS2) - an innovative inverse-synthesis model that introduces a differentiable synthesizer-proxy capable of learning to "imitate" the behavior of any given synthesizer. This allows for a differentiable relationship between the synthesizer's parameters and the produced audio signal. As a result, IS2 learns to focus on the synthesizer parameters that have more impact on the reproduced signal. Our evaluations indicate that this approach leads to a better reconstruction of the

© Barkan et al.. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Barkan et al., "InverSynth II: Sound Matching via Self-Supervised Synthesizer-Proxy and Inference-Time Finetuning", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

original audio signal in terms of spectral loss and human perception.

Our contributions are as follows: (1) We introduce IS2 that effectively incorporates the synthesizer's functionality into the computational graph. By learning a differentiable *synthesizer-proxy*, IS2 facilitates self-supervision based on the difference between the input and reproduced audio signals. This is in contrast to previous model-based works that optimized on the predicted synthesizer parameters alone [8, 9, 23]. (2) We introduce a novel self-supervised finetuning technique that utilizes the learned synthesizer-proxy to further refine predictions at inference time. (3) We compare IS2 against the state-of-the-art methods from [10] and [9] on the three datasets, including the datasets from both of these works. Our findings show that IS2 outperforms both methods on all datasets, across all metrics.

2. INVERSYNTH II

2.1 Problem Setup

Let $x \in \mathcal{X}$ be the audio signal i.e., raw waveform, Short-time Fourier transform (STFT) spectrogram, etc. Let $f : \mathcal{Y} \rightarrow \mathcal{X}$ be a synthesizer function that generates a signal $f(y) \in \mathcal{X}$ according to the parameters configuration $y \in \mathcal{Y}$, where y encodes the exact value for each of the configurable synthesizer parameters. For example, these parameters determine the oscillators' waveform types, the amplitudes' values, modulation indexes, ADSR envelopes, filter cutoff frequency, etc. The inverse-synthesis task is to learn an *encoder* function $e_\theta : \mathcal{X} \rightarrow \mathcal{Y}$, parameterized by θ, that receives an audio $x \in \mathcal{X}$ and predicts the parameters configuration $e_\theta(x) \in \mathcal{Y}$ s.t.

$$f(e_\theta(x)) = x' \approx x. \tag{1}$$

2.2 The IS Model

The IS model from [9] receives an input signal x and aims at inferring a parameters configuration $\hat{y}$ which best matches the true yet unknown parameters configuration y that produced $x = f(y)$. To this end, a dataset $D = \{(x_i, y_i)\}_{i=1}^N$ is generated, where y_i is the synthesizer's configuration used by f to generate the sound x_i, hence $f(y_i) = x_i$. IS trains an encoder network e_θ to predict y_i from x_i by minimizing the objective

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \sum_{i=1}^{N} \mathcal{L}_p(e_\theta(x_i), y_i), \tag{2}$$

where $\mathcal{L}_p : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$ is the *parameters loss* that quantifies the difference between the predicted configuration $e_\theta(x_i)$ and the ground truth configuration y_i. In [9], each synthesizer parameter was treated as a categorical variable (continuous parameters were quantized), hence solving multiple classification problems simultaneously (one for each parameter). Accordingly, the loss $\mathcal{L}_p$ was the sum of P cross-entropy (CE) losses, where P is the number of the synthesizer parameters.

2.3 The IS2 Model

IS does not optimize on the actual reproduced audio signal. Instead, it only optimizes on the parameters configuration according to Eq. 2. However, minimizing $\mathcal{L}_p$ is just a proxy to the original task from Eq. 1 that aims at minimizing the difference between the original signal x and the reproduced signal $f(e_\theta(x))$. This observation motivates an additional self-supervised loss term $\mathcal{L}_a : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$, namely the *audio loss*, that measures the discrepancy between the input signal x and the reproduced signal $f(e_\theta(x))$:

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \sum_{i=1}^{N} \mathcal{L}_p(e_\theta(x_i), y_i) + \lambda \mathcal{L}_a(f(e_\theta(x_i)), x_i), \tag{3}$$

where λ is a hyperparameter. The audio loss term $\mathcal{L}_a$ provides feedback on the quality of the reproduced signal $f(e_\theta(x_i))$ itself, hence better aligns with the ultimate task of Eq. 1.

Yet, a key challenge arises - how to backpropagate the error induced by $\mathcal{L}_a$ via f? A naive approach may propose implementing the synthesizer f as part of the computational graph. However, this approach suffers from several limitations: First, it requires a specific implementation per synthesizer and hence does not scale. Second, most commercial synthesizers are not open-source, and even if the source code was provided, it would still require rewriting of the entire codebase to support an auto-differentiation platform (e.g., PyTorch). Furthermore, some synthesizer functionalities are not differentiable and require workarounds that may incur discrepancies and hinder gradient-based optimization.

To this end, IS2 introduces a *synthesizer-proxy* decoder network $d_\phi : \mathcal{Y} \rightarrow \mathcal{X}$, parameterized by ϕ, that serves as a differential replacement to the true synthesizer function f. d_ϕ is trained to minimize $\mathcal{L}_a$ w.r.t. ϕ over the dataset D, which leads to the IS2 training objective

$$\Theta^* = \underset{\Theta}{\operatorname{argmin}} \sum_{i=1}^{N} \mathcal{L}_p(e_\theta(x_i), y_i) + \lambda \mathcal{L}_a(d_\phi(e_\theta(x_i)), x_i), \tag{4}$$

with $\Theta = \{\theta, \phi\}$.

2.4 IS2 Training

IS2 employs stochastic gradient descent optimization [24] on the objective from Eq. 4 as depicted in Fig. 1(a) (the exact implementation and optimization details will follow in Secs. 2.6 and 3.2). We apply a K-fold cross-validation procedure over the dataset D, where each fold defines different training, validation, and test sets. For each fold, we train the IS2 model on the training set and monitor the following measure on the validation set $V \subset \{1..N\}$:

$$\mathcal{L}_V^f := \sum_{i \in V} \mathcal{L}_a(f(e_\theta(x_i)), x_i). \tag{5}$$

Finally, the best-performing model (in terms of $\mathcal{L}_V^f$ across all epochs) is used for reporting results on the test set.

Note that the predicted parameters $e_\theta(x)$ in Eq. 5 are propagated to the *true* synthesizer f and not to the synthesizer-proxy d_ϕ (see Fig. 1(b)). This enables selecting the model that minimizes the discrepancy between x and $f(e_\theta(x))$, which aligns with the ultimate task of Eq. 1. Yet, f does not participate in the optimization objective (Eq. 4) since it is not necessarily differentiable. Instead,

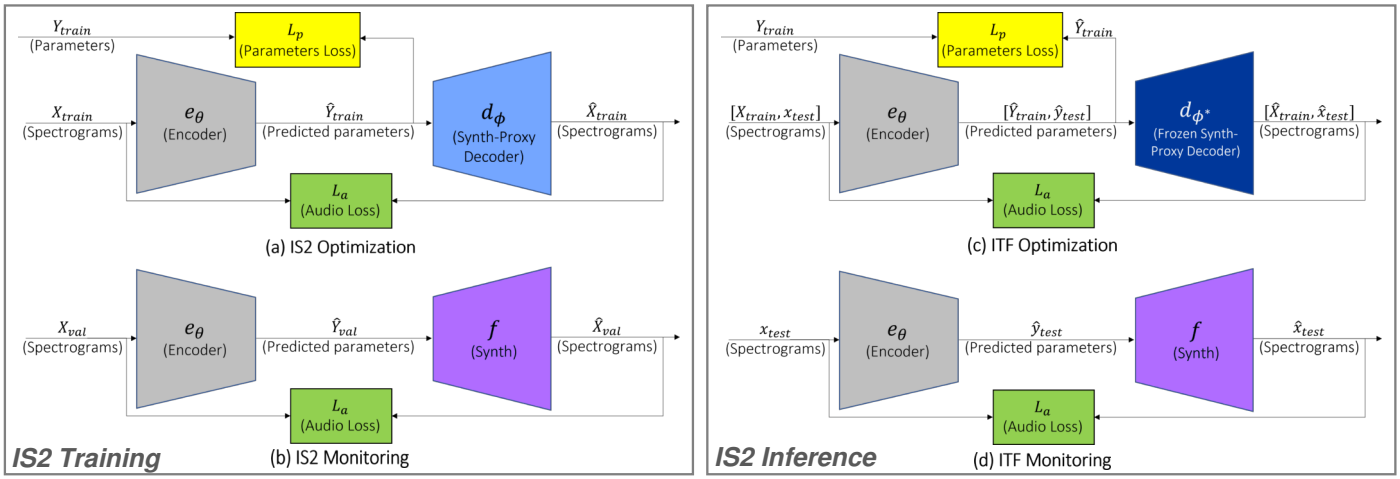


Figure 1: (a)-(b) depict the IS2 training phase (Sec. 2.4) that utilizes the differentiable synthesizer-proxy d_ϕ, while monitoring for the best model via the true synthesizer f. (c)-(d) depict the IS2 inference phase (Sec. 2.5) that employs ITF, utilizing the optimized d_{ϕ^*} for improved parameters prediction on the specific test example. Again, f is used for monitoring.

the audio loss term $\mathcal{L}_a$ in Eq. 4 utilizes d_ϕ as a differentiable proxy to f in order to propagate gradients as part of the optimization process.

2.5 IS2 Inference

A unique feature of IS2 is the ability to improve the predictions at inference time, by employing Inference-Time Finetuning (ITF). Given a test input x_t, we utilize the audio loss $\mathcal{L}_a$ for leveraging self-supervision from x_t, and refine the prediction specifically for x_t. To this end, we freeze the trained decoder parameters ϕ^* (Eq. 4) and finetune the trained encoder parameters θ^* to obtain finetuned parameters θ^t:

$$\theta^t = \underset{\theta}{\text{argmin}}\ \mathcal{L}_t + \lambda_B \mathcal{L}_B, \tag{6}$$

where

$$\mathcal{L}_t = \mathcal{L}_a(d_{\phi^*}(e_\theta(x_t)), x_t),$$
$$\mathcal{L}_B = \frac{1}{|B|}\sum_{i \in B} \mathcal{L}_p(e_\theta(x_i), y_i) + \lambda \mathcal{L}_a(d_{\phi^*}(e_\theta(x_i)), x_i),$$

and $B \subset \{1..N\}$ is a subset of indexes from the training set (a training batch). While one could optimize only $\mathcal{L}_t$ w.r.t. θ, we found that the inclusion of $\mathcal{L}_B$ serves as a regularization (controlled by the hyperparameter λ_B) that leads to more accurate predictions. This can be explained by the fact that $\mathcal{L}_B$ enforces the encoder to predict accurate configurations for the examples in B, effectively safeguarding the encoder from forgetting what it has learned during the training phase (Sec. 2.4) and avoid overfitting the test example x_t. In practice, the ITF procedure alternates between sampling a batch of examples from the training set $B \subset \{1..N\}$, and performing gradient descent update to θ according to the objective in Eq. 6, until either convergence w.r.t. $\mathcal{L}_t^f := \mathcal{L}_a(f(e_\theta(x_t)), x_t)$ is met or the number of alternations exceeds a prescribed threshold. ITF optimization and monitoring are depicted in Fig. 1(c)-(d).

It is important to clarify that ITF is applied per test example, i.e., for each test example x_t, we first initialize $\theta \leftarrow \theta^*$, where θ^* are the optimal encoder parameters obtained from the IS2 training procedure (Eq. 4). Then, ITF alternations are employed according to Eq. 6 to obtain finetuned encoder parameters θ^t that might improve $\mathcal{L}_t^f$. However, improvement is not guaranteed due to an inherent discrepancy that may exist between the synthesizer-proxy decoder d_{ϕ^*} (used in $\mathcal{L}_t$) and the synthesizer f (used in $\mathcal{L}_t^f$). Therefore, if none of the ITF alternations yield improvement to $\mathcal{L}_t^f$, we fallback to the prediction obtained by the originally trained encoder $e_{\theta^*}(x_t)$ (that serves as a starting point for the ITF procedure).

2.6 IS2 Implementation

In [9], various encoder implementations were investigated and the spectrogram-based strided CNN encoder stood out as the best performer. Following this finding, we implement the encoder e_θ and decoder d_ϕ as strided CNNs. Accordingly, $x \in \mathcal{X}$ stands for the *processed* log-magnitude spectrogram or mel-spectrogram domain (where the spectrogram is obtained by the application of the STFT to the waveform), and $\mathcal{L}_a$ is set to the Mean Absolute Error, hence measuring the spectral difference between the input and reproduced signals.

The synthesizer parameters configuration is encoded by a super-vector $y \in \mathcal{Y}$ that concatenates one-hot vectors and normalized scalars representing the categorical and continuous parameters, respectively. Accordingly, the parameters loss $\mathcal{L}_p$ is set to the average of the cross-entropy and L2 losses for categorical and continuous parameters, respectively. The exact details of the data processing, data representation, and hyperparameters settings appear in Sec. 3.

3. EXPERIMENTAL SETUP AND RESULTS

3.1 Datasets, preprocessing, and data representation

In this study, we present findings from analyses conducted on three distinct datasets. As a consequence of space constraints, it is not feasible to detail all the numerous configurable parameters of every synthesizer utilized in our experiments. Nevertheless, a comprehensive account of

Metric	Flow	IS	IS2xITF	IS2
FM Dataset				
Spec (x100)	4.89	1.61	1.54	**1.51**
Melspec (x100)	193.93	56.77	54.65	**53.84**
MFCC (x100)	73.49	28.83	27.74	**27.29**
SC	0.0941	0.0383	0.0367	**0.0361**
ACC (%)	93.01	93.89	93.97	**94.04**
DX7 Dataset				
Spec (x100)	65.31	58.83	58.59	**58.18**
Melspec (x100)	24.04	19.29	19.37	**19.26**
MFCC (x100)	1502.2	1309.5	1300.4	**1280**
SC	1.0472	0.8578	0.8594	**0.8532**
ACC (%)	85.36	86.07	86.34	**86.74**
MAEparam (x100)	10.77	9.79	9.68	**9.56**
TAL Dataset				
Spec (x100)	0.44	0.1809	0.177	**0.173**
Melspec (x100)	106.5	68.06	67.07	**64.64**
MFCC (x100)	8.95	5.85	**5.72**	5.8
SC	0.51	0.512	0.467	**0.424**
ACC (%)	80.94	80.62	80.73	**81.17**

Table 1: Aggregated results on all datasets and metrics.

each synthesizer parameter can be found in the supplementary material accompanying this manuscript. The datasets which were used in this research are as follows: (1) **FM**: is based on the FM synthesizer implementation that is available in IS2 GitHub repository. The synthesizer is composed of a FM oscillator, AM modulation, and low-pass filter. It includes 9 configurable parameters, each represented by a categorical variable. Continuous parameters were discretized and binned to create a finite set of values. A dataset of 180K audio samples (1 second, 16KHz) was generated based on a random sampling of parameter configurations. Samples were transformed into 257x129 spectrograms using log-magnitude STFT (with window size 512 and hop size 128) followed by normalization to [-1,1]. (2) **DX7**: is the dataset from [10] which is based on the Dexed synthesizer [1] which is a virtual replica of the Yamaha DX7 synthesizer with 144 configurable parameters (represented by 54 categorical and 90 continuous variables). It contains 30K audio samples (3 seconds, 22.05KHz). Each sample was transformed into a 257x347 mel-spectrogram (257-bins) of the log-magnitude STFT (with window size 1024 and a hop size 256), followed by normalization to [-1,1]. (3) **TAL** is based on the commercial synthesizer: TAL-NoiseMaker [2]. It consists of 180k audio samples generated using 9 configurable parameters controlling the oscillator, LFO1, LFO2, and cutoff parameters. Each sound has a duration of 1 second sampled at 16kHz and is converted into a 257x129 spectrogram. The spectrograms are normalized to the range [-1, 1] using the same method as the FM synthesizer dataset. The code for the generation of the datasets, including the parameter discretization process is available in our GitHub repository.

The above datasets encompass a broad spectrum of sounds that vary from basic sine waves to intricate waveforms with a wealth of harmonics. The TAL Noisemaker and FM synthesizer datasets comprise a range of sounds including bass, leads, pads, plucks, and percussion, while the DX7 dataset comprises percussive, bell-like, and metallic sounds, in addition to rich pads and complex bass sounds.

[1] https://github.com/asb2m10/dexed

[2] https://tal-software.com/products/tal-noisemaker

Our GitHub repository includes scripts that can reproduce the these datasets.

3.2 Evaluated methods and hyperparameters setting

The following models were evaluated: (1) **IS2**: our model from Sec. 2. e_θ and d_ϕ are implemented by strided and transposed CNNs with 9 hidden LeakyReLU activated layers and Batch Normalization [25] (the exact hyperparameters which were chosen for each layer can be seen in our GitHub code). The IS2 objective (Eq. 4) was optimized with $\lambda = 1$ using the Adam optimizer [26] with $\beta_1 = 0.9$, $\beta_2 = 0.99$, batch size 64 and learning rate scheduling from 10^{-4} to 10^{-6} for 100 epochs. While training, we monitored $\mathcal{L}_V^f$ (Eq. 5) on the validation set, and the best-performing model was selected eventually. For each test sample, we employed 30 ITF alternations according to the objective from Eq. 6, with $\lambda_B = 1$, and B is a stochastic sample of 64 examples drawn randomly from the training set at each alternation. Finally, L_t^f was monitored for selecting the best result as explained in Sec. 2.5. (2) **IS2xITF**: an ablated version of IS2, in which ITF is not employed and the predictions are performed by the trained encoder e_{θ^*}. (3) **IS**: the IS method from [9]. (4) **Flow**: the method from [10] which is based on variational inference with normalizing flows. We tuned hyperparameters for all models using the validation set.

3.3 Evaluation metrics

We report the average results obtained by a 5-fold cross-validation procedure with 80%-10%-10% (training, validation, test) splits, on the following metrics: (1) **Spec**: the Mean Absolute Error (MAE) between the log-magnitude STFTs of a - the signal reproduced by the application of f to the predicted parameters configuration, and b - the ground truth configuration signal. (2) **Melspec**: the MAE between the mel-spectrograms of a and b. (3) **MFCC**: the MAE between the 40-band MFCCs of a and b. (4) **SC**: the Spectral Convergence [27] between a and b. Note that SC was found less correlated with human perception [10], nevertheless we report this metric for the sake of completeness. (5) **ACC**: the accuracy of the predicted categorical synthesizer parameters. (6) **MAEparam**: MAE between the predicted and ground truth numerical synthesizer parameter values. This metric is reported for the DX7 dataset only, as all parameters in the FM and TAL dataset are modeled by categorical variables. Metrics (1)-(4) measure errors in the reproduced signal $f(e_\theta(x))$, while metrics (5)-(6) measure accuracy / error w.r.t. the ground truth parameter configurations.

To complete our evaluations, we also present the results of a MOS (Mean Opinion Score) test [28] with $N = 20$. The MOS test involves presenting a set of synthesized sounds to a panel of listeners, who are then asked to rate the sound quality of the reconstructed sound with respect to the original sound using a standardized rating scale: $[1-5]$.

3.4 Quantitative Results

Table 1 displays the results obtained by all methods across all datasets and evaluation metrics. The ACC and MAEparam are averages across all categorical and continuous parameters, respectively. It is important to note that the Flow results reported in [10] were replicated, and our

	TAL Dataset				FM Synth Dataset				DX7 Dataset			
	IS2	**IS2xITF**	**Flow**	**IS**	**IS2**	**IS2xITF**	**Flow**	**IS**	**IS2**	**IS2xITF**	**Flow**	**IS**
Low (x100)	**608**	627	619	642	**0.62**	0.63	1.88	0.79	**0.53**	0.54	0.61	0.533
Mid (x100)	**86.82**	90.26	102	91.64	**0.18**	0.19	0.51	0.24	0.02658	0.02652	0.0338	**0.0265**
High (x100)	**5.82**	6.12	7.7	6.64	**0.0036**	0.0037	0.0224	0.0098	0.0038	0.0037	0.0043	**0.0035**
All bands (x100)	**654**	675	676	691	**0.63**	0.64	0.19	0.8	**0.53**	0.538	0.6	0.537

Table 2: Spectral analysis for different Mel-frequency bands (x100).

DX7 Param	**Type**	**Flow**	**IS**	**IS2**	**IS2xITF**
ALGORITHM	cat	0.5758	0.6660	**0.6676**	0.6627
FEEDBACK	cat	0.6938	0.7056	**0.7179**	0.7151
OSCKEYSYNC	cat	0.8227	0.8269	**0.8356**	0.8356
LFOSPEED	num	12.5070	**11.5924**	11.6528	11.6337
LFODELAY	num	16.5244	15.0604	14.9376	**14.8934**
LFOPMDEPTH	num	13.0251	11.7224	**11.5284**	11.5593
LFOAMDEPTH	num	17.8000	17.7326	**17.5715**	17.6148
LFOKEYSYNC	cat	0.8008	0.8112	**0.8163**	0.8150
LFOWAVE	cat	0.7599	0.7622	**0.7665**	0.7618
PMODESENS	cat	0.6214	0.6473	**0.6598**	0.6590
PITCHEGRATE1	num	17.6189	16.8161	**16.6582**	16.6706
PITCHEGRATE2	num	17.7838	**16.7808**	16.8625	16.8100
PITCHEGRATE3	num	18.2013	17.6938	**17.4543**	17.5203
PITCHEGRATE4	num	19.5772	**18.6485**	18.7511	18.7084
PITCHEGLEVEL1	num	**6.2437**	6.3104	6.3948	6.3948
PITCHEGLEVEL2	num	**6.5503**	6.8803	6.8803	6.8803
PITCHEGLEVEL3	num	**6.8834**	7.1543	7.1543	7.1543
PITCHEGLEVEL4	num	**6.1773**	6.4220	6.4220	6.4220
OP_EGRATE1	num	13.4020	12.2691	12.1219	**12.1169**
OP_EGRATE2	num	17.7775	**16.8140**	16.9248	16.8298
OP_EGRATE3	num	17.9113	17.0677	17.1096	**17.0545**
OP_EGRATE4	num	12.6359	11.8326	11.8028	**11.7685**
OP_EGLEVEL1	num	**10.6051**	10.7961	10.8860	10.9058
OP_EGLEVEL2	num	17.9278	16.8495	**16.7416**	16.7842
OP_EGLEVEL3	num	21.2140	20.3835	20.2471	**20.2462**
OP_EGLEVEL4	num	**12.1882**	15.1754	15.1754	15.1754
OP_OUTPUTLEVEL	num	12.4545	11.5483	**11.5102**	11.5341
OP_MODE	cat	0.9359	0.9404	**0.9446**	0.9430
OP_FCOARSE	cat	0.6951	0.7486	**0.7538**	0.7513
OP_FFINE	num	13.8020	13.3172	**13.2250**	13.2555
OP_OSCDETUNE	cat	0.6578	0.7275	**0.7346**	0.7299
OP_BREAKPOINT	num	16.3170	**15.4886**	15.5501	15.4929
OP_LSCALEDEPTH	num	16.0303	16.0030	**15.9739**	16.0440
OP_RSCALEDEPTH	num	16.3427	15.7286	**15.6248**	15.6984
OP_LKEYSCALE	cat	0.8476	0.8505	**0.8574**	0.8526
OP_RKEYSCALE	cat	0.8533	0.8541	**0.8643**	0.8580
OP_RATESCALING	cat	0.7187	0.7528	**0.7598**	0.7579
OP_AMODSENS	cat	0.9112	0.8987	**0.9185**	0.9052
OP_KEYVELOCITY	cat	0.6777	0.7236	**0.7290**	0.7267
MEAN CAT	-	0.7551	0.7796	**0.7875**	0.7838
MEAN NUM	-	14.3	13.8434	**13.8064**	13.8067

Table 3: Aggregated DX7 parameters' accuracy. The functionality of each parameter is explained in the supplementary materials (appears in our GitHub repository).

Param TAL	Flow	IS	IS2	IS2xITF
x3_FilterCutoff (%)	**86.08**	72.01	75	72.8
x24_Osc2Waveform (%)	**99.82**	99.38	99.48	99.39
x20_Osc2Tune (%)	**95**	93.49	93.7	93.6
x26_Lfo1Waveform (%)	68.28	78.33	**78.8**	78.38
x28_Lfo1Rate (%)	47.54	52.93	**53.34**	52.97
x30_Lfo1Amount (%)	**73.37**	71.74	72.19	71.78
x27_Lfo2Waveform (%)	88.86	**88.87**	88.85	88.76
x29_Lfo2Rate (%)	**84.77**	84.56	84.67	84.59
x31_Lfo2Amount (%)	**84.77**	84.28	84.45	84.31
MEAN (%)	80.94	80.62	**81.17**	80.73

Table 4: TAL parameters' accuracy. The parameters are prefixed with "xAB", where AB denotes the index of the parameter within the synthesizer. The functionality of each parameter is explained in the supplementary materials (appears in our GitHub repository).

Param FM	Flow	IS	IS2	IS2xITF
osc1_wave (%)	**99.98**	99.94	99.94	99.94
osc1_freq (%)	91.26	98.7	**98.83**	98.81
osc1_mod_index (%)	93.08	96.43	**96.5**	96.45
lfo1_freq (%)	**99.95**	99.86	99.88	99.87
lfo1_wave (%)	**99.52**	98.75	98.69	98.67
am_mod_wave (%)	67.73	71.02	**71.74**	71.59
am_mod_freq (%)	**86.23**	82.71	82.88	82.86
am_mod_amount (%)	**99.43**	97.62	97.64	97.61
filter_freq (%)	**99.98**	99.93	99.95	99.94
MEAN (%)	93.02	93.89	**94.01**	93.97

Table 5: FM Synth parameters' accuracy. The functionality of each parameter is explained in the supplementary materials (appears in our GitHub repository).

Dataset	**Flow**	**IS**	**IS2xITF**	**IS2**
FM	4.7	4.85	4.6	**5**
DX7	1.35	2.45	3.28	**3.5**
TAL	3.87	3.37	3.85	**3.95**

Table 6: MOS test results. Scores on a scale of $[1-5]$ represent the perceptual reconstruction quality w.r.t. the original audio.

results are consistent with the original findings. Table 1 demonstrates that our IS2 method outperforms the other baselines in all metrics and datasets, except for the MFCC score on the TAL datasets, where the ablated version of IS2, IS2xITF, outperforms it. Furthermore, the results indicate that the ablated version IS2xITF is highly effective in comparison to previous baselines which highlights the general utility of the IS2 architecture even without the ITF phase. In the following section, we aim to provide a more comprehensive analysis and interpretation of these results.

To provide additional perspective, we conducted the following analysis: We partitioned the 257 mel-spectrogram bins into "Low", "Mid", and "High" equally sized mel-frequency bands. Then, for each sound in the test set, we computed the L2 loss between the reproduced version and the ground-truth of each mel-frequency band. The results for the different frequency bands, including the entire mel-spectrogram ('All bands') are presented in Table 2. First, We observe that IS2 outperforms the other models on the entire mel-spectrogram ('All bands'), across all datasets, which is consistent with the results presented in Table 1 (note that Tables 1 and 2 report different metrics, i.e, MAE vs. L2). Specifically, IS2 performs particularly well on low frequency regime ('Low'). Arguably, this finding might be explained later where we shall see that the IS2 model attains the best loss in 4 out of 6 low-frequency oscillator (LFO) parameters, which have a stronger impact on the low bands. This calls for further research into the relationship between parameter prediction accuracy and the mel-spectrogram error.

In terms of the "Mid" band, the IS2 model demonstrated superior performance on the TAL and FM datasets,

whereas the IS model exhibited better results on the DX7 dataset. For the "High" band, different baselines achieved the best outcomes. This observation is not surprising since high frequencies typically undergo rapid changes and can be less perceptible even to experienced listeners. Overall, our findings indicate that the IS2 approach exhibits robust performance across various datasets and frequency bands, with exceptional accuracy in estimating low frequencies.

Next, we turn to evaluate the accuracy of predicting each parameter specifically. The DX7 synthesizer consists of two types of parameters: categorical, denoted as "cat", and numerical, denoted as "num". To evaluate performance, ACC was reported for categorical parameters, while MAEparam was calculated for numerical parameters, as previously mentioned. The DX7 parameters are further categorized into several groups, including Algorithm, Feedback, Operators, Pitch Envelope Generator, LFO, and Filter, with a comprehensive explanation of each parameter available in the supplementary material. Table 3 outlines the prediction results for the DX7 parameters. Note that parameters beginning with the prefix "OP" are an average aggregation of the six operators of the synthesizer, as explained in the supplementary material.

The results reveal that the IS2 method consistently outperformed other models in predicting all categorical parameters. Furthermore, the IS2 model also outperformed other models in 9 out of 25 numerical parameters. Notably, the IS2xITF model performed best among all models for the "OP_EGRATEi" (i=1...4) numerical parameters, demonstrating superior performance even without fine-tuning (ITF). Specifically, this model exhibited better performance in all numerical parameters and outperformed other models in 5 out of 25 numerical parameters. In contrast, the IS and Flow models demonstrated similarly robust performance, outperforming other models in fewer numerical parameters, namely 11 out of 25.

Overall, Table 3 displays a trend where all categorical parameters are more accurately estimated by the IS2 model. In terms of numerical parameters, the IS2 model performs better in predicting parameters with greater perceptual significance, such as LFO. However, parameters of lesser perceptual significance, such as the envelope level of pitch (PITCHEGLEVEL), exhibit lower estimation accuracy. These trends are consistent with the findings presented earlier in Table 2

The trends observed in Table 3 repeat themselves in the TAL dataset (Table 4) and the FM Synth dataset (Table 5), with slightly improved accuracy for the alternative models. Nevertheless, the IS2 model maintains the highest mean accuracy. IS2 does not achieve the highest accuracy in certain parameters, such as filter cutoff, which are of lesser significance for perceived quality. For example, if the filter cutoff is estimated for class A instead of the correct class B, and A and B are neighboring classes, the impact on perception might be insignificant. Another set of parameters that demonstrate negligible differences in accuracy between models is Oscillator 2, LFO1 amount, and LFO2 values. In contrast, parameters such as LFO1 waveform and rate play a crucial role in controlling Oscillator 2 modulation and impacting the low frequencies of the sound, making them significant in terms of perception. Here, IS2 achieves significantly higher accuracies compared to other models. These findings are consistent with the low losses of the IS2 model for low frequencies, as presented in Table 2.

In Table 5, Oscillator 1 waveform, LFO1, and filter cutoff frequency exhibit negligible differences in accuracy in favor of the alternative baselines. Higher differences are observed in AM modulation parameters. Nevertheless, these parameters primarily affect the Tremolo effect, which has a relatively no impact at all on the frequency composition, and for small changes, leading to less influence on human perception and less impact on the metrics presented in Table 1. Compared to the Flow model, the parameters with the most significant differences are Oscillator 1 frequency and modulation index, which have a significant impact on perception by affecting the carrier frequency of the signal.

Overall, the results in Tables 1-5 indicate that by leveraging information on the difference between the original and reproduced signal during training and inference (ITF), IS2 promotes accurate predictions for parameters that have the most significant impact on human perception, especially FM modulator parameters which very challenging to estimate. Consequently, IS2 produces reconstructions that more closely resemble the original signal, which is the ultimate goal of sound matching. In what follows, we further substantiate our findings via human subjective evaluation.

3.5 MOS Test and Qualitative Results

Table 6 presents MOS test results conducted using $N = 20$ individuals. Participants were asked to rate the reconstruction score of 80 random audio samples on a $[1 - 5]$ scale. The results are inline with those of Table 1, showing that the IS2 model outperforms the other models. Furthermore, the MOS test results indicate that the IS2 model has the highest perception quality of all the models evaluated. The samples from this test are available for listening on the GitHub repository.

Finally, in the supplementary materials, we provide extensive qualitative comparison between the ground-truth spectrograms and the reconstructions produced by each of the evaluated methods. Additionally, the audio signals for these examples are provided are available in our Google Drive folder [3] .

4. CONCLUSION

We presented IS2 - a novel model for automatic synthesizer sound matching. IS2 introduces two novel contributions: (1) a differentiable synthesizer-proxy decoder that enables gradient-based optimization of the reproduced audio signals, and (2) the ITF technique that enables improved model predictions at inference time. These contributions lead to state-of-the-art results compared to existing methods across multiple datasets and metrics.

5. ACKNOWLEDGMENTS

This research was supported by the ISRAEL SCIENCE FOUNDATION (Grant No. 2243/20).

[3] https://drive.google.com/drive/folders/1VFnE5fcEfbbmKdNT-NxMvXotSdz5mOQk?usp=sharing

6. REFERENCES

[1] G. De Poli, "A tutorial on digital sound synthesis techniques," *Computer Music Journal*, vol. 7, no. 4, pp. 8–26, 1983.

[2] J. Shier, "The synthesizer programming problem: improving the usability of sound synthesizers," Ph.D. dissertation, 2021.

[3] A. Horner, J. Beauchamp, and L. Haken, "Machine tongues xvi: Genetic algorithms and their application to fm matching synthesis," *Computer Music Journal*, vol. 17, no. 4, pp. 17–29, 1993.

[4] S. Heise, M. Hlatky, and J. Loviscach, "Automatic cloning of recorded sounds by software synthesizers," in *Audio Engineering Society Convention 127*. Audio Engineering Society, 2009.

[5] S. Luke, "Stochastic synthesizer patch exploration in edisyn," in *International Conference on Computational Intelligence in Music, Sound, Art and Design (Part of EvoStar)*. Springer, 2019, pp. 188–200.

[6] M. Yee-King and M. Roth, "Synthbot: An unsupervised software synthesizer programmer," in *ICMC*, 2008.

[7] K. Tatar, M. Macret, and P. Pasquier, "Automatic synthesizer preset generation with presetgen," *Journal of New Music Research*, vol. 45, no. 2, pp. 124–144, 2016.

[8] M. J. Yee-King, L. Fedden, and M. d'Inverno, "Automatic programming of vst sound synthesizers using deep networks and other techniques," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 2, pp. 150–159, 2018.

[9] O. Barkan, D. Tsiris, O. Katz, and N. Koenigstein, "Inversynth: Deep estimation of synthesizer parameter configurations from audio signals," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 12, pp. 2385–2396, 2019.

[10] G. L. Vaillant, T. Dutoit, and S. Dekeyser, "Improving synthesizer programming from variational autoencoders latent space," in *2021 24th International Conference on Digital Audio Effects (DAFx)*, 2021, pp. 276–283.

[11] D. Rezende and S. Mohamed, "Variational inference with normalizing flows," in *International conference on machine learning*. PMLR, 2015, pp. 1530–1538.

[12] P. Esling, N. Masuda, A. Bardet, R. Despres, and A. Chemla-Romeu-Santos, "Flow synthesizer: Universal audio synthesizer control with normalizing flows," *Applied Sciences*, vol. 10, no. 1, p. 302, 2019.

[13] "Dexed github repository," https://github.com/asb2m10/dexed, 2021.

[14] N. Masuda and D. Saito, "Synthesizer sound matching with differentiable dsp." in *ISMIR*, 2021, pp. 428–434.

[15] ——, "Improving semi-supervised differentiable synthesizer sound matching for practical applications," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.

[16] F. Caspe, A. McPherson, and M. Sandler, "Ddx7: Differentiable fm synthesis of musical instrument sounds," *arXiv preprint arXiv:2208.06169*, 2022.

[17] 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.

[18] P. Chandna, M. Blaauw, J. Bonada, and E. Gómez, "Wgansing: A multi-voice singing voice synthesizer based on the wasserstein-gan," in *2019 27th European signal processing conference (EUSIPCO)*. IEEE, 2019, pp. 1–5.

[19] J. Engel, K. K. Agrawal, S. Chen, I. Gulrajani, C. Donahue, and A. Roberts, "Gansynth: Adversarial neural audio synthesis," *arXiv preprint arXiv:1902.08710*, 2019.

[20] K. Kumar, R. Kumar, T. de Boissiere, L. Gestin, W. Z. Teoh, J. Sotelo, A. de Brébisson, Y. Bengio, and A. C. Courville, "Melgan: Generative adversarial networks for conditional waveform synthesis," *Advances in neural information processing systems*, vol. 32, 2019.

[21] A. Défossez, N. Zeghidour, N. Usunier, L. Bottou, and F. Bach, "Sing: Symbol-to-instrument neural generator," *Advances in neural information processing systems*, vol. 31, 2018.

[22] J. Engel, L. Hantrakul, C. Gu, and A. Roberts, "Ddsp: Differentiable digital signal processing," *arXiv preprint arXiv:2001.04643*, 2020.

[23] O. Barkan and D. Tsiris, "Deep synthesizer parameter estimation," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 3887–3891.

[24] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747*, 2016.

[25] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*. PMLR, 2015, pp. 448–456.

[26] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[27] S. Ö. Arık, H. Jun, and G. Diamos, "Fast spectrogram inversion using multi-head convolutional neural networks," *IEEE Signal Processing Letters*, vol. 26, no. 1, pp. 94–98, 2018.

[28] R. C. Streijl, S. Winkler, and D. S. Hands, "Mean opinion score (mos) revisited: methods and applications, limitations and alternatives," *Multimedia Systems*, vol. 22, no. 2, pp. 213–227, 2016.

A SEMI-SUPERVISED DEEP LEARNING APPROACH TO DATASET COLLECTION FOR QUERY-BY-HUMMING TASK

Amantur Amatov[1] **Dmitry Lamanov**[2] **Maksim Titov**[2]
Ivan Vovk[2] **Ilya Makarov**[3] **Mikhail Kudinov**[2]
[1] Higher School of Economics, Moscow, Russia
[2] Huawei Noah's Ark Lab, Moscow, Russia
[3] AI Center, NUST MISiS, Moscow, Russia

ABSTRACT

Query-by-Humming (QbH) is a task that involves finding the most relevant song based on a hummed or sung fragment. Despite recent successful commercial solutions, implementing QbH systems remains challenging due to the lack of high-quality datasets for training machine learning models. In this paper, we propose a deep learning data collection technique and introduce Covers and Hummings Aligned Dataset (CHAD), a novel dataset that contains 18 hours of short music fragments, paired with time-aligned hummed versions. To expand our dataset, we employ a semi-supervised model training pipeline that leverages the QbH task as a specialized case of cover song identification (CSI) task. Starting with a model trained on the initial dataset, we iteratively collect groups of fragments of cover versions of the same song and retrain the model on the extended data. Using this pipeline, we collect over 308 hours of additional music fragments, paired with time-aligned cover versions. The final model is successfully applied to the QbH task and achieves competitive results on benchmark datasets. Our study shows that the proposed dataset and training pipeline can effectively facilitate the implementation of QbH systems.

1. INTRODUCTION

Query-by-Humming (QbH) is a well-known task in Music Information Retrieval. It aims to enable users to find a particular song within a retrieval system by providing a small audio segment of their voice or humming as a query. Such systems rely on a large database of songs and display the most similar matches to the user's query.

One significant benefit of the QbH system compared to other music search systems [1] is that users do not have to play a copy of the song or recall its lyrics. Instead, they can hum or sing the melody of the desired song, and the system will use advanced audio processing and deep learning techniques to locate it.

A similar task to QbH is Cover Song Identification (CSI) task [2–4]. CSI aims to identify cover songs performed by different artists as versions of original songs within a music database. Although CSI systems often rely on neural networks, traditional QbH systems mainly utilize audio processing and music information retrieval techniques like pitch estimation, note extraction, and time series matching [5–7]. The main reason QbH lacks deep learning models is the absence of large datasets for training. This is primarily due to the high cost and limited availability of humming/singing data for QbH compared to CSI, where multiple versions of the same song are sufficient for training. Additionally, QbH requires the alignment of humming/singing fragments with the original versions of the song.

To overcome the challenges of limited data, we propose a novel dataset CHAD - Covers and Hummings Aligned Dataset. This dataset contains groups of time-aligned music fragments, primarily consisting of vocal segments from popular songs. *Time alignment* is the process of synchronizing a fragment from a humming or cover version of a song with its corresponding fragment from the original version to have the same temporal structure. The groups are separated into two categories: one with humming fragments collected via crowdsourcing and another with cover fragments collected using a semi-supervised training pipeline. We use this dataset to train our deep learning model for matching audio fragments with similar melodies using metric learning paradigm. We demonstrate that these techniques can also be successfully applied in the QbH task, achieving results comparable to the best performing scores on popular QbH benchmarks. Furthermore, we evaluate our model's performance on a large internal song database, showing its ability to generalize to a wider range of songs.

The paper is structured as follows. Section 2 briefly reviews existing approaches to the QbH task. Section 3 describes the proposed deep learning model and training method for the QbH task. Section 4 outlines the dataset and semi-supervised data collection pipeline. Section 5 describes the experiments conducted on public and private data. Finally, Section 6 concludes the paper.

© A. Amatov, D. Lamanov, M. Titov, I. Vovk, I. Makarov, M. Kudinov. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** A. Amatov, D. Lamanov, M. Titov, I. Vovk, I. Makarov, M. Kudinov, "A Semi-Supervised Deep Learning Approach to Dataset Collection for Query-by-Humming Task", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.*, Milan, Italy, 2023.

2. RELATED WORK

QbH systems typically have two components: audio transcription and search modules. Many approaches in QbH research have focused on designing effective representations of hummings that can be easily matched with MIDI targets. Some standard methods include using Hidden Markov Models [5] to transcribe hummings into a sequence of symbols, discretizing fundamental frequency into semitones [6], and transcribing hummings into a note-like structure using pitch, interval, and duration features [7].

Once the humming has been transcribed into a format that can be compared to database entries, the search module is responsible for finding the most relevant songs. Dynamic Time Warping [6, 8] has been a popular algorithm for comparing the humming query to MIDI-audio entries in the database. This algorithm finds the minimum path between the discretized humming and the MIDI-audio database. Another approach, top-down melody matching [9], involves dynamically aligning the humming query with a song from the database. A third approach, progressive filtering [10], involves multiple stages of song recognition with increasingly complex recognition mechanisms. These algorithms serve to match humming queries with songs in a database effectively. In the approach [11], authors use melody extraction network to extract robust features from audios and match them with songs from database using an ensemble of melody matching algorithms.

In contrast to QbH, the latest research on the Cover Song Identification (CSI) task has been focused on deep learning-based techniques. A popular approach in CSI is to use deep neural networks for audio representation and metric learning for similarity search. In [12], the authors use Constant-Q Transform (CQT) of audio and train a modified version of ResNet with two losses - triplet loss for intra-class compactness and classification loss for inter-class discrimination. In the next study, [13], the authors improve results by integrating the PCA module into the fully-connected layer of ResNet. Metric learning is widely used in CSI. It is shown that different model and loss architectures like Siamese Network [3], triplet loss [12], and contrastive loss [4] can produce competitive results.

In [2], the authors use VGG on CQT features with variable length to tackle the problem of tempo changes of the cover songs. In [14], the authors use an audio signal's Mel-Frequency Cepstrum Coefficients (MFCC) as the representation. They build cross-similarity matrices between songs and collect the nearest neighbors of each song based on these matrices.

Several datasets are available for CSI tasks [15–18]. These datasets contain audio features alongside music metadata and provide researchers with a way to evaluate and validate their models without collecting large amounts of audio data.

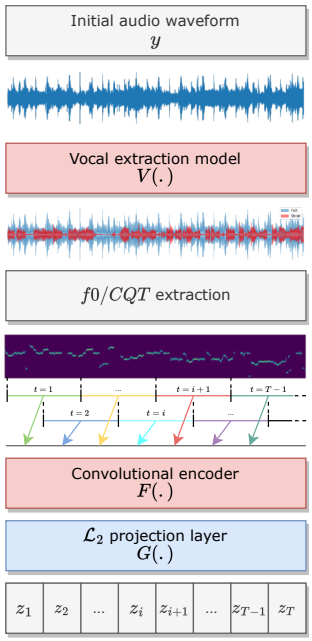


Figure 1: Audio encoder model. Vocal part is extracted from the input waveform. Then, either $f0$ or CQT features are calculated on the vocal part. Finally, the features are processed by a convolutional encoder model and, then, the output embeddings are normalized.

3. MODEL

Our encoder model, M, is presented in Figure 1 and inspired by [4]. The whole fingerprints extraction pipeline can be described in the following steps:

1. The first step of the encoding process is to extract the vocal part of the audio waveform y using a pre-trained audio source separation model $V(.)$. The model is applied only to cover fragments since humming fragments do not contain any accompaniments. We used Spleeter [19] as a model due to its high-speed performance.

2. The vocal part of the audio is then sent to the feature extractor model. In this study, two different extracted feature types are used: the first is the fundamental frequency ($f0$) extracted using the CREPE model [20], which is considered a robust representation of the melody. The second is Constant-Q Transform (CQT) as its faster alternative. The melody is crucial for search as it contains essential song information while ignoring irrelevant singing person details.

3. The extracted feature matrix is then separated into overlapping segments, called analysis windows with length W and step H, which are fed separately to a convolutional encoder $F(.)$ ResNet18 [21]. The final layer of the encoder is a $\mathcal{L}_2$-normalization layer $G(.)$, which normalizes the output of the encoder along the embedding dimension.

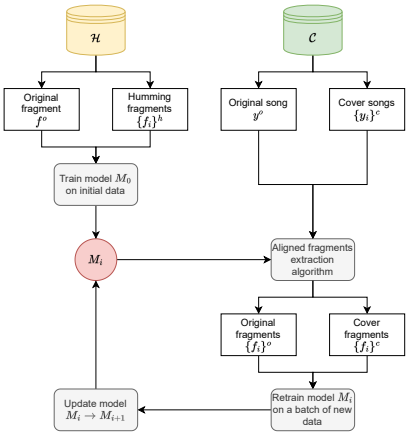


Figure 2: Semi-supervised training and data collection pipeline used to train the initial model, and iteratively gather new aligned audio fragments and retrain the model.

4. The output fingerprints $Z = \{z_i\}_{i=1...T}$, where T is the total number of fingerprints for a waveform and 128 is its dimension size.

We use the metric learning method similar to [4] as a learning framework. To form a batch of audio fragments for training, we randomly sample K groups of time-aligned audio fragments. By *group*, we refer to a collection of original song and humming/singing fragments. Then, we select n random audio fragments from each group and extract a random analysis window of size W. Since our data is aligned, all windows from each group will represent the variations of the same data. Afterward, we apply our model and extract in total $N = K \cdot n$ fingerprints for n in each group.

Our loss is defined as follows:

$$\ell = -\sum_{k=1}^{K} \sum_{z_k^i, z_k^j \in Z_k} \log \frac{\exp(\frac{sim(z_k^i, z_k^j)}{\tau})}{\sum_{z_l \notin Z_k} \exp(\frac{sim(z_k^i, z_l)}{\tau})}, \quad (1)$$

where $Z_k = \{z_k^0, \dots z_k^{n-1}\}$ is the group of fingerprints, z_k^i and z_k^j are different fingerprints from Z_k, $z_l \notin Z_k$ stands for all fingerprints not in a given group k, $sim(z_i, z_j) = z_i^T z_j$ is the similarity function, and τ is a temperature parameter. The final loss is computed across all possible positive pairs and averaged afterward.

4. DATASET

This section describes the process of collecting a dataset for the QbH task, its statistics, and its limitations.

4.1 Semi-supervised pipeline

Figure 2 presents our proposed semi-supervised pipeline. The dataset used in this study is structurally divided into two parts: $\mathcal{H}$ and $\mathcal{C}$. The first part, $\mathcal{H}$, consists of original music fragments f^o paired with time-aligned humming/singing fragments f^h, making groups F^h. The f^o fragments are represented by various vocal and instrumental parts of music clips. The f^h fragments were collected using a crowdsourcing service Yandex.Toloka. The second part of the dataset, $\mathcal{C}$, was created by collecting the 100 most popular songs from the Billboard Charts for each year from 1960 to 2020. For each song, up to 10 cover versions were retrieved from YouTube top results using query "{song name} {artist name} cover".

Algorithm 1: Aligned fragments extraction algorithm of data collection pipeline.

Input : y^o - original song;
Y^c - set of cover songs;
d_{min} - fragment's minimal length;
d_{max} - fragment's maximal length;
D_p - set of pause lengths;
L_{db} - set of dB levels;
α_{corr} - threshold value to exclude same fragments;
Output: $\mathbb{F}$ - set of groups of aligned fragments from original and cover songs

$\mathbb{F}^o, \mathbb{F}^o_{prev} \leftarrow \{\}, \{\}$;
$M \leftarrow$ initialize_model(.);
$rms \leftarrow$ rms(y^o);
foreach $d_p \in D_p$ **do**
  foreach $l_{db} \in L_{db}$ **do**
    $m_{silence} \leftarrow$ find_silence_mask(rms, l_{db});
    $\mathbb{F}^o \leftarrow$ split_by_silence($y^o, m_{silence}$);
    $\mathbb{F}^o \leftarrow$ merge_fragments($\mathbb{F}^o, d_p, d_{min}, d_{max}$);
    $\mathbb{F}^o_{emb} \leftarrow M(\mathbb{F}^o)$;
    $\mathcal{A}_{corr} \leftarrow$ build_correlation_matrix($\mathbb{F}^o_{emb}$);
    $\mathbb{F}^o \leftarrow$ find_unique_fragments($\mathbb{F}^o, \mathcal{A}_{corr}, \alpha_{corr}$);
    $\mathbb{F}^o, \mathbb{F}^o_{prev} \leftarrow max(\mathbb{F}^o, \mathbb{F}^o_{prev})$;
  end
end
$\mathbb{F} \leftarrow \{\}$;
foreach $f^o \in \mathbb{F}^o$ **do**
  foreach $y^c \in \mathbb{Y}^c$ **do**
    $f^o_{emb} \leftarrow M(f^o)$;
    $y^c_{emb} \leftarrow M(y^c)$;
    $\mathbb{F}^c \leftarrow$ cross_correlation($y^c, y^c_{emb}, f^o_{emb}$);
    $\mathbb{F}^c \leftarrow$ filter($\mathbb{F}^c, \beta_{rel}, \beta_{irrel}$);
    $\mathbb{F} \leftarrow \mathbb{F} \cup (f^o, \mathbb{F}^c)$
  end
end

Because $\mathcal{H}$ already has groups of time-aligned fragments, we can train the initial encoder model M_0 with this data. However, $\mathcal{C}$ only contains groups of full song versions instead of time-aligned fragments, so extracting fragments from these groups is necessary. We propose Algorithm 1 for this task. This algorithm is designed to extract the maximum amount of unique fragments from the original versions of the songs and find the corresponding aligned fragments from cover versions of the songs in $\mathcal{C}$. The algorithm can be described in three stages:

Initialization stage

1. As input, the algorithm takes the vocal part of the original song y^o and a group of cover songs Y^c. Additionally, the algorithm takes the minimal and maximal length of the fragment d_{min} and d_{max}, respectively, the set of dB levels L_{db} by which to count the region in song as silent or non-silent, the set of maximal pause lengths D_p between adjacent fragments in a song separated by silence to be considered as one fragment, and threshold values α_{corr}, β_{rel}, and β_{irrel} to exclude unwanted fragments from the out-

put set.

2. Initialize empty sets of unique fragments $\mathbb{F}^o$ and $\mathbb{F}^o_{prev}$, the encoder model M, and rms of the waveform y^o (lines 1-3).

Fragmentation stage

1. To find the best combination of D_p and L_{db} to yield $\mathbb{F}^o$ of maximal size, start two loops by iterating over these sets (lines 4-5).

2. Compute the binary mask of non-silent regions $m_{silence}$ using rms and $l_{db} \in L_{db}$ and find a set of initial fragments by splitting the waveform y^o using this mask. Then, merge adjacent fragments, the pause between which is less than $d_p \in D_p$. Additionally, the length of such fragments should satisfy the condition $d_{min} < |f| < d_{max}, f \in \mathbb{F}^o$. (lines 6-8).

3. Apply model M_i to the found fragments and extract the fingerprints. Then, build the correlation matrix $\mathcal{A}_{corr}$ based on the fragments' fingerprints $\mathbb{F}^o_{emb}$ and exclude the ones with a correlation higher than the threshold α_{corr}. We used the maximum of cross-correlation function to measure the correlation of fingerprints with different lengths (lines 9-11).

4. Find the parameters of dB levels and pause lengths that yield the maximum amount of unique fragments (line 12).

Matching stage

1. Once the unique fragments from the original version of the song $\mathbb{F}^o$ are extracted, initialize the empty set $\mathbb{F}$ to be filled with groups of the time-aligned fragments from original and cover songs and iterate over each found original fragment $f^o \in \mathbb{F}^o$ and each cover song $y^c \in Y_c$ (lines 15-17).

2. Extract fingerprints from original fragment f^o_{emb} and cover song y^c_{emb} using M. Search for the same fragments in the cover song using a cross-correlation function and peak detection algorithm (lines 18-19).

3. Filter out noise cover fragments by establishing two thresholds:

   (a) The cover fragments with correlation above β_{rel} are considered relevant, indicating a high level of certainty that the content of the cover fragment is similar to that of the original fragment.

   (b) The cover fragments with correlation below β_{irrel} are considered irrelevant fragments and are excluded. Fragments with a correlation between these two thresholds are counted as uncertain and require double-checking via additional crowdsourcing.

   Save the gathered groups of aligned fragments (lines 20-22).

We apply this pipeline to song batches of $\mathcal{C}$, which generates new groups of aligned data. These groups are then added to $\mathcal{H}$, and the model, M, is retrained on the newly gathered data. In such a way, we first train M_0 on initial humming data, then iteratively update our model from $M_i \rightarrow M_{i+1}$ and fill our dataset with new data.

For the unique fragments extraction algorithm, we set $d_{min} = 8, d_{max} = 20, D_p = \{0.5, 1, 1.5\}$ seconds, $L_{db} = \{52, 56, 60, 64, 68\}$ dB, $\alpha_{corr} = 0.8$. When searching for fragments in cover versions of songs, we set the optimal thresholds to $\beta_{rel} = 0.5$ and $\beta_{irrel} = 0.3$. All fragments from the same group have equal duration to retain the time-alignment consistency.

4.2 Statistics

We call the collected dataset Covers and Hummings Aligned Dataset (CHAD). Here are the dataset's statistics:

- CHAD contains 5494 original songs, 31630 cover songs, and 5164 hummings fragments.
- The total number of audio fragments is 81781, which amounts to over 270 hours of singing/humming audio fragments and 51 hours of original song fragments. The group size varies from 2 to 31, with an average size of 6 fragments.
- In $\mathcal{H}$, the duration of the fragments ranges from 4 to 20 seconds, with a mean of 11.06 ± 2.67 seconds, and a total for original fragments - 2.12 hours, and for humming fragments - 15.83 hours.
- In $\mathcal{C}$, the duration ranges from 8 to 20 seconds, with a mean of 14.66 ± 2.03 seconds, and a total for original fragments - 49.54 hours, and cover fragments - 259.03 hours, where 194.53 hours are for fragments with correlation above β_{rel}, and 64.50 hours are for fragments with correlation between β_{rel} and β_{irrel}.
- Additionally, the metadata is collected. It includes YouTube video ID, title, author, correlation value, and whether the fragment is double-checked. The dataset's audio IDs, metadata, start and end timestamps and data download script are available in our GitHub repository [1] .

4.3 Limitations

However, our semi-supervised pipeline has some limitations. First, it can only extract vocal data, and the algorithm needs modification to extract instrumental segments. Second, the number of covers is limited, as there are usually fewer cover versions for non-popular songs. Lastly, there will still be some noisy unrelated fragments in the final set due to the automatic validation threshold. Future research could explore using generative networks [22] to overcome these limitations.

[1] https://github.com/amanteur/CHAD

5. EXPERIMENTS

5.1 Experimental setup

Input features. We used CREPE [20] activations as $f0$ features, yielding output features with a size of $(360, T)$. We further enhanced the robustness of the melody feature by trimming it to include only 3 octaves around its mean pitch, following the approach used in [25]. Additionally, we downscaled this representation to the size $(80, \frac{T}{4})$. However, we encountered issues with the slow speed of the melody extraction model during evaluation, rendering the overall approach unscalable. To address this, we incorporated CQT features into our model, extracted with the following parameters: 12 bins per octave with a total of 7 octaves, Hann window, hop length 512, and a sampling rate of 16 kHz.

Augmentations. We found an optimal set of augmentations to every batch of waveform fragments, which included continuous pitch shifting (with a shift range of -4.0 to 4.0 semitones and probability of 0.5), time stretching (with a stretch rate range of 0.8 to 1.25 and probability of 0.8), SpliceOut [26] (with 10 random intervals of 500 frames and probability of 0.8), mixing with other audio samples in the batch (with an SNR range of 5 to 10 dB and probability of 0.8), and adding background noises (with an SNR range of 3 to 30 dB and probability of 0.8).

Model. We discovered that as the length of a hummed or sung recording increases, the tempo/rhythm becomes more mismatched from the original song. So we trained two models with different analysis window lengths (W) and hop sizes (S): M_{short} for shorter recordings (up to 15 sec) with W=3 sec and S=0.25 sec, and M_{long} for longer recordings with W=8 sec and S=0.64 sec. Both models used a vanilla ResNet18 encoder model, with output embeddings of size $(128, T)$, where T is the number of fingerprints.

Training setup. We trained the encoder model using the ADAM optimizer, with a learning rate of $lr = 0.001$ and a batch size of 32 for 100 epochs. We used the NT-Xent Loss [27] with a temperature of $t = 0.05$. We employed the Multi-similarity miner [28] and an adaptive batch sampler to improve convergence speed. The batch sampler selects up to 4 fragments with a random starting point for each fragments group. We trained the model under two settings: only on the $\mathcal{C}$ part and on both $\mathcal{C} + \mathcal{H}$ parts of CHAD. Models were trained on 1 NVIDIA GeForce RTX 2080 Ti 12 Gb.

To evaluate the performance of our model, we conducted a series of experiments, which involved:

- Experiments on the MIREX QbH datasets [29], specifically the Roger Jang and ThinkIt datasets, where MIDI recordings were used as references. The MIR-QBSH corpus of Roger Jang consists of 4431 query hummings and 48 original MIDI files, while the Thinkit corpus contains 355 queries and 106 original MIDI files. The song database was constructed according to MIREX QbH Challenge standards, with 2600 MIDI files. These experiments aimed to find the ground-truth MIDI by a given query humming.
- Experiments on MIREX QbH datasets according to Subtask 2 testing protocol in MIREX evaluation system. In this protocol, queries are also considered as "versions" of ground truth, and the objective is to retrieve all variants related to a searched ground truth by given query humming.
- Experiments on a dataset of real recordings, which included MIREX Roger Jang Dataset with all MIDI files replaced with real recordings extracted from YouTube videos (Jang Real), and MTG-QbH [24] dataset with 118 queries and 118 original songs. The additional database comprises 1886 random songs from the internal dataset to serve as imposter songs.
- Experiments on a large-scale internal database (DB90K) containing more than 90k real song recordings. For this experiment, we used two types of queries: 126 humming fragments for 126 songs collected by our team as search-by-humming setup and 2000 singing fragments from karaoke recordings gathered from the DAMP-VPB dataset [30] as search-by-singing setup. In the latter case, we selected 5 of 16 original songs and their sung performances and manually split them into fragments.

For all real recordings, we extract the vocal part beforehand. Also, we ensure that CHAD does not contain any songs that are also present in the evaluation datasets. This was achieved by excluding such songs from the training set.

Retrieval. We use two variants of sequence matching methods at the retrieval phase: maximum Pearson correlation coefficient (Corr) or Dynamic Time Warping (DTW) [31]. For the large-scale experiment on DB90K, we use a two-step search procedure with a first step of fast retrieval of preliminary candidates using the FAISS Approximate Nearest Neighbors (ANN) algorithm with Euclidean distance followed by a second step of reranking. After further analysis, we discovered that Euclidean distance and Cosine distance yielded similar results. To maintain simplicity, we chose to use Euclidean distance. The ANN search returns the top 5000 candidates, which are reranked based on the Pearson correlation score.

Metrics. We follow the MIREX evaluation protocols [29] for the QbH task and compute the mean of the Top-n hit rate for every humming/singing fragment. There was only one related song in the database for every query fragment.

5.2 Results

We compare our model M_{short} trained on $\mathcal{C} + \mathcal{H}$ with 2 best performing methods according to the latest available result of MIREX QbH Challenge [32]. The first one [8] is based on $f0$-matching technique. The second one is a proprietary method for which only scores were reported in the leaderboard. We use only one of our models (M_{short} on CREPE and CQT features) in this experiment as most

Method		Top-10 hit rate ↑				
		Jang [23]	Thinkit	Subtask 2	Jang Real	MTG-QBH [24]
Ours	metric learning(CREPE)	0.921	0.966	0.959	0.868	0.883
	metric learning(CQT)	0.840	0.786	0.866	0.867	0.747
Stasiak [8]	$f0$-matching	0.948	0.907	0.968	-	-
ACRCloud	proprietary	**0.990**	**0.986**	**0.972**	-	-

Table 1: Evaluation of model M_{short} trained on dataset $\mathcal{C} + \mathcal{H}$ with two types of features - CREPE and CQT. Evaluation is provided on MIREX datasets - Jang, Thinkit, and Subtask 2, and datasets - Jang Real, and MTG-QBH, which are more applicable to real-world scenarios.

query fragments are shorter than 16 seconds. We use DTW for matching feature sequences on MIDI-based datasets and Corr for non-MIDI datasets as we found that correlation coefficient gives performance improvement on real data.

The results are summarized in Table 1. Our model demonstrates competitive though slightly inferior performance on the given benchmarks. On real (non-MIDI) data our implementation of [8] produced near-random results which can be explained by the difficulty of tracking and matching $f0$ in real music recordings. Also, we see that while using CQT features led to a performance drop, it is not prohibitively large so CQT features can be used when computing $f0$ is infeasible.

Table 2 shows the scalability of our approach in the experiment with DB90k. We do not track the top-1 hit rate as the database contains several versions of the same song. Table 2a reports the results of the search-by-humming setup. We used M_{short}, M_{long}, and their combination model M_{fused}, which worked on a simple rule: M_{short} was used for hummings shorter than 15 seconds, while M_{long} was used otherwise. All presented models are trained with CQT features. We observed that M_{fused} worked better than M_{short} and M_{long} separately in all scenarios. Comparing models trained on $\mathcal{C}$ and $\mathcal{C} + \mathcal{H}$, the accuracy gap suggests that training on real humming data is crucial for search-by-humming setup. In Table 2b, we report our results for search-by-singing setup with M_{fused} on DAMP-VPB. Our model, trained on both $\mathcal{C}$ and $\mathcal{C} + \mathcal{H}$, retrieves the correct songs with high precision, with no performance drop observed for the model trained on $\mathcal{C}$ alone, due to the dominance of sung fragments in our training dataset.

Additionally, we evaluate the retrieval speed of our models M_{short} and M_{long} on DB90k, as shown in Table 3. We find that M_{long} performs better than M_{short} in both search steps (ANN and Reranking) due to its ability to process longer humming recordings and thus require less processing of fragments. Our results demonstrate the scalability and efficiency of our search system in efficiently achieving high-precision results.

Partition	Model	Top-n hit rate↑			
		100	10	5	3
$\mathcal{C}$	M_{short}	0.643	0.548	0.524	0.476
	M_{long}	0.412	0.277	0.270	0.262
	M_{fused}	0.759	0.621	0.603	0.517
$\mathcal{C} + \mathcal{H}$	M_{short}	0.659	0.595	0.571	0.484
	M_{long}	0.595	0.508	0.413	0.389
	M_{fused}	0.776	0.707	0.691	0.586

(a) Results on humming queries.

Partition	Model	Top-n hit rate↑			
		100	10	5	3
$\mathcal{C}$	M_{fused}	0.931	0.904	0.885	0.865
$\mathcal{C} + \mathcal{H}$		0.923	0.899	0.885	0.856

(b) Results on singing queries.

Table 2: Evaluation on DB90K with humming and singing fragments using models M_{short}, M_{long}, and their fusion model M_{fused} trained on $\mathcal{C}$ and $\mathcal{C} + \mathcal{H}$ with CQT features.

Model	Search step, s	
	ANN	Reranking
M_{short}	1.41 ± 0.57	5.37 ± 0.87
M_{long}	0.52 ± 0.11	2.39 ± 0.43

Table 3: Query search speed on DB90K using models M_{short} and M_{long} trained on $\mathcal{C} + \mathcal{H}$ with CQT features using 32 CPU.

6. CONCLUSIONS

In this paper, we propose a novel dataset CHAD alongside a semi-supervised data collection and training pipeline for a Query-by-Humming system. We show that cover songs could be used to train query-by-humming models with competitive performance. Although the model trained on open data performs well on sung queries, the pure search-by-humming setup requires adding a portion of real humming data into the training set for acceptable performance. The main disadvantage of the proposed approach is that it cannot be used for searching instrumental tracks. One possible solution to this problem would lie in the field of dominant melody extraction and generative networks and is left for future research.

7. REFERENCES

[1] A. Wang, "An industrial strength audio search algorithm," in *ISMIR*, 2003.

[2] Z. Yu, X. Xu, X. Chen, and D. Yang, "Learning a representation for cover song identification using convolutional neural network," *CoRR*, vol. abs/1911.00334, 2019. [Online]. Available: http://arxiv.org/abs/1911.00334

[3] G. Doras and G. Peeters, "A prototypical triplet loss for cover detection," *CoRR*, vol. abs/1910.09862, 2019. [Online]. Available: http://arxiv.org/abs/1910.09862

[4] S. Chang, D. Lee, J. Park, H. Lim, K. Lee, K. Ko, and Y. Han, "Neural audio fingerprint for high-specific audio retrieval based on contrastive learning," *CoRR*, vol. abs/2010.11910, 2020. [Online]. Available: https://arxiv.org/abs/2010.11910

[5] E. Unal, E. Chew, P. G. Georgiou, and S. S. Narayanan, "Challenging uncertainty in query by humming systems: A fingerprinting approach," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 2, pp. 359–371, 2008.

[6] R. A. Putri and D. P. Lestari, "Music information retrieval using query-by-humming based on the dynamic time warping," in *2015 International Conference on Electrical Engineering and Informatics (ICEEI)*, 2015, pp. 65–70.

[7] L. Lu, H. You, and H.-J. Zhang, "A new approach to query by humming in music retrieval," in *IEEE International Conference on Multimedia and Expo, 2001. ICME 2001.*, 2001, pp. 595–598.

[8] B. Stasiak, "Follow that tune – adaptive approach to dtw-based query-by-humming system," *Archives of Acoustics*, vol. 39, pp. 467 –, 01 2014.

[9] X. Wu, M. Li, J. Liu, J. Yang, and Y. Yan, "A top-down approach to melody match in pitch contour for query by humming," 2006.

[10] J.-S. R. Jang and H.-R. Lee, "A general framework of progressive filtering and its application to query by singing/humming," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 2, pp. 350–358, 2008.

[11] M. Ulfi and R. Mandala, "Improving query by humming system using frequency-temporal attention network and partial query matching," in *2022 9th International Conference on Advanced Informatics: Concepts, Theory and Applications (ICAICTA)*, 2022, pp. 1–6.

[12] X. Du, Z. Yu, B. Zhu, X. Chen, and Z. Ma, "Bytecover: Cover song identification via multi-loss training," *CoRR*, vol. abs/2010.14022, 2020. [Online]. Available: https://arxiv.org/abs/2010.14022

[13] 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)*, 2022, pp. 616–620.

[14] C. J. Tralie and P. Bendich, "Cover song identification with timbral shape sequences," *CoRR*, vol. abs/1507.05143, 2015. [Online]. Available: http://arxiv.org/abs/1507.05143

[15] D. P. Ellis and G. E. Poliner, "Identifying 'cover songs' with chroma features and dynamic programming beat tracking," in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*, vol. 4, 2007, pp. IV–1429–IV–1432.

[16] T. Bertin-Mahieux and D. Ellis, "Large-scale cover song recognition using hashed chroma landmarks," in *Proceedings of IEEE WASPAA*. New Platz, NY: IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), 2011.

[17] Y. Bayle, L. Maršík, M. Rusek, M. Robine, P. Hanna, K. Slaninová, J. Martinovic, and J. Pokorný, "Kara1k: A karaoke dataset for cover song identification and singing voice analysis," in *2017 IEEE International Symposium on Multimedia (ISM)*, 2017, pp. 177–184.

[18] F. Yesiler, C. J. Tralie, A. A. Correya, D. F. Silva, P. Tovstogan, E. Gómez, and X. Serra, "Da-tacos: A dataset for cover song identification and understanding," in *ISMIR*, 2019.

[19] 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, 2020, deezer Research. [Online]. Available: https://doi.org/10.21105/joss.02154

[20] J. W. Kim, J. Salamon, P. Li, and J. P. Bello, "Crepe: A convolutional representation for pitch estimation," 2018. [Online]. Available: https://arxiv.org/abs/1802.06182

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015. [Online]. Available: http://arxiv.org/abs/1512.03385

[22] C. Frank, "The machine learning behind hum to search," https://ai.googleblog.com/2020/11/the-machine-learning-behind-hum-to.html, accessed: 2022-10-23.

[23] J.-S. R. Jang, "Qbsh: A corpus for designing qbsh (query by singing/humming) systems." [Online]. Available: http://www.cs.nthu.edu.tw/~jang

[24] J. Salamon, J. Serrà, and E. Gómez, "Tonal representations for music retrieval: From version identification to query-by-humming," *International Journal of Multimedia Information Retrieval, special issue on Hybrid Music Information Retrieval*, vol. 2, pp. 45–58, 03 2013.

[25] G. Doras and G. Peeters, "Cover detection using dominant melody embeddings," *CoRR*, vol. abs/1907.01824, 2019. [Online]. Available: http://arxiv.org/abs/1907.01824

[26] A. Jain, P. R. Samala, D. Mittal, P. Jyothi, and M. Singh, "Spliceout: A simple and efficient audio augmentation method," *CoRR*, vol. abs/2110.00046, 2021. [Online]. Available: https://arxiv.org/abs/2110.00046

[27] K. Sohn, "Improved deep metric learning with multi-class n-pair loss objective," in *Advances in Neural Information Processing Systems*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds., vol. 29. Curran Associates, Inc., 2016. [Online]. Available: https://proceedings.neurips.cc/paper/2016/file/6b180037abbebea991d8b1232f8a8ca9-Paper.pdf

[28] X. Wang, X. Han, W. Huang, D. Dong, and M. R. Scott, "Multi-similarity loss with general pair weighting for deep metric learning," 2019.

[29] "MIREX QBSH challenge results," https://www.music-ir.org/mirex/wiki/2021:Query_by_Singing/Humming, accessed: 2022-10-23.

[30] I. Smule, "DAMP-VPB: Digital Archive of Mobile Performances - Smule Vocal Performances Balanced," Nov. 2017. [Online]. Available: https://doi.org/10.5281/zenodo.2616690

[31] M. Müller, "Dynamic time warping," *Information Retrieval for Music and Motion*, vol. 2, pp. 69–84, 01 2007.

[32] "MIREX QBSH 2016 results," https://www.music-ir.org/mirex/wiki/2016:MIREX2016_Results, accessed: 2022-10-23.

TOWARDS IMPROVING HARMONIC SENSITIVITY AND PREDICTION STABILITY FOR SINGING MELODY EXTRACTION

Keren Shao* **Ke Chen*** **Taylor Berg-Kirkpatrick** **Shlomo Dubnov**

University of California San Diego

{k5shao, knutchen, tberg, sdubnov}@ucsd.edu

ABSTRACT

In deep learning research, many melody extraction models rely on redesigning neural network architectures to improve performance. In this paper, we propose an input feature modification and a training objective modification based on two assumptions. First, harmonics in the spectrograms of audio data decay rapidly along the frequency axis. To enhance the model's sensitivity on the trailing harmonics, we modify the Combined Frequency and Periodicity (CFP) representation using discrete z-transform. Second, the vocal and non-vocal segments with extremely short duration are uncommon. To ensure a more stable melody contour, we design a differentiable loss function that prevents the model from predicting such segments. We apply these modifications to several models, including MSNet, FTANet, and a newly introduced model, PianoNet, modified from a piano transcription network. Our experimental results demonstrate that the proposed modifications are empirically effective for singing melody extraction.

1. INTRODUCTION

Singing melody extraction is a challenging task that aims to detect and identify the fundamental frequency (F0) of singing voice in polyphonic music recordings. This task is more complicated than the monophonic pitch detection task due to the presence of various instrumental accompaniments and background noises, making it more difficult to accurately extract the singing melody. Singing melody extraction is not only crucial for music analysis by itself, but also has many downstream applications, such as cover song identification [1], singing evaluation [2], and music recommendation [3].

Deep neural networks have been widely adopted in the singing melody extraction task to produce promising performance in terms of extraction accuracy. Early models [4–6] simply leveraged deep neural networks (DNN) and convolutional neural networks (CNN) [7]. In more recent models, musical and structural priors were incorporated to improve performance. These include MSNet [8] with a vocal detection component at the encoder-decoder bottleneck, joint detection model [9] setting up an auxiliary network, and TONet [10] with tone-octave predictions. Additionally, models can capture frequency relationships better with multi-dilation [11], cross-attention networks [12], graph-based neural networks [13], or harmonic constant-Q transform (HCQT) [14].

One of our observations relates to the input representations of the models, which play an important role in affecting the extraction performance. Timbre, which is closely related to harmonics, is one of the key components that helps models distinguish the vocal from other instruments. When the CFP representation [15] is chosen as the input representation, its second feature, the generalized cepstrum, allows the model to learn the strength of harmonics of any given fundamental frequency in a localized manner. However, in music, the harmonics of a single sound usually decays rapidly along the frequency axis (detail in section 2.1), which can pose a challenge for the model to distinguish sounds that only differ significantly at the trailing harmonics.

The transformation from the spectrum to the generalized cepstrum in CFP is a Fourier transform, and hence mostly captures the first few peaks with large magnitudes. As a result, this representation is not helpful in sensing the trailing harmonics. This motivates us to apply a different transformation function that produces a generalized cepstrum with better harmonics sensitivity.

Another observation relates to the vocal detection component. Extremely short vocal segments surrounded by non-vocal regions, and vice versa, rarely occur since vocalists typically sing a melody for at least half a second or rest for at least a few hundred milliseconds. Threshold-based removal [16], mean or median filtering [17, 18] and Viterbi-based smoothing [19, 20] are frequently used to address the problem. When they are implemented alongside a network-based algorithm, however, the network remains unaware of our smoothing intention and configuration. To investigate whether such awareness can increase the prediction performance, we derive a differentiable loss component that specifically penalizes spurious short-term predictions of these kinds during training, thus potentially guiding the model to produce consistently stable predictions.

In this paper, we propose two techniques that attempt

*The first two authors have equal contribution.

© K. Shao, K. Chen, T. Berg-Kirkpatrick, S. Dubnov. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** K. Shao, K. Chen, T. Berg-Kirkpatrick, S. Dubnov, "Towards Improving Harmonic Sensitivity and Prediction Stability for Singing Melody Extraction", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

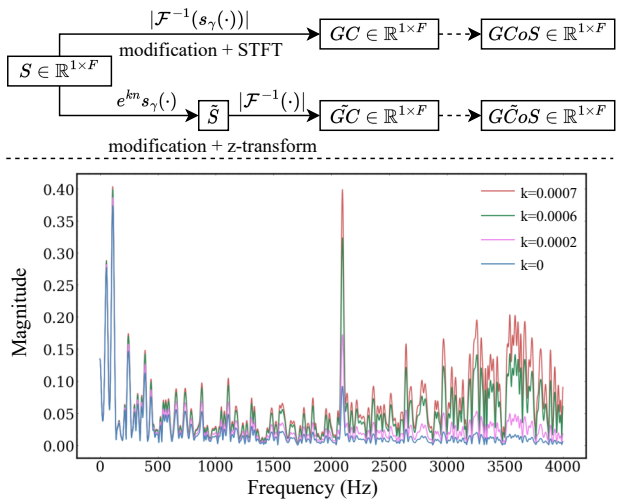


Figure 1. Top: the transformation pipeline of the original CFP representation, and our proposed z-CFP representation. Bottom: modified Spectrum $\tilde{S}$ with different growing rates k applied. Note that the original CFP corresponds to the case of $k = 0$.

to improve the two concerns mentioned above, namely the harmonic sensitivity and the prediction stability of singing melody extraction models. Our contributions are as follows:

- We propose to use exponentially growing sinusoids along the frequency axis to transform the spectrum into the generalized cepstrum of the CFP representation. This approach is equivalent to taking a z-transform instead of Fourier transform, which increases the harmonic sensitivity of the input.
- We design a differentiable loss function as part of the training objective to teach the network to avoid predicting unrealistically short sequences of vocal and non-vocal at the voice detection bin.
- We evaluate our techniques by applying them on several melody extraction models. Additionally, we adapt PianoNet [21], originally developed for piano transcription, into the melody extraction task. Experimental results demonstrate state-of-the-art performance of our improved models.

2. METHODOLOGY

In this section, we introduce three main parts of our methodology. First, we propose a modified CFP representation, z-CFP, to enhance the harmonic sensitivity of the network input. Second, we introduce extraction models used for evaluating our techniques, namely MSNet, FTANet, and PianoNet. Third, we propose a new loss function as part of training objective to improve the prediction stability of models.

2.1 z-CFP Representation for Harmonic Sensitivity

Our input representation of audio data is a modified version of the CFP representation. A CFP representation $X \in \mathbb{R}^{3\times T\times F}$ contains three features, with T the length of time frames and F the number of frequency bins. **At each time slice**, it contains: (1) a power spectrum $S \in \mathbb{R}^{1\times F}$; (2) a generalized cepstrum $GC \in \mathbb{R}^{1\times F}$; and (3) a generalized cepstrum of spectrum $GCoS \in \mathbb{R}^{1\times F}$,

As illustrated in the upper part of Figure 1, the standard CFP generation process begins by computing the frame-wise spectrum of an input audio waveform using short-time Fourier transform (STFT). We then obtain the magnitude of each spectrum, which serves as the first feature of CFP, denoted as S. To derive the second feature, we compute the generalized cepstrum using the following equation:

$$GC = |\mathcal{F}^{-1}(s_\gamma(S))| = |\mathcal{F}(s_\gamma(S))| \tag{1}$$

where $\mathcal{F}$ and $\mathcal{F}^{-1}$ denotes the Fourier transform and its inverse, $s_\gamma : \mathbb{R} \to \mathbb{R}$ is an element-wise applied, logarithm-like modification function as described in [15], and the absolute value sign represents an element-wise complex norm operation. The second equality comes directly from the fact that norm of a complex number equals to that of its conjugate.

As mentioned in the introduction, GC is not sensitive to the trailing harmonic dynamics, as it mostly captures the first few peaks with large magnitudes. Since the harmonics decay rapidly along with the frequency axis, we shall revert the decay to better preserve such dynamics. In other words, instead of applying complex sinusoids $\sum_n s_\gamma(S[n])e^{-iwn}$ as in Fourier transform (n is the entry of frequency bins in S), we apply growing complex sinusoids $\sum_n s_\gamma(S[n])e^{(k-iw)n}$, where $k \in \mathbb{R}$ and $k > 0$. This is equivalent to taking a discrete z-transform $\sum_n s_\gamma(S[n])z^{-n}$, where $z = e^{iw-k}$.

In the actual implementation, k is manually assigned and fixed across different w. Therefore, as illustrated in Figure 1, we can separate the computation of k part and w part as follows:

$$\tilde{S}[n] = e^{kn}s_\gamma(S[n]) \text{ for } \forall n \tag{2}$$

$$\tilde{GC} = |\mathcal{F}^{-1}(\tilde{S})| = |\mathcal{F}(\tilde{S})| \tag{3}$$

In the lower part of Figure 1, we present $\tilde{S}$ of an audio waveform with different values of k. We can observe that the harmonics of $\tilde{S}$ at the tail gets amplified so that the subsequent Fourier transform can better capture their dynamics. While we observe some amplifications of harmonics at frequencies other than the fundamental frequencies, their magnitudes are always smaller than those of nearby fundamental frequencies. Therefore, they pose no sufficient distraction for the extraction model, as long as the chosen k is not too large. In our experiments, we set $k = 0.0006$.

We then generate the generalized cepstrum of spectrum $\tilde{GCoS}$ from cepstrum $\tilde{GC}$ the same way as in the original CFP. Finally, **each time slice** of our modified CFP representation $\tilde{X} \in \mathbb{R}^{3\times T\times F}$ consists of $\{S, \tilde{GC}, \tilde{GCoS}\}$ with log-scaled frequency axis. For the rest of the paper, we denote it z-CFP.

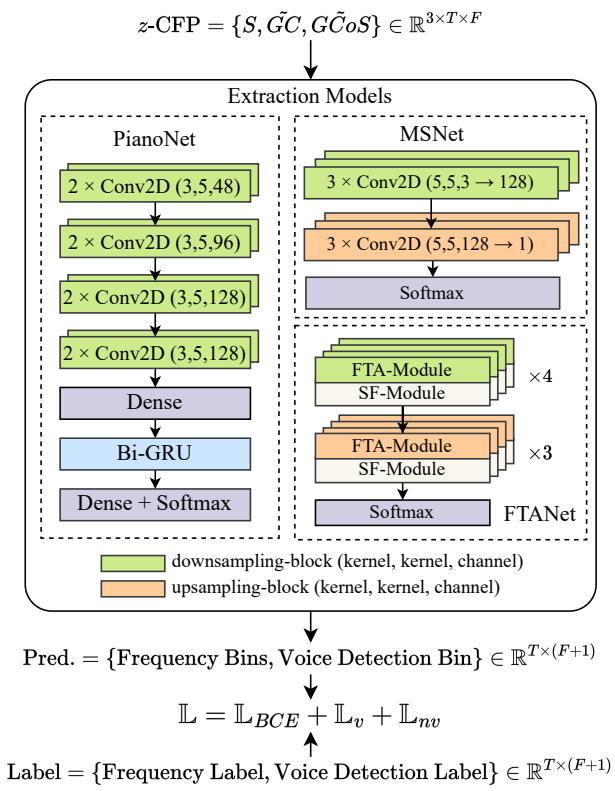


Figure 2. The model architecture. Note that we choose only one of the three extraction models at a time.

2.2 Model Architecture

Our extraction models are referred from three state-of-the-art (SoTA) models, MSNet [8], FTANet [12], and PianoNet [21]. Different from MSNet and FTANet, PianoNet is the SoTA model of piano transcription. Given its superior performance on piano transcription, we incorporate a sub-network of PianoNet into singing melody extraction, as we hypothesize that it may also yield good results for melody extraction.

MSNet contains a 3-layer encoder, a 3-layer decoder, and a bottleneck module. The channel size is shifted as $3 \to 32 \to 64 \to 128 \to 64 \to 32 \to 1$. The bottleneck module maps the encoder output to a 1-channel featuremap for voice detection. All 2D-convolutional layers come with (5×5) kernel size. FTANet contains a 4-layer encoder, a 3-layer decoder, and a 4-layer bottleneck module. Both encoder and decoder contain FTA-modules and SF-modules to process the audio latent features. The channel size is shifted from 3 to 128, then back to 1. More specifications of MSNet and FTANet can be found in their papers [8, 12].

The PianoNet we use for this task is modified from a sub-network of [21]. It starts with four convolutional blocks, each block containing two 2D-convolutional layers with kernel sizes (3, 5) and (3, 3) respectively, a batch normalization layer and a ReLU activation. Then it is followed by bidirectional-GRU and softmax layers, with dense layers as transitions. The layer bias is turned off for all layers before the Bi-GRU.

Figure 2 illustrates a more detailed structure of the three extraction models. Following the pipeline, we first process the audio waveform into z-CFP representations. Then we feed them into the extraction model, which produces output feature maps $\tilde{Y} \in \mathbb{R}^{T\times(F+1)}$. The additional one feature along the frequency axis denotes the voice detection bin output. It is then compared against the ground truth label $Y \in \mathbb{R}^{T\times(F+1)}$, through the loss function introduced in the following section.

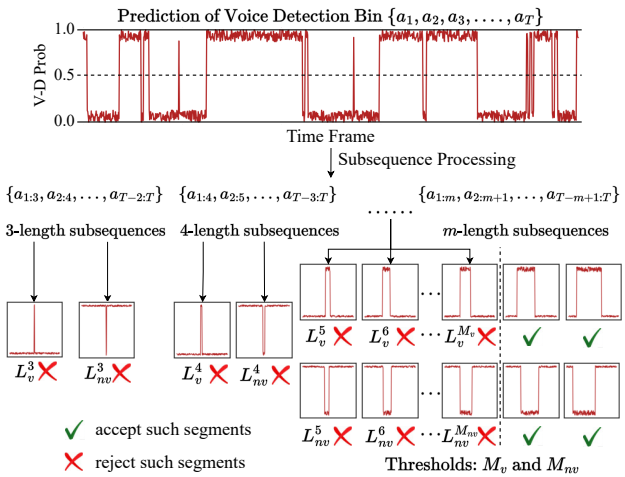


Figure 3. The illustration of how we perform the loss functions $\mathbb{L}_v$ and $\mathbb{L}_{nv}$ on the subsequences of the voice detection prediction. Each loss components L are used to give large penalties (i.e., ✗) to certain types of subsequences.

2.3 Loss Function for Prediction Stability

We add two differentiable training objectives, $\mathbb{L}_v$ and $\mathbb{L}_{nv}$, to the conventional binary cross entropy loss $\mathbb{L}_{BCE}$ to teach the extraction model to avoid unrealistically short vocal and non-vocal sequences at **the vocal detection bin**. Since the design for these two cases are symmetric, we first introduce the loss object $\mathbb{L}_v$, for the vocal case.

As shown on the top of Figure 3, the predictions at the vocal detection bin is a time series $\{a_1, a_2, ..., a_T\}$. First, since our training objectives are dealing with certain types of short burst segments of vocal and non-vocal, we extract all possible subsequences, with stride 1. For example, for 3-length subsequences we have $\{a_{1:3}, a_{2:4}, ..., a_{T-2:T}\}$, and similarly $\{a_{1:4}, a_{2:5}, ..., a_{T-3:T}\}$ for subsequences of length 4, etc.

Second, to simplify the problem a bit at the beginning, we assume that the voice detection output is binary valued $a \in \{0, 1\}$. Formally, we do not want "sharp-burst" sequences inside the following set:

$$B_v = \bigcup_{m=3}^{M_v} \{a_1 ... a_m | a_1 = a_m = 0, a_i = 1 \text{ for } \forall i \neq 1, m\} \quad (4)$$

where M_v is a hyperparameter threshold, above which the duration of vocal segments becomes reasonable. Figure 3 illustrates examples of "sharp-burst" sequences in B_v (and B_{nv}) as red segments inside black-border boxes.

Suppose $m = 3$, all possible binary sequences are $\{000, 001, 010, 011, 100, 101, 110, 111\}$ and $010 \in B_v$. To make the model avoid predicting the short burst vocal

segment, i.e., 010, we construct a polynomial objective that can fulfill the goal by satisfying the following:

$$L_v^3(a_1a_2a_3) = \begin{cases} 1 & \text{where } a_1a_2a_3 = 010 \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

A decent choice will then be

$$L_v^3(a_1a_2a_3) = (1-a_1)a_2(1-a_3) \tag{6}$$

which can be easily extended to sequences with longer length m.

$$\begin{aligned} L_v^4(a_1a_2a_3a_4) &= (1-a_1)a_2a_3(1-a_4) \\ &\vdots \\ L_v^m(a_1...a_m) &= (1-a_1)(1-a_m)\prod_{i=2}^{m-1} a_i \end{aligned} \tag{7}$$

However, there is a small caveat in this extension when we move back from binary values to probability values $a \in [0,1]$. For example, our loss component will be having trouble capturing sequences like $\{0.1, 0.4, 0.6, 0.1\}$ and $\{0.1, 0.6, 0.4, 0.1\}$ as both L_v^3 and L_v^4 result in relatively small values. However, we observe that polynomials $(1-a_1)(1-a_2)a_3(1-a_4)$ and $(1-a_1)a_2(1-a_3)(1-a_4)$ respectively works better than our original L_v^4, but still insufficient to work standalone.

Since none of the polynomials above gives high values to sequences outside of B_v in 4-length, a simple solution would be to redefine L_v^4 to be the sum of all such polynomials:

$$\begin{aligned} L_v^4 &= (1-a_1)(1-a_4)(a_2a_3 + a_2(1-a_3) + (1-a_2)a_3) \\ &\vdots \\ L_v^m &= (1-a_1)(1-a_m)\sum_{\substack{c_1...c_m\in\{0,1\}^m \\ \text{at least one } c_i\neq 0}} \prod_{i=2}^{m-1} a_i^{c_i}(1-a_i)^{1-c_i} \\ &= (1-a_1)(1-a_m)(1-\prod_{i=2}^{m-1}(1-a_i)) \end{aligned} \tag{8}$$

This redefined loss L_v allows better recognition of the bad sequences mentioned above while not falsely flagging sequences outside of B_v. Furthermore, when dealing with longer sequences, for example $\{0.1, 0.9, ..., 0.9, 0.1\}$ with increasingly many 0.9s in the middle, the original L_v's output quickly diminishes while the redefined L_v does not.

This redefined objective does come with a small side effect, as it over-counts the shorter bad sequences. For example, $(0.1, 0.9, 0.1, 0.1)$ now gets a high loss value not only in L_v^3, but also in L_v^4. However, we believe this side effect does not have significant impact as it does not matter whether neural network decides to stop producing shorter bad sequences or longer bad sequences first.

A further improvement is to pass the value of L_v^m into the S-curve function:

$$L_v^m \leftarrow \frac{(L_v^m)^r}{(L_v^m)^r + (1-L_v^m)^r} \tag{9}$$

where $r \in \mathbb{R}$ and $r > 1$. It will amplify those sequences that receive loss values closer to 1 and suppress those sequences with loss values closer to 0.

Finally, for each $m \in [3, M_v]$, we compute L_v^m across all m-length subsequences in the model's output. The aggregated loss function $\mathbb{L}_v$ is then computed by concatenating all these L_v^m arrays and taking the average.

Now analogously, assuming non-vocal sequences beyond length M_{nv} become reasonable, we can perform the same analysis on the following set of sequences:

$$B_{nv} = \bigcup_{m=3}^{M_{nv}} \{a_1...a_m | a_1 = a_m = 1, a_i = 0 \text{ for } \forall i \neq 1, m\} \tag{10}$$

and consequently obtain $\mathbb{L}_{nv}$. Practically, L_{nv}^m of any sequence $a_1...a_m$ can be computed as L_v^m of the flipped sequence $b_1...b_m$, where $b_i = 1 - a_i$ for all $i \in \{1..m\}$. Our final loss function will then be:

$$\mathbb{L} = \mathbb{L}_{BCE} + \mathbb{L}_v + \mathbb{L}_{nv} \tag{11}$$

3. EXPERIMENTS

3.1 Datasets and Experiment Setup

For the training data, we complied with the setting of [10, 12] and chose all 1000 Chinese pop songs from MIR-1K [1] and 35 vocal tracks from MedleyDB [22]. For the testing data, we chose 12 tracks in ADC2004 and 9 tracks in MIREX05 [2]. We also selected 12 tracks from MedleyDB that are disjoint from those already used for training.

For the signal processing part, we used 8000 Hz sampling rate to process audio tracks. We use a window size of 768, a hop size of 80 to compute the STFT of audio tracks. Note that the time resolution of our labels is 0.01 seconds, and this hop size was chosen to match that. Then, when creating z-CFP representations, we set the time dimension of the representation to be $T = 128$, or 1.28 seconds, and the number of frequency bins $F = 360$, or 60 bins per octave across 6 octaves. The start and stop frequencies are 32.5 Hz and 2050 Hz. Hence, the input shape becomes $X \in \mathbb{R}^{3\times128\times360}$ and the output/label shape becomes $Y \in \mathbb{R}^{128\times361}$.

Within the extra loss component, we set the duration threshold of vocal segments $M_v = 30$ (0.3 seconds), the duration threshold of non-vocal segments $M_{nv} = 7$ (0.07 seconds), and the S-curve exponent parameter $r = 5$.

For the training hyperparameters, we use a batch size of 10, the Adam optimizer [23] with a fixed learning rate of 1×10^{-4}. The maximum training epoch is 500. During the evaluation, we use the standard metrics of the singing melody extraction task, namely, voice recall (VR), voicing false alarm (VFA), raw pitch accuracy (RPA), raw chroma accuracy (RCA), and overall accuracy (OA) from the `mir_eval` library [24]. Following the convention of this task, overall accuracy (OA) is regarded as the most important metric. All models are trained and tested in NVIDIA RTX 2080Ti GPUs and implemented in PyTorch [3].

[1] http://mirlab.org/dataset/public/MIR-1K.zip
[2] https://labrosa.ee.columbia.edu/projects/melody/
[3] https://pytorch.org/

Dataset	ADC 2004					MIREX 05					MEDLEY DB				
Metrics	VR	VFA↓	RPA	RCA	OA	VR	VFA↓	RPA	RCA	OA	VR	VFA↓	RPA	RCA	OA
PianoNet	87.21	14.62	84.28	84.30	84.48	91.98	6.14	86.54	86.55	89.19	69.38	13.74	61.81	62.80	73.70
PianoNet + z-CFP	88.25	**7.58**	84.87	84.93	86.27	**93.44**	6.21	86.78	86.79	89.33	68.76	**11.91**	62.22	63.10	**74.80**
PianoNet + 3 point median	87.33	14.58	84.35	84.38	84.55	92.08	6.15	86.60	86.62	89.23	69.49	13.77	61.86	62.86	73.71
PianoNet + 7 point median	87.58	14.53	84.46	84.48	84.65	92.47	6.14	86.78	86.8	89.35	69.71	13.83	61.92	62.91	73.71
PianoNet + 15 point median	89.13	14.21	84.89	84.91	85.06	93.27	6.58	86.82	86.84	89.21	70.31	14.43	61.91	62.90	73.42
PianoNet + $\{\mathbb{L}_v, \mathbb{L}_{nv}\}$	**90.92**	13.58	**86.06**	**86.12**	86.13	91.87	**5.79**	87.50	87.50	**89.94**	**71.16**	15.77	**63.66**	**64.81**	73.66
PianoNet + z-CFP + $\{\mathbb{L}_v, \mathbb{L}_{nv}\}$	90.50	7.99	85.76	85.82	**86.92**	92.84	6.39	**87.57**	**87.59**	89.76	68.88	12.29	62.05	62.91	74.53
MSNet	89.78	23.12	80.83	81.60	80.10	84.85	**11.44**	77.76	78.09	81.68	53.49	**9.41**	46.90	48.24	68.15
MSNet + z-CFP + $\{\mathbb{L}_v, \mathbb{L}_{nv}\}$	**90.61**	**14.62**	**81.96**	**82.57**	**82.59**	**88.38**	14.85	**80.83**	**81.01**	**82.39**	**62.95**	14.60	**53.60**	**55.31**	**69.07**
FTANet	81.26	**2.70**	77.17	77.36	80.89	87.34	**5.11**	81.56	81.61	86.40	62.44	10.41	55.94	56.58	72.30
FTANet + z-CFP + $\{\mathbb{L}_v, \mathbb{L}_{nv}\}$	**90.29**	10.83	**85.06**	**85.19**	**85.82**	**90.50**	6.63	**83.94**	**83.99**	**87.36**	**63.71**	**9.35**	**56.32**	**57.29**	**73.02**

Table 1. Ablation studies on ADC2004, MIREX05 and MedleyDB testsets. Baselines use CFP as the input representation and $\mathbb{L}_{BCE}$ as the loss function. $\{\mathbb{L}_v, \mathbb{L}_{nv}\}$ denotes the use of our proposed loss function in section 2.3. Among median filter sizes in the range $[3, 100] \subset \mathbb{Z}$, 3 point works best for MedleyDB, 7 point works best for MIREX 05, and 15 point works best for ADC 2004. But they neither significantly outperform our proposed loss component in any single dataset, nor uniformly outperform in all three datasets.

3.2 Ablation Study

We choose three extraction models, namely MSNet [8], FTANet [12], and PianoNet [21], to evaluate our z-transform and loss functions. We conducted ablation studies and presented the results in Table 1. We re-trained these models from scratch, and the results are largely consistent with the original reports of [8, 10, 12]. The option z-transform denotes the use of z-CFP representations. Note that $\{\mathbb{L}_v, \mathbb{L}_{nv}\}$ in the table denote the use of loss functions to address short burst segments of vocal and non-vocal. Due to the page limitation, we present a detailed ablation study on PianoNet while ablating MSNet and FTANet in an all-or-nothing fashion.

From Table 1 we can clearly observe decent performance of both z-CFP and $\{\mathbb{L}_v, \mathbb{L}_{nv}\}$ when added to the PianoNet, MSNet, and FTANet. Among these results, the addition of loss functions $\{\mathbb{L}_v, \mathbb{L}_{nv}\}$ increases the overall accuracy while improving the VR, RPA, and RCA. The median filter postprocessing [18] is used as a comparison. Since our loss component focuses on the vocal detection, we took the pitches predicted by median filters only when the original predictions are non-vocal. Further, to ensure fairness, we optimized the filter size against each single dataset within the range $[3, 100] \subset \mathbb{Z}$ and listed the evaluation results of those optimal ones. As we can see in Table 1, none of these median filters outperforms our loss component in a consistent manner, nor do they obtain considerable margins in any single dataset.

The z-CFP also increases several metrics, especially either VR or VFA, on each dataset. This indicates that by preserving more dynamics in the high frequency bins, the model can distinguish different sounds better and consequently improve the extraction performance. Also, note that unlike TONet [10] and JDC [9], which achieved this through model design or music inductive bias, this technique relies solely on the inherent characteristics of the data.

When we incorporate both techniques into the extraction models, we observe a promising increase in each metric compared to the original models. However, we notice that some models with both techniques carried do not yield better performance than the models carrying only one of the techniques. These models appear to be an averaging weighting or an ensemble of models improved with either technique, implying better generalization.

3.3 Comprehensive Performance Comparison

Table 2 presents the results as we compare our best model, i.e., PianoNet with z-transform and $\{\mathbb{L}_v, \mathbb{L}_{nv}\}$, with other SoTA models. Among these SoTAs, there are two models with "*", indicating that these are only partial comparisons. For SpecTNT [25], since there is no official open-source implementation, we report its results based on our own re-implementation. For H-GNN [13], we directly copied its reported performance from the original paper.

From Table 2, our improved PianoNet with z-transform and $\{\mathbb{L}_v, \mathbb{L}_{nv}\}$ yield the best OA performance over all datasets, the best RPA and RCA on ADC 2004 and MIREX 05 datasets. We do note, despite the use of the extra loss component, that our model's VFA is not necessarily the smallest. This is because the extra loss component only targets a particular type of false positive, and is not meant to minimize the false positive rate in general. For example, sometimes the network's vocal to non-vocal transition happens later than the reference labels. In this case, since the vocal sequence itself lasts long enough, the extra loss component will not mark this type of false positives. Addressing this type of errors is potentially a future work.

Another thing we found is that the PianoNet, as one of SoTAs in the piano transcription task and ported by us to the melody extraction task in this paper, has already yields very high performance on MIREX 05 dataset. This indicates that there may exist more powerful network architectures for this task yet to be explored. Additionally, it is

Dataset	ADC 2004				
Metrics	VR	VFA↓	RPA	RCA	OA
MCDNN [4]	65.0	10.5	61.6	63.1	66.4
DSM [14]	89.2	51.3	75.4	77.6	69.8
MSNet [8]	89.8	23.1	80.8	81.6	80.1
FTANet [12]	81.3	**2.7**	77.2	77.4	80.9
TONet [10]	**91.8**	17.1	82.6	82.9	82.6
SpecTNT* [25]	85.4	8.2	83.5	83.6	85.0
H-GNN* [13]	89.2	21.3	84.8	86.1	83.9
Ours	90.5	8.0	**85.7**	**85.8**	**86.9**
Dataset	MIREX 05				
Metrics	VR	VFA↓	RPA	RCA	OA
MCDNN [4]	66.5	**4.6**	64.1	64.4	75.4
DSM [14]	91.4	45.3	75.7	77.0	68.4
MSNet [8]	84.8	11.4	77.8	78.1	81.7
FTANet [12]	87.3	5.1	81.6	81.6	86.4
TONet [10]	91.6	8.5	83.8	84.0	86.6
SpecTNT* [25]	82.2	8.7	77.4	77.5	82.5
H-GNN* [13]	**93.2**	21.7	85.2	86.4	81.3
Ours	92.8	6.4	**87.6**	**87.6**	**89.8**
Dataset	MEDLEY DB				
Metrics	VR	VFA↓	RPA	RCA	OA
MCDNN [4]	37.4	**5.3**	34.2	35.3	62.3
DSM [14]	**86.6**	44.3	**70.2**	**72.4**	64.8
MSNet [8]	53.5	9.4	46.9	48.2	68.1
FTANet [12]	62.4	10.4	55.9	56.6	72.3
TONet [10]	64.2	12.5	56.6	58.0	71.6
SpecTNT* [25]	62.7	18.8	54.7	56.4	63.9
H-GNN* [13]	71.7	21.6	61.2	65.8	67.9
Ours	68.9	12.3	62.1	62.9	**74.5**

Table 2. The comprehensive performance comparison among our improved models and current baselines.

noteworthy that our proposed PianoNet architecture has a small number of parameters (5.5 million), which is comparable with MCDNN (5.6 million), FTANet (3.4 million) and far less than TONet (152 million). This demonstrates its potential in practical applications where computational resources are limited. Again, as demonstrated in Table 1, our techniques could help models other than PianoNet achieve higher performance than their original versions.

3.4 Loss Value and Extraction Visualization

To empirically verify if applying the polynomial loss functions $\mathbb{L}_v$ and $\mathbb{L}_{nv}$ could reduce the voice detection errors, i.e., short burst segments of vocal and non-vocal, we visualize two types of plots in Figure 4. The top three plots demonstrate the loss values of L_v^{30} between the original extraction models and the improved models with $\mathbb{L}_v$ and $\mathbb{L}_{nv}$, across the entire MIREX05 dataset (i.e., we concatenate all tracks in the dataset). We see that cases in which the improved models' prediction receive loss values close to 1 diminishes comparing to those of the original models. This phenomenon implies that after applying $\mathbb{L}_v$ and $\mathbb{L}_{nv}$, the chance of models to predict short burst segments

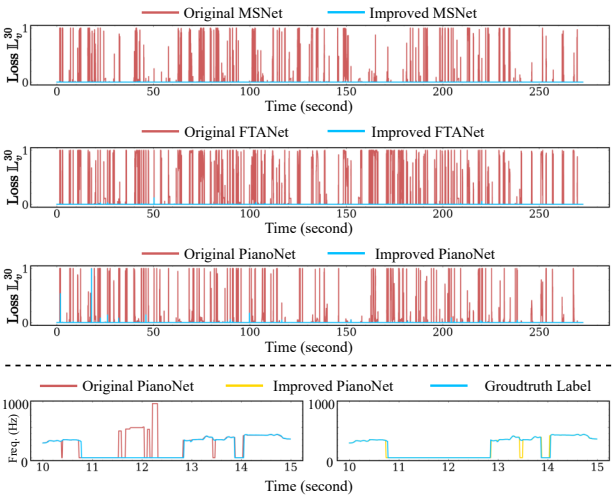


Figure 4. The effect of applying the loss $\mathbb{L}_v$ and $\mathbb{L}_{nv}$. The top three plots are values of L_v^{30} across the entire MIREX05 dataset. The bottom two plots are one 5-sec MIREX05 predictions.

significantly reduces.

The pair of plots in the last row compares the prediction performance of PianoNets, trained without and with the extra loss components, on a zoomed-in section of MIREX05. Note that the original PianoNet has a short burst non-vocal segment in between the 10th second and 11th second. Further, it has a considerable number of short burst vocal segments around the 12th second. Once trained with the extra loss components, these issues are resolved. Also note that both the original version and the improved version make a mistake in between the 13th and the 14th second. This is because the length of that non-vocal transition is greater than our threshold M_{nv}, which ends up not triggering $\mathbb{L}_{nv}$. All these observations further verify the effectiveness of our proposed loss components.

4. CONCLUSION

In this paper, we propose two techniques to respectively utilize the two assumptions we made for improving the performance of singing melody extraction models. First, comparing to Fourier transform, the use of z-transform in generating cepstrum allows the network to better recognize the strength of harmonics of any fundamental frequencies. Empirically, while the trailing harmonics of those frequencies that do not actually appear in the audio also get elevated, the benefit of the technique is greater than its setback. Second, our extra loss components make the network less prone to predict vocal and non-vocal sequences are unreasonably short, while not affecting the network's overall accuracy due to its differentiability. Along with different extraction models, we achieve better performance when compared to their original version and other state-of-the-art models. We regard these two techniques as decent improvements on singing melody extraction models.

5. ACKNOWLEDGMENTS

We would like to thank the Institute for Research and Coordination in Acoustics and Music (IRCAM) and Project REACH: Raising Co-creativity in Cyber-Human Musicianship for supporting this project. This project has received funding from the European Research Council (ERC REACH) under the European Union's Horizon 2020 research and innovation programme (Grant Agreement #883313).

6. REFERENCES

[1] X. Du, K. Chen, Z. Wang, B. Zhu, and Z. Ma, "Bytecover2: Towards dimensionality reduction of latent embedding for efficient cover song identification," in *Proc. ICASSP*, 2022, pp. 616–620.

[2] N. Zhang, T. Jiang, F. Deng, and Y. Li, "Automatic singing evaluation without reference melody using bidense neural network," in *Proc. ICASSP*, 2019, pp. 466–470.

[3] K. Chen, B. Liang, X. Ma, and M. Gu, "Learning audio embeddings with user listening data for content-based music recommendation," in *Proc. ICASSP*, 2021, pp. 3015–3019.

[4] S. Kum, C. Oh, and J. Nam, "Melody extraction on vocal segments using multi-column deep neural networks," in *Proc. ISMIR*, 2016, pp. 819–825.

[5] S. Li, "Vocal melody extraction using patch-based cnn," in *Proc. ICASSP*, 2018, pp. 371–375.

[6] J. W. Kim, J. Salamon, P. Li, and J. P. Bello, "Crepe: A convolutional representation for pitch estimation," in *Proc. ICASSP*. IEEE, 2018, pp. 161–165.

[7] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, 1998.

[8] T.-H. Hsieh, L. Su, and Y.-H. Yang, "A streamlined encoder/decoder architecture for melody extraction," in *Proc. ICASSP*, 2019, pp. 156–160.

[9] S. Kum and J. Nam, "Joint detection and classification of singing voice melody using convolutional recurrent neural networks," *Applied Sciences*, 2019.

[10] K. Chen, S. Yu, C. Wang, W. Li, T. Berg-Kirkpatrick, and S. Dubnov, "Tonet: Tone-octave network for singing melody extraction from polyphonic music," in *Proc. ICASSP*, 2022, pp. 626–630.

[11] P. Gao, C. You, and T. Chi, "A multi-dilation and multi-resolution fully convolutional network for singing melody extraction," in *Proc. ICASSP*, 2020, pp. 551–555.

[12] S. Yu, X. Sun, Y. Yu, and W. Li, "Frequency-temporal attention network for singing melody extraction," in *Proc. ICASSP*, 2021, pp. 251–255.

[13] S. Yu, X. Chen, and W. Li, "Hierarchical graph-based neural network for singing melody extraction," in *Proc. ICASSP*, 2022, pp. 626–630.

[14] R. M. Bittner, B. McFee, J. Salamon, P. Li, and J. P. Bello, "Deep salience representations for f0 estimation in polyphonic music." in *Proc. ISMIR*, 2017, pp. 63–70.

[15] L. Su and Y.-H. Yang, "Combining spectral and temporal representations for multipitch estimation of polyphonic music," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 23, no. 10, pp. 1600–1612, 2015.

[16] R. M. Bittner, J. Salamon, J. J. Bosch, and J. P. Bello, "Pitch contours as a mid-level representation for music informatics," in *Audio engineering society conference: 2017 AES international conference on semantic audio*, 2017.

[17] J. Salamon and E. Gómez, "Melody extraction from polyphonic music signals using pitch contour characteristics," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 6, pp. 1759–1770, 2012.

[18] S. Rosenzweig, F. Scherbaum, and M. Müller, "Detecting stable regions in frequency trajectories for tonal analysis of traditional georgian vocal music." in *Proc. ISMIR*, 2019, pp. 352–359.

[19] M. Mauch and S. Dixon, "pyin: A fundamental frequency estimator using probabilistic threshold distributions," in *Proc. ICASSP*. IEEE, 2014, pp. 659–663.

[20] J. J. Bosch and E. Gómez Gutiérrez, "Melody extraction based on a source-filter model using pitch contour selection," in *Proceedings SMC 2016. 13th Sound and Music Computing Conference*, 2016.

[21] Q. Kong, B. Li, X. Song, Y. Wan, and Y. Wang, "High-resolution piano transcription with pedals by regressing onset and offset times," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 29, pp. 3707–3717, 2021.

[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. ISMIR*, 2014, pp. 155–160.

[23] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Proc. ICLR*, 2014.

[24] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, L. Dawen, D. P. Ellis, and C. C. Raffel, "mir_eval: A transparent implementation of common mir metrics," in *Proc. ISMIR*, 2014, pp. 367–372.

[25] W. T. Lu, J. Wang, M. Won, K. Choi, and X. Song, "Spectnt: a time-frequency transformer for music audio," in *Proc. ISMIR*, 2021, pp. 396–403.

Papers – Session VI

SINGING VOICE SYNTHESIS USING DIFFERENTIABLE LPC AND GLOTTAL-FLOW-INSPIRED WAVETABLES

Chin-Yun Yu **György Fazekas**
Centre for Digital Music, Queen Mary University of London, UK
chin-yun.yu@qmul.ac.uk, george.fazekas@qmul.ac.uk

ABSTRACT

This paper introduces GlOttal-flow LPC Filter (GOLF), a novel method for singing voice synthesis (SVS) that exploits the physical characteristics of the human voice using differentiable digital signal processing. GOLF employs a glottal model as the harmonic source and IIR filters to simulate the vocal tract, resulting in an interpretable and efficient approach. We show it is competitive with state-of-the-art singing voice vocoders, requiring fewer synthesis parameters and less memory to train, and runs an order of magnitude faster for inference. Additionally, we demonstrate that GOLF can model the phase components of the human voice, which has immense potential for rendering and analysing singing voice in a differentiable manner. Our results highlight the effectiveness of incorporating the physical properties of the human voice mechanism into SVS and underscore the advantages of signal-processing-based approaches, which offer greater interpretability and efficiency in synthesis.

1. INTRODUCTION

Singing voice synthesis (SVS) has attracted substantial interest as a research topic over the last decades, and a variety of techniques have been developed. Early successful SVS systems were usually based on sample concatenation [1–4], while parametric systems have become much more prevalent. The actual synthesis process in parametric systems is carried out by a *vocoder* controlled by synthesis parameters generated from a separate acoustic model given some musical context factors (i.e. note number, duration, phoneme, etc.). Early systems of this kind use a linear source-filter model as vocoder [5, 6]. Deep Neural Networks (DNNs) have subsequently become the dominant approach for state-of-the-art vocoders [7–13]. However, mel-spectrograms are often chosen as input features to these models, which are less interpretable than traditional vocoder parameters (e.g. f0, aperiodicity ratios). Also, a significant amount of data is needed to cover various vocal expressions to achieve generalisation.

© C.-Y. Yu and G. Fazekas. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** C.-Y. Yu and G. Fazekas, "Singing Voice Synthesis Using Differentiable LPC and Glottal-Flow-Inspired Wavetables", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

In contrast, Differentiable Digital Signal Processing (DDSP) models [14–16] incorporate existing signal processing operations into neural networks as an inductive bias, making them more interpretable and generalisable. DDSP additive synthesis has been proposed for SVS by Alonso et al. [17]. Wu et al. [18] improved this further by using subtractive synthesis and sawtooth as the harmonic source. Nercessian et al. [19] proposed a differentiable version of the WORLD vocoder [20] for doing end-to-end singing voice conversion. Yoshimura et al. [21] used Taylor expansion to approximate the mel-log spectrum approximation filter's (MLSA) exponential function and embedded it into an SVS system. However, most of their architectures only assume the target signal is a monophonic instrument, which can potentially lead to solutions that do not reflect some properties of voice. In their design, the harmonic sources are fixed to a specific shape (e.g. sawtooth, pulse train), and the filters are symmetric in the time domain, except Yoshimura et al. [21] which use a minimum-phase MLSA filter. Incorporating constraints specific to the human voice on the harmonic source and the filters could lead to a more interpretable and compact SVS vocoder.

In this work, we propose GlOttal-flow LPC Filter (GOLF), an SVS module informed by the physical properties of the human voice. We build upon the Harmonic-plus-Noise architecture of DDSP [14] and the subtractive synthesis of SawSing [18], but replace the harmonic source with a glottal model and use IIR filters. We developed a differentiable IIR implementation in PyTorch [22] for training efficiency. We then used this module as a neural vocoder and compared its performance with other DDSP-based vocoders. Specifically, a simple and lightweight NN encoder converts the mel-spectrogram into synthesis parameters, and the synthesiser decodes the signal from it. We paired different synthesisers with the same encoder and trained them jointly.

Our contributions are twofold. First, GOLF has significantly fewer synthesis parameters but is still competitive with state-of-the-art SVS vocoders. Second, GOLF requires less than 40% of memory to train and runs ten times faster than its alternatives for inference. Moreover, we indirectly show that GOLF could model the phase components of the human voice by aligning the synthesised waveforms to the ground truth and calculating the differences. This characteristic has excellent potential for analysing singing voice in a differentiable manner. Decomposing the

human voice into the glottal source and vocal tract could also enable us to adjust the singing style in different ways, such as altering the amount of vocal effort with varying shapes of the glottal pulse.

2. BACKGROUND

We first introduce the relevant notation. $\mathbf{x}_i$ denotes the ith column vector and $x_{i,j}$ denotes the entry at the ith row and the jth column of the matrix $\mathbf{X}$. Concatenating two matrices along the column dimension is denoted by $[;]$. x_i denotes the ith entry of the vector $\mathbf{x}$ or a time sequence x indexed by i. $X(z)$ denotes the response of x_n in the z-domain. Unless stated otherwise, we use n as the time index and k as the frame index. Angular frequencies and periods are normalised to the interval $[0, 1]$. We use one-based indexing for elements with finite dimensions.

2.1 Glottal Source-Filter Model

In the source-filter model, we have the following simplified voice production model:

$$S(z) = (G(z) + N(z))\, H(z)L(z), \tag{1}$$

where $G(z)$ represents the periodic vibration from the vocal folds, $N(z)$ represents random components of the glottal source, $H(z)$ represents the vocal-tract filter, and $L(z)$ represents the radiation at the lips [23]. Since this formulation is linear, the radiation filter $L(z)$ and the glottal pulse $G(z)$ can be merged into a single source $G'(z)$ called *the radiated glottal pulse*. If we assume $L(z)$ is a first-order differentiator $1 + z^{-1}$ [24], then $G'(z)$ is the derivative of the glottal pulse, which can be described by the LF model [25], a four-parameter model of glottal flow. $H(z)$ is usually a Linear Predictive Coding (LPC) filter.

2.2 Linear Predictive Coding

LPC assumes that the current speech sample s_n can be predicted from a finite number of previous M samples s_{n-1} to s_{n-M} by a linear combination with residual errors e_n:

$$s_n = e_n - \sum_{i=1}^{M} a_i s_{n-i}, \tag{2}$$

where a_i are the linear prediction coefficients. This is the same as filtering the residuals, equivalent to the glottal source in our case, with an Mth-order all-pole filter, a filter that has an infinite impulse response (IIR). We can use the LPC filter to represent the response of the vocal tract if the vocal tract is approximated by a series of cylindrical tubes with varying diameters [26], providing a physical interpretation.

Using LPC for neural audio synthesis is not new [8, 27, 28], and works have been conducted to incorporate IIR filters and train them jointly with deep learning models [29–35]. The difficulty of training IIR in deep learning framework (e.g. PyTorch) using Eqn (2) is that its computation is recursive, i.e. the output at each step depends on the previous results, and to make the calculation differentiable, separated tensors are allocated in each step. This generates a significant number of memory allocations and overheads for creating tensors, thus leading to performance issues, especially for long sequences. One way to mitigate this is to allocate shared continuous memory before computation. However, in-place modification is not differentiable in these frameworks. Some studies sidestep the recursion by approximating IIR in the frequency domain using Discrete Fourier Transform (DFT) [27, 30, 32–35], but the accuracy of this approximation depends on the DFT resolution. Moreover, the IIRs used in practice are usually low-order; in this case, it is faster to compute them directly, especially on long sequences.

3. PROPOSED MODEL

Usually, $N(z)$ in Eqn (1) is treated as amplitude-modulated Gaussian noise [23, 24]. Our early experiments found this formulation to be challenging to optimise. As an alternative, we move the noise components $N(z)$ outside the glottal source and filter it with time-varying filter $C(z)$, resulting in

$$S(z) = G'(z)H(z) + N(z)C(z). \tag{3}$$

This resembles the classic *Harmonic-plus-Noise* model [36] and was used in previous DDSP-based SVS [17, 18]. Alonso et al. [17] modelled $G'(z)H(z)$ jointly using additive harmonic oscillators and time-varying Finite Impulse Responses (FIRs) as $C(z)$; Wu et al. [18] introduced a sawtooth oscillator as $G'(z)$ and zero-phase time-varying FIRs as $H(z)$. In this work, we use a glottal flow model to synthesise harmonic sources and time-varying IIRs as filters.

3.1 Glottal Flow Wavetables

We adopted the transformed-LF model [37] for generating glottal pulses. This model re-parameterises the LF model [25] using just one parameter R_d, which has been found to correspond to the perceived vocal effort well and covers a wide range of different glottal flow shapes. We sampled K values of $\log(R_d)$ with equal spacing inside $[\log(0.3), \log(2.7)]$ according to the value range suggested by [23]. We calculate the flow derivative function $g'(t; R_d)$ in continuous time t for each sampled R_d and then sampled L points in one period to get its discrete version. The details for calculating $g'(t; R_d)$ were given in [38]. By stacking these sampled glottal flows, we built wavetables $\mathbf{D} \in \mathbb{R}^{K \times L}$, with each row containing one period of a sampled glottal pulse (see Fig. 1). The rows are sorted based on R_d.

The model generates glottal pulses g'_n by linearly interpolating the two $\mathbf{D}$ axes. The encoder network first predicts instantaneous frequency $f_n \in [0, 0.5]$ and the fractional index $\tau_n \in [0, 1]$ for R_d. We then use the instantaneous phase $\phi_n = \sum_{i=1}^{n} f_i$ to interpolate the waveform

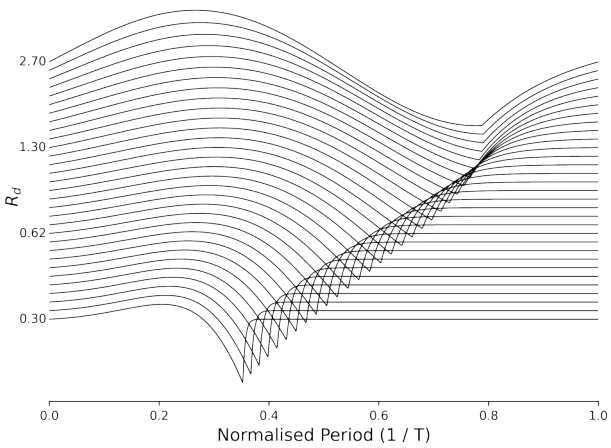


Figure 1. An example of the wavetables we used, corresponding to matrix $\mathbf{D}$ with $K = 31$.

as:

$$g'_n = (1-p)\left((1-q)\hat{d}_{\lfloor k \rfloor,\lfloor l \rfloor} + q\hat{d}_{\lfloor k \rfloor,\lceil l \rceil}\right) + p\left((1-q)\hat{d}_{\lceil k \rceil,\lfloor l \rfloor} + q\hat{d}_{\lceil k \rceil,\lceil l \rceil}\right), \quad (4)$$

where $l = (\phi_n \mod 1)L + 1, k = \tau_n(K-1) + 1, p = k - \lfloor k \rfloor, q = l - \lfloor l \rfloor$, and $\hat{\mathbf{D}} = [\mathbf{D}; \mathbf{d}_1] \in \mathbb{R}^{K \times (L+1)}$. The wavetables $\mathbf{D}$ are fixed in our case, contrary to [16], and we only pick one wavetable at a time, not a weighted sum.

3.2 Frame-Wise LPC Synthesis

Time-varying LPC synthesis is usually done by linearly interpolating the LPC coefficients to the audio resolution and filtering sample by sample. This is not parallelisable and slows down the training process. As an alternative, we approximate LPC synthesis by treating each frame independently and using overlap-add:

$$s_n = \sum_k LPC(g'_n \gamma_n u_{n-kT}; \mathbf{a}_k) w_{n-kT}, \quad (5)$$

where $LPC(e_n; \mathbf{a})$ represents Eqn (2), $\mathbf{a}_k \in \mathbb{R}^M$ are the filter coefficients at the kth frame, u_n and w_n are the windowing functions, $\gamma_n \in \mathbb{R}^+$ is the gain, and T is the hop size. u_n is fixed to the square window. In this way, the computation can be parallelised. We found that the voice quality differences between overlap-add LPC and sample-by-sample LPC are barely noticeable if we use a sufficiently small hop size. We empirically found that a 200 Hz frame rate is sufficient.

3.3 LPC Coefficients Parameterisation

For the LPC filter to be stable, all of its poles must lie inside the unit circle on the complex plane. Stability can be guaranteed using robust representations, such as reflection coefficients [28]. The representation we chose in this work is cascaded 2nd-order IIR filters, and we solve the stability issue by ensuring all the 2nd-order filters are stable. We use the *coefficient representation* from [33] to parameterise the ith IIR filter's coefficients $1 + \eta_{i,1}z^{-1} + \eta_{i,2}z^{-2}$ from the encoder's outputs and cascade them together to form an Mth-order LPC filter:

$$\begin{aligned}&(1 + \eta_{1,1}z^{-1} + \eta_{1,2}z^{-2})(1 + \eta_{2,1}z^{-1} + \eta_{2,2}z^{-2}) \\ &\cdots (1 + \eta_{\frac{M}{2},1}z^{-1} + \eta_{\frac{M}{2},2}z^{-2}) \\ &= 1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_M z^{-M} = A(z).\end{aligned} \quad (6)$$

3.4 Unvoiced Gating

The instantaneous frequency f_n predicted by the encoder is always non-zero and keeps the oscillator working. Without constraint, the model would utilise these harmonics in the unvoiced region creating buzzing artefacts [18]. We propose to mitigate this problem by jointly training the model to predict the voiced/unvoiced probabilities as $v_n \in [0, 1]$ and feeding the gated frequency $\hat{f}_n = v_n f_n$ to the oscillator instead.

4. OPTIMISATION

Training deep learning models is usually accomplished by backpropagating the gradients evaluated at a chosen loss function $\mathcal{L}$ throughout the whole computational graph back to the parameters. Partially inspired by Bhattacharya et al. [29], we derived the closed form of *backpropagation through time* to utilise efficient IIR implementation to solve the problems we mentioned in Section 2.2 while keeping the filter differentiable. Here, $\mathbf{e} \in \mathbb{R}^N$ is the input, $\mathbf{a} \in \mathbb{R}^M$ is the filter coefficients, and $\mathbf{s} \in \mathbb{R}^N$ is the output. Assuming we know $\frac{\partial \mathcal{L}}{\partial \mathbf{s}}$, we can get the derivatives $\frac{\partial \mathcal{L}}{\partial \mathbf{e}}$ and $\frac{\partial \mathcal{L}}{\partial \mathbf{a}}$, using chain rules $\frac{\partial \mathcal{L}}{\partial \mathbf{s}}\frac{\partial \mathbf{s}}{\partial \mathbf{e}}$ and $\frac{\partial \mathcal{L}}{\partial \mathbf{s}}\frac{\partial \mathbf{s}}{\partial \mathbf{a}}$.

4.1 Backpropagation Through the Coefficients

Taking the derivatives of Eqn (2) with respect to a_i we get:

$$\frac{\partial s_n}{\partial a_i} = -s_{n-i} - \sum_{k=1}^{M} a_k \frac{\partial s_{n-k}}{\partial a_i}, \quad (7)$$

which equals $LPC(-s_{n-i}; \mathbf{a})$. $s_n|_{n \le 0}$ does not depend on a_i so the initial conditions $\frac{\partial s_n}{\partial a_i}|_{n \le 0}$ are zeros. We can get $\frac{\partial s_n}{\partial \mathbf{a}}$ with one pass of filtering because $\frac{\partial s_n}{\partial a_j}$ is $\frac{\partial s_n}{\partial a_i}$ shifted by an offset $j - i$. Lastly, we calculate $\frac{\partial \mathcal{L}}{\partial a_i}$ as $\sum_{n=1}^{N} \frac{\partial \mathcal{L}}{\partial s_n}\frac{\partial s_n}{\partial a_i}$.

4.2 Backpropagation Through the Input

To get the derivatives for input e_n, we first re-write Eqn (2) as the following convolutional form:

$$s_n = \sum_{m=1}^{n} e_m h_{n-m}, \quad (8)$$

where $h_n = \mathcal{Z}^{-1}\{H(z)\}, H(z) = \frac{1}{A(z)}$. From Eqn (8) we see that $\frac{\partial s_n}{\partial e_m} = h_{n-m}$. The derivative of loss $\mathcal{L}$ with respect to e_m depends on all future samples s_n, which is:

$$\frac{\partial \mathcal{L}}{\partial e_m} = \sum_{n=m}^{N} \frac{\partial \mathcal{L}}{\partial s_n}\frac{\partial s_n}{\partial e_m} = \sum_{n=m}^{N} \frac{\partial \mathcal{L}}{\partial s_n} h_{n-m}. \quad (9)$$

By swapping the variables n, m and considering the equivalence of Eqn (2) and Eqn (8), Eqn (9) can be simplified to

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial e_n} &= \sum_{m=n}^{N} \frac{\partial \mathcal{L}}{\partial s_m} h_{m-n} \\ &= \frac{\partial \mathcal{L}}{\partial s_n} - \sum_{i=1}^{M} a_i \frac{\partial \mathcal{L}}{\partial e_{n+i}}.\end{aligned} \tag{10}$$

Eqn (10) shows that we can get the derivatives $\frac{\partial \mathcal{L}}{\partial e_n}$ by just filtering $\frac{\partial \mathcal{L}}{\partial s_n}$ with the same filter, but running in backwards. The initial conditions $\frac{\partial \mathcal{L}}{\partial e_n}|_{n>N}$ are naturally zeros.

In conclusion, backpropagation through an IIR filter consists of two passes of the same filter and one matrix multiplication [1]. We implemented the IIR in C++ and CUDA with multi-threading to filter multiple sequences simultaneously [2]. The differentiable IIR is done by registering the above backward computation in PyTorch, and we submitted the implementation to TorchAudio [39] as part of the `torchaudio.functional.lfilter`.

5. EXPERIMENTAL SETUP

5.1 Dataset

We test GOLF as a neural vocoder on the MPop600 dataset [40], a high-quality Mandarin singing voice dataset featuring nearly 600 singing recordings with aligned lyrics sung by four singers. We used the audio recordings from the `f1` (female) and `m1` (male) singers. For each singer, we selected the first three recordings as the test set, the following 27 recordings as the validation set, and used the rest as training data (around three hours in total). All the recordings were downsampled to 24 kHz. The vocoder feature we choose is the log mel-spectrogram. We computed the feature with a window size of 1024 and 80 mel-frequency bins and set the hop size T to 120. We normalised the feature to between zero and one and sliced the training data into two seconds excerpts with 1.5 seconds overlap.

5.2 Model Details

We adopted the encoder from SawSing but replaced the transformer layers with three layers of Bi-LSTM for favourable implementation, resulting in around 0.7M parameters in total. A final linear layer predicts the synthesis parameters $\{f_k, v_k, \gamma_k, \beta_k, \mathbf{a}_k, \mathbf{b}_k\}$. The first four parameters are linearly upsampled to $\{f_n, v_n, \gamma_n, \beta_n\}$. β_n and $\mathbf{b}_k$ are the Gaussian noise's gain and filter coefficients. We added an average pooling layer with a size of 10 and two convolution layers after the encoder to predict the R_d fractional index τ_o at a lower rate and then linearly upsampled to τ_n. This step avoids possible modulation effects caused by switching the wavetables too quickly. A system diagram of GOLF is shown in Fig. 2. We set $K = 100, L = 2048$, Hanning window for w_n, and $M = 22$ for both LPC filters. We used the same hop size T and a window size of 480 for frame-wise LPC. We normalised all wavetables to have equal energy and aligned them with the negative peak.

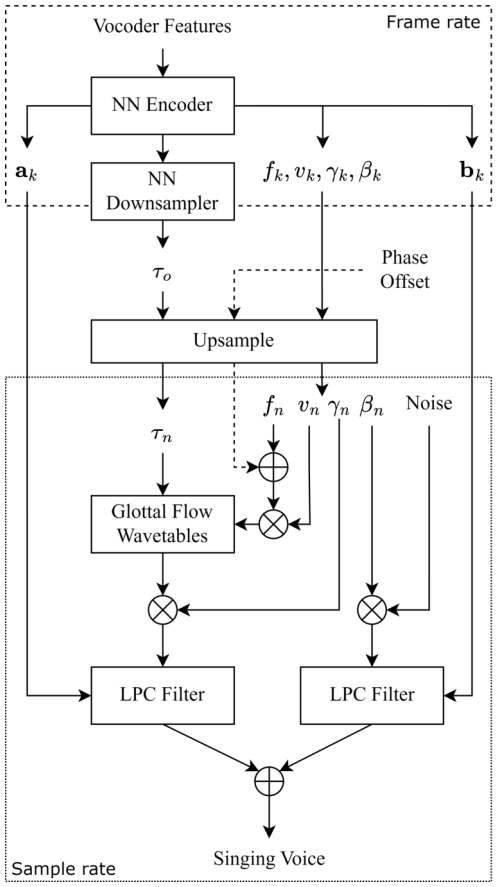


Figure 2. Overview of the GOLF synthesis process. *phase offset* is only introduced at test time, where the details are given in Section 6.1.

We compare GOLF with three DDSP-based baselines using the same NN encoder to predict their synthesis parameters. The first two are the original DDSP [14] and SawSing [18]. We set their noise filter length to 80 and harmonic filter length to 256 for SawSing. The third model is PUlse-train LPC Filter (PULF), which is similar to GOLF but replaces the glottal flow wavetables with a band-limited pulse train [19] using additive synthesis, while the LPC order for the harmonic source is increased to 26 to accommodate the glottal pulse response. The number of oscillating sinusoids was set to over 150 for all the baselines. We did not compare GOLF with Nercessian et al. [19] and Yoshimura et al. [21] because these architectures are based closely on the source-filter model, and use additional postnets to enhance the voice, which makes it harder to compare directly with GOLF.

5.3 Training Configurations

We trained separate models for each singer, resulting in 8 models. The loss function is the summation of the

[1] The computation of the IIR does not need to fulfil the implementation requirements set by the automatic differentiation framework, thus can be highly optimised.

[2] Although the single-core performance of a GPU is usually inferior to a CPU, and we can only use at most one thread for each IIR, the GPU has a much higher number of cores, which is beneficial for training on a large number of sequences at once.

multi-resolution STFT loss (MSSTFT) and f0 loss from SawSing with FFT sizes set to $\{512, 1024, 2048\}$, plus a binary cross entropy loss on voiced/unvoiced prediction. We stopped the gradients from the harmonic source to the f0s and voiced decisions to stabilise the training. We used Adam [41] for running all optimisations. For DDSP and SawSing, the batch size and learning rate were set to 32 and 0.0005; for GOLF and PULF, the numbers were 64 and 0.0001. We used the ground truth f0s (extracted by WORLD [20]) for the harmonic oscillator of PULF during training due to stability issues. We trained all the models for 800k steps to reach sufficient convergence and picked the checkpoint with the lowest validation loss as the final model [3].

6. EVALUATIONS

6.1 Objective Evaluation

The objective metrics we choose are the MSSTFT, the mean absolute error (MAE) in f0, and the Fréchet audio distance (FAD) [42] on the predicted singing of the test set. Table 1 shows that DDSP has the lowest MSSTFT and f0 errors, while SawSing reaches the lowest FAD. GOLF and PULF show comparable results in f0 errors to other baselines. We report the memory usage when training these models and their real-time factor (RTF), both on GPU and CPU, in Table 2. The amount of memory required to train GOLF is around 35% of others, and it runs extremely fast, especially on the CPU.

Singers	Models	MSSTFT	MAE-f0 (cent)	FAD
f1	DDSP	**3.09**	**74.47**±1.19	0.50±0.02
	SawSing	3.12	78.91±1.18	**0.38**±0.02
	GOLF	3.21	77.06±0.88	0.62±0.02
	PULF	3.27	76.90±1.11	0.75±0.04
m1	DDSP	**3.12**	**52.95**±1.03	0.57±0.02
	SawSing	3.13	56.46±1.04	**0.48**±0.02
	GOLF	3.26	54.09±0.30	0.67±0.01
	PULF	3.35	54.60±0.73	1.11±0.04

Table 1. Evaluation results on the test set. We omit the standard deviation if it is smaller than 0.01.

As an additional metric we use the L2 loss between the predicted and the ground truth waveform. The intuition behind this is that GOLF and PULF are the only two models introducing non-linear phase response because of IIR filtering. The filters in DDSP and SawSing are all zero-phase, and the initial phases of the sinusoidal oscillators are fixed to zeros. We emphasise that this test is not targeting human perception but the phase reconstruction ability of the models. Humans cannot perceive the absolute frequency phase, but accurate reconstruction could be important in sound matching and mixing use cases. We evaluate the loss on one of the test samples from `m1` we used in the subjective evaluation. We created a new parameter called *phase offset* sampled at 20 Hz. We linearly upsampled *phase offset* and added it to the instantaneous phase ϕ_n, introducing a slowly varying phase shift. We optimised this parameter by minimising the predicted waveform's L2 loss to the ground truth using Adam with a learning rate of 0.001 and 1000 steps. We wrapped the differences between the points of *phase offset* during optimisation to [-0.5, 0.5]. We ran this optimisation five times for each model. Each time the *phase offset* was initialised randomly. We report the minimum and maximum final losses from these trials. Table 2 shows the lowest losses GOLF and PULF can reach are significantly smaller than the others, with GOLF having the smallest among all.

Models	Memory	RTF		Waveform L2	
		GPU	CPU	Min	Max
DDSP	7.3	0.015	0.237	71.83	88.77
SawSing	7.3	0.015	0.240	75.72	93.16
GOLF	**2.6**	**0.009**	**0.023**	**21.98**	**64.82**
PULF	7.5	0.015	0.248	44.08	70.59

Table 2. The required number of VRAM (GB) for training with a batch size of 32, real-time factor (RTF), and the minimum/maximum L2 loss on waveform using one of the test samples. The benchmark was conducted on an Ubuntu 20.04 LTS machine with an i5-4790k processor and an NVIDIA GeForce RTX 3070 GPU.

6.2 Subjective Evaluation

We conducted an online listening test using Go Listen [43]. We picked one short clip from each test set recording, resulting in 6 clips with duration ranging from 6 to 11 seconds. The test is thus divided into six examples, each consisting of one ground truth clip and four synthesised results from different models, and their loudness was normalised to –16dB LUFS. The order of the examples and the stimulus were randomised for each subject. Each subject was requested to rate the quality of these stimuli on a score from 0 to 100. We collected responses from 33 anonymous participants. We dropped one of the participants who did not indicate using headphones. We normalise scores to fall between 1 to 5 and report the Mean Opinion Score (MOS) in Fig. 4. DDSP has the highest opinion scores overall, and a Wilcoxon signed-rank test shows that it is not statistically significantly different from the `f1` ground truth (p = 0.168). We applied the one-side Wilcoxon test on GOLF and PULF to compare them with SawSing, and the results show that GOLF significantly outperforms SawSing ($p <$ 0.0001), and PULF performs better than SawSing on `m1` ($p < 0.022$).

7. DISCUSSIONS

Given the evaluation results and the number of synthesis parameters in GOLF is roughly six times smaller than DDSP and SawSing, it is clear that GOLF's synthesis parameters are a more compact representation. Comparing

[3] The trained checkpoints, source codes, and audio samples are available at `https://github.com/yoyololicon/golf`.

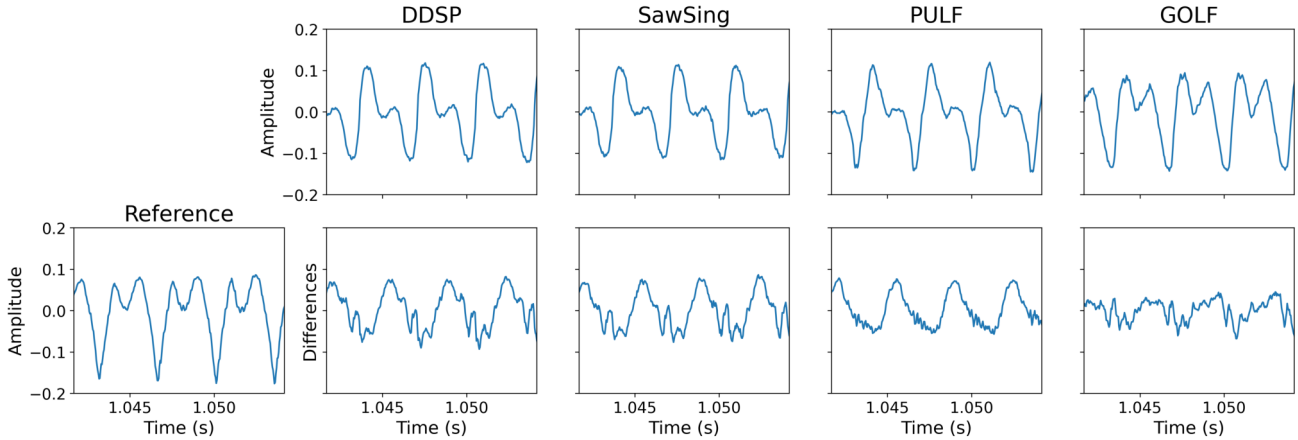


Figure 3. The predicted waveforms of a short segment from one of the `m1` test samples. The differences were computed by subtracting the predicted signal from the reference.

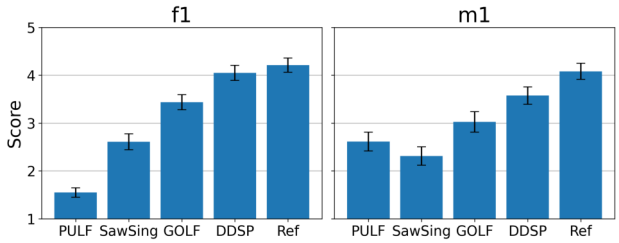


Figure 4. The MOS results of the vocoders trained on different singers with 95% confidence interval.

the differences between GOLF and PULF in Table 2, we can see that the performance gain is due to the use of wavetables. Other baselines synthesise band-limited harmonic sources with many sinusoids oscillating simultaneously, thus increasing the computational cost. PULF's MOS score is much worse on `f1`, with noticeable artefacts in the unvoiced region and random components of the voice. After investigation, we found the noise gains β_n predicted by PULF fluctuating at high speed, producing a modulation effect. This behaviour is also found in PULF trained on `m1` and GOLF, even on the harmonic gains γ_n. Still, the amount of fluctuation is small and barely noticeable in the test samples. Given the available results, we could only conclude that this effect relates to the type of harmonic source and the range of f0, i.e., female singers have higher f0. This amplitude modulation effect cannot be observed in spectrograms and thus is not captured by the training loss we used. It could be an intrinsic drawback of using frame-wise LPC approximation, but more experiments and comparisons with sample-wise LPC are needed. In addition, SawSing produced low scores for both singers because of the buzzing artefacts in the unvoiced region. Although unvoiced gating (Sec. 3.4) reduces this problem to a large degree, human ears are susceptible to this effect. This could be an inherent problem in using a sawtooth as the harmonic source.

The L2 loss shown in Table 2 demonstrates that GOLF matches phase-related characteristics more accurately than other models. Fig. 3 shows GOLF produces the most similar waveform to the ground truth. Other baselines' waveforms are similar because they use the same additive synthesiser. It is possible to reduce their L2 loss by optimising the initial phases of the oscillators, but this cannot account for time-varying source shapes. Low L2 loss is a positive effect of the deterministic phase responses embedded in GOLF. This opens up many possibilities, such as decomposing and analysing the voice in a differentiable manner and training the vocoder using the time domain loss function. The latter could be a possible way to reduce the fluctuation problem discussed in the previous paragraph. The waveform matching of GOLF can be improved further by using a more flexible glottal source model, adding FIR and all-pass filters to account for the voice's mixed-phase components and the recording environment's acoustic response.

Lastly, we note that cascaded IIR filters provide an orderless representation (i.e. the cascading order does not affect the outputs). This results in the *responsibility problem* [44, 45] for the last layer of the encoder, which might be one of the reasons why GOLF and PULF are less stable to train than other baselines. Developing architectures that can handle orderless representation or switch to other robust representations are possible ways to address this.

8. CONCLUSIONS

We present a lightweight singing voice vocoder called GOLF, which uses wavetables with different glottal flows as entries to model the time-varying harmonic components and differentiable LPC filters for filtering both the harmonics and random elements. We show that GOLF requires less memory to train and runs an order of magnitude faster on the CPU than other DDSP-based vocoders, but still attains competitive voice quality in subjective and objective evaluations. Furthermore, we empirically show that the predicted waveforms from GOLF represent the voice's phase response more faithfully, which could allow us to use GOLF to decompose and analyse human voice.

9. ACKNOWLEDGEMENTS

The authors want to thank Moto Hira (`mthrok`), Christian Puhrsch (`cpuhrsch`), and Alban Desmaison (`albanD`) for reviewing our pull requests to TorchAudio. We are incredibly grateful to Parmeet Singh Bhatia (`parmeet`) for implementing the first version of efficient IIR in the TorchAudio codebase as `lfilter.cpp`. We thank Ben Hayes for giving feedback on the equations of backpropagation through an IIR. The first author wants to exclusively thank Ikuyo Kita for giving positive, energetic support during the writing process. The first author 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 Queen Mary University of London.

10. REFERENCES

[1] M. W. Macon, L. Jensen-Link, J. Oliverio, M. A. Clements, and E. B. George, "A singing voice synthesis system based on sinusoidal modeling," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 1. IEEE, 1997, pp. 435–438.

[2] J. Bonada, Ò. Celma Herrada, À. Loscos, J. Ortolà, X. Serra, Y. Yoshioka, H. Kayama, Y. Hisaminato, and H. Kenmochi, "Singing voice synthesis combining excitation plus resonance and sinusoidal plus residual models," in *International Computer Music Conference (ICMC)*, Havana, Cuba, 2001.

[3] J. Bonada and X. Serra, "Synthesis of the singing voice by performance sampling and spectral models," *IEEE Signal Processing Magazine*, vol. 24, no. 2, pp. 67–79, 2007.

[4] J. Bonada, M. Umbert Morist, and M. Blaauw, "Expressive singing synthesis based on unit selection for the singing synthesis challenge 2016," in *INTERSPEECH*. International Speech Communication Association (ISCA), 2016.

[5] K. Saino, H. Zen, Y. Nankaku, A. Lee, and K. Tokuda, "An HMM-based singing voice synthesis system," in *9th International Conference on Spoken Language Processing*, 2006.

[6] Y. Hono, S. Murata, K. Nakamura, K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda, "Recent development of the DNN-based singing voice synthesis system—Sinsy," in *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2018, pp. 1003–1009.

[7] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan *et al.*, "Natural TTS synthesis by conditioning WaveNet on mel spectrogram predictions," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4779–4783.

[8] J.-M. Valin and J. Skoglund, "LPCNet: Improving neural speech synthesis through linear prediction," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 5891–5895.

[9] R. Yoneyama, Y.-C. Wu, and T. Toda, "Unified source-filter GAN with harmonic-plus-noise source excitation generation," in *INTERSPEECH*. International Speech Communication Association (ISCA), 2022.

[10] R. Prenger, R. Valle, and B. Catanzaro, "WaveGlow: A flow-based generative network for speech synthesis," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 3617–3621.

[11] J. Liu, C. Li, Y. Ren, F. Chen, and Z. Zhao, "DiffSinger: Singing voice synthesis via shallow diffusion mechanism," in *AAAI Conference on Artificial Intelligence*, vol. 36, no. 10, 2022, pp. 11 020–11 028.

[12] Y.-P. Cho, F.-R. Yang, Y.-C. Chang, C.-T. Cheng, X.-H. Wang, and Y.-W. Liu, "A survey on recent deep learning-driven singing voice synthesis systems," in *2021 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)*. IEEE, 2021, pp. 319–323.

[13] N. Takahashi, M. Kumar, Singh, and Y. Mitsufuji, "Hierarchical diffusion models for singing voice neural vocoder," *arXiv preprint arXiv:2210.07508*, 2022.

[14] J. Engel, L. H. Hantrakul, C. Gu, and A. Roberts, "DDSP: Differentiable digital signal processing," in *International Conference on Learning Representations*, 2020.

[15] B. Hayes, C. Saitis, and G. Fazekas, "Neural waveshaping synthesis," in *Proc. International Society for Music Information Retrieval*, 2021.

[16] S. Shan, L. Hantrakul, J. Chen, M. Avent, and D. Trevelyan, "Differentiable wavetable synthesis," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 4598–4602.

[17] J. Alonso and C. Erkut, "Latent space explorations of singing voice synthesis using DDSP," *arXiv preprint arXiv:2103.07197*, 2021.

[18] 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 *Proc. International Society for Music Information Retrieval*, 2022.

[19] S. Nercessian, "Differentiable WORLD synthesizer-based neural vocoder with application to end-to-end audio style transfer," *arXiv preprint arXiv:2208.07282*, 2022.

[20] M. Morise, F. Yokomori, and K. Ozawa, "WORLD: a vocoder-based high-quality speech synthesis system for real-time applications," *IEICE TRANSACTIONS on Information and Systems*, vol. 99, no. 7, pp. 1877–1884, 2016.

[21] T. Yoshimura, S. Takaki, K. Nakamura, K. Oura, Y. Hono, K. Hashimoto, Y. Nankaku, and K. Tokuda, "Embedding a differentiable mel-cepstral synthesis filter to a neural speech synthesis system," *arXiv preprint arXiv:2211.11222*, 2022.

[22] S. Li, Y. Zhao, R. Varma, O. Salpekar, P. Noordhuis, T. Li, A. Paszke, J. Smith, B. Vaughan, P. Damania *et al.*, "PyTorch distributed: Experiences on accelerating data parallel training," *Proceedings of the VLDB Endowment*, vol. 13, no. 12, 2020.

[23] G. Degottex, "Glottal source and vocal-tract separation," Ph.D. dissertation, Université Pierre et Marie Curie-Paris VI, 2010.

[24] H.-L. Lu and J. O. Smith III, "Glottal source modeling for singing voice synthesis." in *International Computer Music Conference (ICMC)*, 2000.

[25] G. Fant, J. Liljencrants, Q.-g. Lin *et al.*, "A four-parameter model of glottal flow," *STL-QPSR*, vol. 4, no. 1985, pp. 1–13, 1985.

[26] J. D. Markel and A. H. Gray, *Linear Prediction of Speech*, ser. Communication and Cybernetics. Berlin, Heidelberg: Springer, 1976, vol. 12.

[27] S. Oh, H. Lim, K. Byun, M.-J. Hwang, E. Song, and H.-G. Kang, "ExcitGlow: Improving a WaveGlow-based neural vocoder with linear prediction analysis," in *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2020, pp. 831–836.

[28] K. Subramani, J.-M. Valin, U. Isik, P. Smaragdis, and A. Krishnaswamy, "End-to-end LPCNet: A neural vocoder with fully-differentiable LPC estimation," *arXiv preprint arXiv:2202.11301*, 2022.

[29] P. Bhattacharya, P. Nowak, and U. Zölzer, "Optimization of cascaded parametric peak and shelving filters with backpropagation algorithm," in *International Conference on Digital Audio Effects*, 2020, pp. 101–108.

[30] S. Nercessian, "Neural parametric equalizer matching using differentiable biquads," in *International Conference on Digital Audio Effects*, 2020, pp. 265–272.

[31] B. Kuznetsov, J. D. Parker, and F. Esqueda, "Differentiable IIR filters for machine learning applications," in *International Conference on Digital Audio Effects*, 2020, pp. 297–303.

[32] J. T. Colonel, C. J. Steinmetz, M. Michelen, and J. D. Reiss, "Direct design of biquad filter cascades with deep learning by sampling random polynomials," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 3104–3108.

[33] S. Nercessian, A. Sarroff, and K. J. Werner, "Lightweight and interpretable neural modeling of an audio distortion effect using hyperconditioned differentiable biquads," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 890–894.

[34] T. Kim, Y.-H. Yang, A. Sincia, and J. Nam, "Joint estimation of fader and equalizer gains of dj mixers using convex optimization," in *International Conference on Digital Audio Effects*. DAFx, 2022, pp. 312–319.

[35] 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, 2022.

[36] X. Serra and J. Smith, "Spectral modeling synthesis: A sound analysis/synthesis system based on a deterministic plus stochastic decomposition," *Computer Music Journal*, vol. 14, no. 4, pp. 12–24, 1990.

[37] G. Fant, "The LF-model revisited. transformations and frequency domain analysis," *Speech Trans. Lab. Q. Rep., Royal Inst. of Tech. Stockholm*, vol. 2, no. 3, p. 40, 1995.

[38] C. Gobl, "Reshaping the transformed LF model: Generating the glottal source from the waveshape parameter Rd," in *INTERSPEECH*. International Speech Communication Association (ISCA), Aug. 2017, pp. 3008–3012.

[39] Y.-Y. Yang, M. Hira, Z. Ni, A. Astafurov, C. Chen, C. Puhrsch, D. Pollack, D. Genzel, D. Greenberg, E. Z. Yang, J. Lian, J. Hwang, J. Chen, P. Goldsborough, S. Narenthiran, S. Watanabe, S. Chintala, and V. Quenneville-Bélair, "Torchaudio: Building blocks for audio and speech processing," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 6982–6986.

[40] C.-C. Chu, F.-R. Yang, Y.-J. Lee, Y.-W. Liu, and S.-H. Wu, "MPop600: A mandarin popular song database with aligned audio, lyrics, and musical scores for singing voice synthesis," in *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2020, pp. 1647–1652.

[41] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[42] 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.

[43] 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.

[44] Y. Zhang, J. Hare, and A. Prugel-Bennett, "Deep set prediction networks," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[45] B. Hayes, C. Saitis, and G. Fazekas, "The responsibility problem in neural networks with unordered targets," 2023. [Online]. Available: https://openreview.net/forum?id=jd7Hy1jRiv4

HARMONIC ANALYSIS WITH NEURAL SEMI-CRF

Qiaoyu Yang **Frank Cwitkowitz** **Zhiyao Duan**
University of Rochester
{qyang15, fcwitkow}@ur.rochester.edu, zhiyao.duan@rochester.edu

ABSTRACT

Automatic harmonic analysis of symbolic music is an important and useful task for both composers and listeners. The task consists of two components: recognizing harmony labels and finding their time boundaries. Most of the previous attempts focused on the first component, while time boundaries were rarely modeled explicitly. Lack of boundary modeling in the objective function could lead to segmentation errors. In this paper, we introduce a novel approach named *Harana*, to jointly detect the labels and boundaries of harmonic regions using neural semi-CRF (conditional random field). In contrast to rule-based scores used in traditional semi-CRF, a neural score function is proposed to incorporate features with more representational power. To improve the robustness of the model to imperfect harmony profiles, we design an additional score component to penalize the match between the candidate harmony label and the absent notes in the music. Quantitative results from our experiments demonstrate that the proposed approach improves segmentation quality as well as frame-level accuracy compared to previous methods. The source code used in this paper is available on GitHub [1] .

1. INTRODUCTION

In music, harmony is the sound resulted from two or more pitches being performed together. It is the vertical aspect of music [1], and is essential for both music creation and perception. During music analysis, a harmony label is often assigned to a music segment that is harmonically coherent. Many composers use harmonic progressions to set up a musical template in which texture could then be filled [2]. For listeners, harmonic structure is a crucial mid-level representation of music that can influence the perception of other music elements such as melody and rhythm [3].

The task of harmonic analysis aims to find the correct segmentation of a music piece and to identify the corresponding label for each segmented region. These two goals are closely related. Regions with strong confidence of a candidate harmony label tend to possess the boundaries of a true segmentation [4]. On the other hand, the oracle segmentation could help the prediction of the true underlying harmony for the notes in each region [4]. Therefore, to achieve successful analysis of harmony, both of the two goals as well as their relationship should be considered.

Targeting the two indispensable components of harmonic analysis simultaneously, we propose an approach to jointly predict the boundaries and labels of harmonic regions using neural semi-Markov conditional random field (semi-CRF). It is well-known that the harmonic regions in music do not always share the same length [5]. Compared to conventional sequence labeling models, semi-CRF is more suitable for the task because it allows for various lengths among the labeled regions [6].

In the original setting of semi-CRF, a score is computed in each segmented region using the weighted sum of rule-based features [6]. However, rule-based features are bounded by pre-defined rules and might not exploit the interaction between notes and other intermediate music representations deeply enough. To solve this problem, we design a neural scoring function that first estimates the frame-level harmony distributions using a neural network and then adapts them to candidate harmony labels with an attention mechanism. The attention mechanism could make the scoring module more efficient by concentrating on sub-regions that are more harmonically related to the candidate label. In addition, an absence score is added to the scoring function to improve the robustness of the model to imperfect harmony profiles of the music. Through experiments we find that the proposed architectural components collectively yield improvement on both segmentation quality and harmony labels accuracy. We focus on MIDI-like symbolic music input in our experiments but the method could be easily adapted to audio.

In summary, our contributions include:

- Proposing the first neural semi-CRF model to jointly estimate harmony labels and their time boundaries;
- Proposing an attention-based score function to alleviate the influence of extra non-chordal notes and missing chordal notes; and
- Proposing a novel absence score to improve the robustness to imperfect harmony profiles.

2. RELATED WORKS

Due to the importance of harmony in music, a substantial amount of automatic systems have been designed for har-

[1] https://github.com/QiaoyuYang/harana

© Q. Yang, F. Cwitkowitz, and Z. Duan. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).
Attribution: Q. Yang, F. Cwitkowitz, and Z. Duan, "Harmonic Analysis with Neural Semi-CRF", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

monic analysis. Early systems tended to focus on using music audio as input and apply domain knowledge from music theory. To encode the audio waveform, a time-frequency representation, or spectrogram, is usually extracted using the short-time Fourier Transform. Then, with the observation that it is the pitch class of notes rather than the absolute pitch height that affects the harmonic content, a common practice is to reduce the spectrogram to a chromagram with 12 bins corresponding to the 12 pitch classes. In the decoding stage, the chromagram can be matched to predefined chord profiles [7, 8] or made to emit explicit labels using probabilistic models such as hidden Markov model (HMM) [9–11] or CRF [11].

With the increasing popularity of deep learning in the past decade, end-to-end models based on deep neural networks have received extensive attention [12–16]. To model the temporal evolution of music context, Boulanger-Lewandowsk et al. extracted audio features using a recurrent neural network (RNN) [17]. To better aggregate context information and learn intermediate representations with a temporal hierarchy, Zhou and Lerch used a convolutional neural network (CNN) with low-pass filters [12]. McFee and Bello further combined CNN and RNN in the feature encoder for chord recognition [13]. As a powerful attention-based architecture designed for long-term sequence modeling, transformers have also been incorporated in some recent approaches to harmonic analysis [14, 15].

While the harmonic progression or context information can be modeled with various techniques, the majority of existing methods do not directly optimize for region-level output. Some methods adopt a two-stage approach, where the first stage outputs frame-level chord labels and the second stage smooths frame-level labels with post-processing [9–11, 18–20]. However, different from other simple sequence labeling tasks such as part-of-speech tagging, a harmonic label could correspond to a region spanning multiple frames. Although temporal smoothing by HMM or CRF regresses some sporadically outliers back to the harmonic streams, these models could still suffer from segmentation errors. Masada and Bunescu relaxed the constraint on fixed-size time-span of the output prediction [21]. They used a generalized variant of CRF, semi-CRF, to jointly detect chord labels and their boundaries. However, the features to the semi-CRF are entirely rule-based, which means they are not necessarily optimal for the end task. In this work, we build on the semi-CRF framework and explore neural features and scoring techniques that are jointly optimized for the end task - harmony labeling and boundary prediction.

3. METHODS

In our proposed model, Harana, we first estimate the harmony (including root, quality, and pitch activation in this work) at the frame-level; then we aggregate the frame-level estimation into region-level segment scores based on candidate segments; finally, we use semi-CRF to find the best segmentation candidate and its corresponding labels. We focus on symbolic music input in our experiments. The following subsections describe the model in detail.

3.1 Data Representation

3.1.1 Symbolic Music Input

Given a symbolic music piece, we slice it into short frames of one eighth of a beat long. We use beat instead of note duration in order to represent the basic time unit because music with different meters may have different distributions on the note length. The pitch information in each frame is summarized with a 12-d pitch class distribution vector, which describes the normalized distribution of the duration of each pitch class in the frame. To help distinguish between harmonies with the same pitch class vector, we also include the bass note (the lowest note) in the input to the model; it is represented as a 12-d one-hot vector indicating the bass pitch class in each frame. Combining the pitch class distribution and the bass note, the input to the model is a sequence of 24-d vectors.

3.1.2 Harmony

A popular representation of music harmony in symbolic music is the Roman numeral encoding, where the full harmonic context of a label, including tonic and degree, is considered [22]. However, the combination of all the components produces 47k different harmony labels, which are intractable for a classification model with limited training data. A possible solution is to classify each harmony component independently, but this is incompatible with semi-CRF because the boundary of each component must be the same. As a compromise, we use a subset of the harmony components, root and quality, and model them jointly.

The root is represented as a 12-d one-hot vector corresponding to the 12 pitch classes. The quality is represented as a 10-d one-hot vector corresponding to 10 commonly used classes. In addition to root and quality, we use another harmony representation, the pitch class activation vector, in the neural score function. Previous works have shown its effectiveness as a label encoding for harmonic analysis [13]. These vectors are 12-d multi-hot and are circularly shifted from the pitch-class activation vectors rooted at C.

3.2 Semi-CRF

Semi-CRF is a probabilistic model for sequence labeling with a variable label-span. Given a sequence of input frames $X = \langle X_1, X_2, ..., X_N \rangle$, semi-CRF provides the conditional probability of the sequence of contiguous non-overlapping labeled segments $Y = \langle Y_1, Y_2, ..., Y_K \rangle$, where N is the number of frames and K is the number of segments. Since the labeled segments could span multiple frames, they are represented as three-dimensional tuples $Y_i = (u_i, v_i, l_i)$, where u_i, v_i and l_i respectively denote the onset, offset and label of the segment. In the context of harmonic analysis, X represents the input music frames and Y represents the harmonic regions.

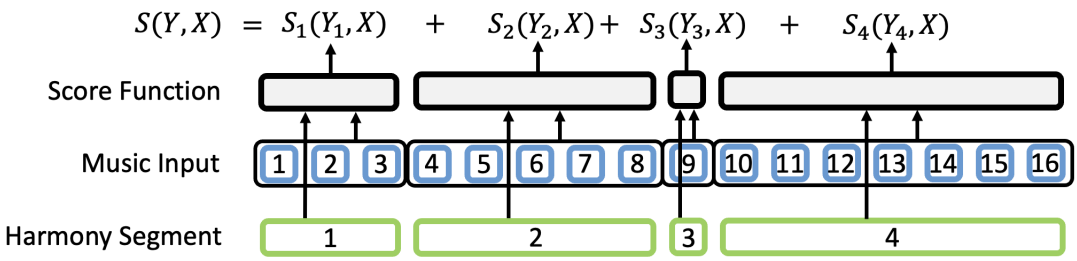


Figure 1: The semi-CRF architecture in the context of harmonic analysis. The total score is computed from music input and a set of candidate harmony segments. Numbers in the blue squares are the frame indices. Numbers in the green rectangles are the indices of candidate harmony segments.

The conditional probability given by semi-CRF takes the form of

$$P(Y|X) = \frac{e^{WF(Y,X)}}{Z(X)}, \tag{1}$$

where F is a feature vector computed from X and Y, W is a learnable weight matrix, and $Z = \sum_Y e^{WF(Y,X)}$ is a normalization factor summarizing all possible segmentation and labeling of the input sequence. In this work, we propose to generalize the weighted feature score to a neural score function $S(Y, X)$ so that

$$P(Y|X) = \frac{e^{S(Y,X)}}{Z(X)}. \tag{2}$$

With the assumption that the harmony labels are Markovian given the music input, the score function could be decomposed into the sum of segment-level scores that are dependent only on the current and the previous segments.

$$S(Y, X) = \sum_{i=1}^{K} S_i(Y_i, X; Y_{i-1}). \tag{3}$$

To simplify the notation, we treat Y_{i-1} as a parameter for the i-th segment's score function and omit it in the following sections. Figure 1 demonstrates the structure of semi-CRF in the context of music harmonic analysis.

3.3 Frame-Level Estimation

Followng Micci et al. [23], the frame-level estimation of harmony information is achieved with a DenseNet-GRU architecture. The DenseNet-GRU module is followed by fully connected layers and finally the vectors corresponding to different types of harmony information are estimated using separate linear heads. The softmax function is used to produce the class distributions of the root and the quality, whereas sigmoid is used to find the activation of each pitch class. Mathematically, the computation of frame-level harmony estimation can be formulated as

$$\begin{aligned} E(n) &= MLP(GRU(DenseNet(X_n))), \\ \hat{D}_R(n) &= \text{Softmax}(FC_R(E(n))), \\ \hat{D}_Q(n) &= \text{Softmax}(FC_Q(E(n))), \\ \hat{PC}(n) &= \text{Sigmoid}(FC_{PC}(E(n))), \end{aligned} \tag{4}$$

where X_n is the nth frame of the input music. $\hat{D}_R(n)$, $\hat{D}_Q(n)$ and $\hat{PC}(n)$ represent the root distribution, quality distribution and the pitch class activations of the estimated harmony for a frame.

3.4 Attention-Based Score Function

As described in Eq. (3), the CRF model evaluates possible sequences of harmony labels and their segmentation. For each segment, i.e., a candidate harmony region, we need to aggregate the frame-level harmony information (root, quality and pitch activation) computed from Eq. (4). A simple method would be taking the average or the mode, but we note that a harmonic region is not likely to contain homogeneous harmonic content. In order to dynamically weigh the harmonic importance of each frame within a region, an attention module is proposed to focus on the frames that are most similar to the candidate harmony label. In particular, the scaled dot-product attention [24] is used:

$$A(Q, K, V) = \frac{\sum_{i=1}^{N} Q^T K_i V_i}{\sqrt{d}}, \tag{5}$$

where Q is the query vector, K is the key sequence, V is the value sequence and d is the vector size.

In the context of our model, the estimated frame-level harmony sequence of a candidate harmony region serves as both the key and value while the candidate region-level harmony itself is the query. Then, the candidate-informed (CI) estimation can be computed as

$$\hat{H}_{CI}(Y_i) = A(H(l_i), \hat{H}(u_i : v_i), \hat{H}(u_i : v_i)), \tag{6}$$

where l_i is the i-th candidate harmony label, and $H(l_i)$ is its harmony representation, which can be root D_R, quality D_Q or pitch class activation PC as defined in Eq. (4). Variables u_i and v_i are the first and last frames of the ith harmonic region Y_i, and $\hat{H}(u_i : v_i)$ is the vector sequence of estimated frame-level harmony representations from the music input.

Now that we have a single embedding vector to summarize the harmonic content in the i-th candidate region, the score of assigning the candidate harmony label l_i to this region can be described by the similarity between the candidate harmony label embedding $H(l_i)$ and the candidate-informed music embedding $\hat{H}_{CI}(Y_i)$. Dot product is used

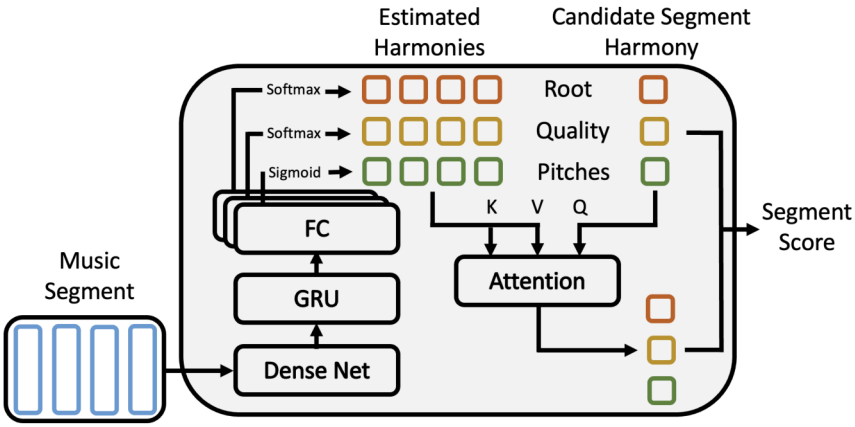


Figure 2: The proposed pipeline of the neural encoder and scoring function.

to calculate the similarity:

$$S_i^H(Y_i, X) = H(l_i)^T \hat{H}_{CI}(Y_i). \tag{7}$$

To further model the transition probability between adjacent harmony labels and enforce more inductive bias in decoding, a transition score between segments is computed:

$$S_i^T(Y_i) = T[l_{i-1}, l_i] + (v_i - u_i)T[l_i, l_i], \tag{8}$$

where T is the transition matrix containing log-probabilities of harmony transitions at the frame level. It is pre-computed from the ground-truth labels in the training data.

Combining the similarity score and the transition score, the score function of a candidate harmony region is

$$S_i(Y_i, X) = \sum_H S_i^H(Y_i, X) + \lambda S_i^T(Y_i), \tag{9}$$

where λ is a hyperparameter to balance the two score components. Figure 2 illustrates the overall structure of the neural front end and the scoring function.

3.5 Absence Score

In Eq. (7), the comparison between the candidate-informed music embedding $\hat{H}_{CI}$ with the candidate harmony representation H indicates the likelihood of the candidate harmony. However, this comparison may not be robust when there are many non-chordal notes or missing chordal notes in the estimation. In this case, the estimated class distributions $\hat{D}_R$ and $\hat{D}_Q$ in Eq. (4) would be relatively flat and the pitch class activation vector $\hat{PC}$ would not align well with a chord template. In other words, the neural front-end may not sufficiently suppress non-chordal notes and recognize missing chordal notes to produce class distributions discriminative enough for the semi-CRF to decode the harmony. To improve the robustness of the model to such issues, we introduce an *absence score* to allow the model to filter out pitch activations that are not active within the input music, the majority of which represent non-chordal notes that should not intersect with chordal notes of the underlying harmony. To compute the absence score, the complement of the input pitch class vector is sent to the neural front-end. That means the input to Eq. (4) is transformed by

$$X_n[1:12] = 1 - X_n[1:12]. \tag{10}$$

The harmony information estimated from the inactive music $\hat{H}^{inact}$ are then compared with the candidate harmony vectors $H(l_i)$. The similarity between them should be minimized. In summary, the absence score of a candidate harmony region is

$$AS_i^H(Y_i, X) = -H(l_i)^T \hat{H}_{CI}^{inact}(Y_i). \tag{11}$$

When the absense score is used, the complete score function becomes

$$S_i(Y_i, X) = \sum_H S_i^H(Y_i, X) + AS_i^H(Y_i, X) + \lambda S_i^T(Y_i), \tag{12}$$

3.6 Optimization

For training, both the input music frames and the ground-truth harmony label segments are provided. The goal is to update the model parameters such that the probability computed in Eq. (2) is maximized. This is equivalent to minimizing the negative log likelihood (NLL) loss:

$$\begin{aligned} NLL(\theta) &= -\log P_\theta(Y|X) \\ &= log(Z_\theta(X)) - S_\theta(Y, X), \end{aligned} \tag{13}$$

where θ are the model parameters. We then compute the gradient of the loss with respect to the parameters to train our model using gradient descent.

During inference, where only the input music frames are provided, the goal becomes finding the correct segmentation and the corresponding labels that maximize the probability $P(Y|X)$. Since the normalization factor as a sum of exponential scores stays positive, maximizing the score function $S(Y, X)$ suffices to decode the segments and labels.

In both training and inference, we used the algorithms based on dynamic programming proposed in the original semi-CRF paper to expedite the optimization process [6].

4. EXPERIMENTS

4.1 Data

A collection of datasets from various sources [22, 25–27] organized by Micchi et al. [28] is used to train and evaluate the proposed architecture. Table 1 summarizes the statistics of the data included in our experiments. MusPy [29] is used to read the compressed MusicXML files and a parser adapted from [28] is employed to handle the proposed data representations. To increase the size of the dataset and help alleviate possible data imbalance, each piece is transposed to 12 different keys. The dataset is split into disjoint subsets for training and testing with a 2:1 split.

4.2 Implementation Details

Following the original paper for faster training [30], DenseNet is implemented in three separate blocks. 1-D convolution along the time frame dimension is used in each convolutional layer. Guided by the observation that harmony changes usually occur on average at a lower frequency than the frame rate, pooling layers are added between blocks to reduce the temporal resolution of the harmony output.

To ensure continuity and completeness of harmony regions in the training samples, we force the sample boundaries to be aligned with measure boundaries. A sample is chosen as 96 frames because it is divisible by all the common measure lengths existed in the dataset. Additionally, to avoid over-sampling from music pieces with longer length, the piece index is sampled uniformly first before a music sample is selected from the piece.

The entire pipeline is implemented using PyTorch. Adam optimizer is applied with learning rate of 10^{-4} and weight decay of 10^{-2}. Dropout with rate 0.2 is added between GRU layers and after each hidden fully-connected layer to avoid over-fitting. The λ in Eq. (12) is chosen empirically to be 0.001.

4.3 Evaluation Metrics

The task of music harmony analysis is two fold: recognizing the correct labels and finding the correct segmentation corresponding to the labels. To obtain a full picture of the model performance, we used two types of evaluation metrics to assess both aspects of the task.

First, the frame-level accuracy is computed for both root and quality. The accuracy on a reduced dictionary of quality including only major and minor is also reported due to its prevalence in the literature and adequacy in many practical uses. During training, the accuracy is computed at the sample level. During inference, the result is averaged across all frames in a song.

	Pieces	Crotchet	Chord Annotations
BPSFH	32	23554	8615
Roman Text	82	18208	7935
Tavern	27	20673	10723
Lopez	180	31367	16666

Table 1: Summary of statistics of the datasets.

The other evaluation metric focuses on the segmentation quality of the output. We use the standard segmentation scores from the mir_eval package [31, 32]. The scores are based on directional Hamming distance and consider the overlap between the estimated harmony intervals and the ground-truth intervals. The directional Hamming distance between the set of estimated intervals $\hat{\mathcal{I}} = \{\hat{I}_i\} = \{[\hat{u}_i, \hat{v}_i]\}$ and the set of ground-truth intervals $\mathcal{I} = \{I_i\}$ is computed as the following:

$$DHD(\hat{\mathcal{I}}, \mathcal{I}) = \frac{\sum_{\hat{I}_i \in \hat{\mathcal{I}}} (|\hat{I}_i| - \max_{I_j \in \mathcal{I}} |\hat{I}_i \cap I_j|))}{\sum_{\hat{I}_i \in \hat{\mathcal{I}}} |\hat{I}_i|}. \tag{14}$$

When a harmony boundary is missing from the estimation, an estimated harmony interval overlaps with multiple ground-truth intervals, but the maximum overlap is bounded by the length of the ground-truth intervals, leaving a large portion of the estimated interval not subtracted hence a large distance value. Therefore, a large $DHD(\hat{\mathcal{I}}, \mathcal{I})$ often indicates under-segmentation, while a large $DHD(\mathcal{I}, \hat{\mathcal{I}})$ often indicates over-segmentation. To summarize the two directional distances in a single metric, the overall segmentation quality score is computed as

$$SQ = 1 - \max(DHD(\mathcal{I}, \hat{\mathcal{I}}), DHD(\hat{\mathcal{I}}, \mathcal{I})). \tag{15}$$

4.4 Baseline Models

Three baseline models are included in our experiments to demonstrate the performance improvement of our proposed method. The chosen baselines are all relevant to our model by sharing parts of the architecture. Since the neural front-end of Harana is CRNN, we first test if a plain CRNN model [23] could achieve comparable results. A second baseline model, frog [28], also relies on CRNN to extract music features. In contrast to our model, it uses a neural autoregressive distribution estimator (NADE) to decode the harmony label. At the decoding stage, it defines an order of the harmony components and iteratively predict the next component conditioned on the current component. The same output harmony categories of root and quality output are considered in the NADE decoder. A third baseline worth comparing to is the rule-based semi-CRF proposed by Masada and Bunescu [21]. It uses handcrafted rules as features to compute the segment scores in semi-CRF. For simplicity, we implemented the two most important features, chord coverage and segment purity, in our experiment. Chord coverage measures what percentage of chordal notes are covered by the music segment while segment purity describes what proportion of notes in the music segment are indeed chordal notes.

Model	Root Acc	Quality Acc	Overall Acc	Under Seg	Over Seg	Overall Seg
Harana	**0.744**	0.743	**0.651**	**0.722**	0.747	0.649
Harana - no semi-CRF	0.732	0.715	0.634	0.678	0.740	0.639
Harana - no Attention Fusing	0.741	0.738	0.650	0.716	**0.749**	0.645
Harana - no Absence Score	0.743	**0.746**	0.643	0.719	0.748	**0.650**

Table 2: The result of ablation studies summarizing the effect of removing each proposed component of the model on both frame-level accuracy and segmentation quality.

Model	Root	Quality	Majmin	Overall
CRNN	0.735	0.714	0.865	0.634
frog	0.733	0.542	0.815	0.459
RuleSCRF	0.684	0.645	0.847	0.600
Harana	**0.744**	**0.743**	**0.886**	**0.651**

Table 3: The frame-level accuracy for different models.

Model	Under Seg	Over Seg	Overall
CRNN	0.681	0.738	0.639
frog	0.681	0.724	0.624
RuleSCRF	0.666	0.741	0.625
Harana	**0.722**	**0.747**	**0.649**

Table 4: The segmentation quality for different models.

5. RESULTS

5.1 Frame-Level Accuracy

Table 3 shows the result on frame-level accuracy. It can be seen that Harana outperforms the baseline models on all the measures. The large gap between Harana and the rule-based semi-CRF model demonstrates the value of a neural score function. Without a neural front-end, the rule-based model even has weaker performance than the plain CRNN. We also notice that frog has lower accuracy than the plain CRNN model. While the autoregressive decoding in frog could help enforce coherence between harmony components, it may require the full spectrum of the harmony components including key and degree. However, only root and quality were used in our experiments. Complete harmony information is difficult to collect so we believe Harana has a greater potential to leverage larger datasets in the future.

5.2 Segmentation Quality

As shown in Table 4, Harana provides improvement on segmentation quality compared to other models. Higher under-segmentation score of Harana means there are fewer missing boundaries in the estimation. Higher over-segmentation score shows that most detected boundaries are indeed true boundaries. An interesting observation is that the rule-based semi-CRF yields the most severe under-segmentation even though it is optimized on the segmentation boundaries. The reason for this might be that rule based-features are unable to clean noises such as the non-chordal notes and missing chordal notes in the input music but directly compute features from them. The noise in the features of short regions may be confused with the intrinsic noise of longer regions.

5.3 Ablation Studies

To show the effectiveness of each component of the architecture, we conduct additional ablation studies by removing each component. Table 2 summarizes the results. We can see that the full architecture achieves the best result overall. Among the missing components, semi-CRF leads to the largest performance drop. That confirms semi-CRF is an indispensable component to capture boundary information in harmony analysis. The attention module, although also helpful, produces relatively smaller performance gain. It is expected because after the neural front-end, the frame-level estimations to be aggregated may be already harmonically coherent; The attention module only helps to focus on the most representative frames. The effect of removing the absence score is less significant. Without it, the quality accuracy and overall segmentation quality even slightly improved. The phenomenon could result from the more difficult training objective. Inactive pitch class activations of the input music are an extreme scenario of noisy harmonic information. More data and a larger neural front-end might be needed to fully leverage the advantage of the absence score [33].

6. CONCLUSIONS

In this paper, we proposed an automated approach for harmonic analysis based on neural semi-CRF to jointly segment the harmonic regions and predict the labels. We developed a neural encoder and an attention mechanism to replace the conventional rule-based score function. We further proposed an absense score to improve the model robustness to imperfect harmony profiles. Experiments showed that our proposed architecture improves the performance on both frame-level accuracy and segmentation quality. Although our experiments focused on music input of symbolic format, the architecture could be adapted to audio input by simple modifications on the neural front-end. One limitation of the semi-CRF architecture is that it has quadratic time complexity with respect to sequence length so it is difficult to train the model on very long sequences. To capture the long-term dependency of harmony progression, more efficient sequence modeling methods could be explored in the future.

7. ACKNOWLEDGEMENTS

This work is partially supported by National Science Foundation grants No. 1846184 and 2222129. Frank Cwitkowitz would like to thank the synergistic activities funded by NSF grant DGE-1922591.

8. REFERENCES

[1] S. Kostka, D. Payne, and B. Almén, *Tonal harmony*. McGraw-Hill Higher Education, 2012.

[2] S. Bennett, "The process of musical creation: Interviews with eight composers," *Journal of Research in Music Education*, vol. 24, no. 1, pp. 3–13, 1976.

[3] W. F. Thompson, "Modeling perceived relationships between melody, harmony, and key," *Perception Psychophysics*, vol. 53, no. 1, pp. 13–24, 1993.

[4] B. Pardo and W. P. Birmingham, "Algorithms for chordal analysis," *Computer Music Journal*, vol. 26, no. 2, pp. 27–49, 2002.

[5] J. Pauwels, K. O'Hanlon, E. Gómez, and M. Sandler, "20 years of automatic chord recognition from audio," in *Int. Society of Music Information Retrieval Conf.*, 2019, pp. 54–63.

[6] S. Sarawagi and W. W. Cohen, "Semi-Markov conditional random fields for information extraction," in *Conf. on Neural Information Processing Systems*, 2004.

[7] T. Fujishima, "Real-time chord recognition of musical sound: A system using common lisp music," in *Int. Computer Music Conf.*, 1999.

[8] C. Harte and M. Sandler, "Automatic chord identifcation using a quantised chromagram," in *Audio Engineering Society Convention*, 2005.

[9] A. Sheh and D. P. Ellis, "Chord segmentation and recognition using em-trained hidden Markov models," in *Int. Society of Music Information Retrieval Conf.*, 2003, pp. 185–191.

[10] J. P. Bello and J. Pickens, "A robust mid-level representation for harmonic content in music signals," in *Int. Society of Music Information Retrieval Conf.*, 2005, pp. 304–311.

[11] J. A. Burgoyne, L. Pugin, C. Kereliuk, and I. Fujinaga, "A cross-validated study of modelling strategies for automatic chord recognition in audio," in *Int. Society of Music Information Retrieval Conf.*, 2007, pp. 251–254.

[12] X. Zhou and A. Lerch, "Chord detection using deep learning," in *Int. Society of Music Information Retrieval Conf.*, 2015.

[13] B. McFee and J. P. Bello, "Structured training for large-vocabulary chord recognition," in *Int. Society of Music Information Retrieval Conf.*, 2017, pp. 188–194.

[14] J. Park, K. Choi, S. Jeon, D. Kim, and J. Park, "A bidirectional transformer for musical chord recognition," in *Int. Society of Music Information Retrieval Conf.*, 2019.

[15] T.-P. Chen and L. Su, "Harmony transformer: Incorporating chord segmentation into harmony recognition," in *Int. Society of Music Information Retrieval Conf.*, 2019.

[16] J. Jiang, K. Chen, W. Li, and G. Xia, "Large-vocabulary chord transcription via chord structure decomposition," in *Int. Society of Music Information Retrieval Conf.*, 2019, pp. 644–651.

[17] N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent, "Audio chord recognition with recurrent neural networks," in *Int. Society of Music Information Retrieval Conf.*, 2013, pp. 335–340.

[18] F. Korzeniowski and G. Widmer, "A fully convolutional deep auditory model for musical chord recognition," in *Int. Workshop on Machine Learning for Signal Processing*, 2016, pp. 1–6.

[19] Y. Wu and W. Li, "Automatic audio chord recognition with MIDI-trained deep feature and BLSTM-CRF sequence decoding model," in *Int. Workshop on Machine Learning for Signal Processing*, 2019, p. 355–366.

[20] J. Park, K. Choi, S. Jeon, D. Kim, and J. Park, "A bidirectional transformer for musical chord recognition," in *Int. Society of Music Information Retrieval Conf.*, 2019.

[21] K. Masada and R. C. Bunescu, "Chord recognition in symbolic music using semi-Markov conditional random fields," in *Int. Society of Music Information Retrieval Conf.*, 2017, pp. 272–278.

[22] D. Tymoczko, M. Gotham, M. S. Cuthbert, and C. Ariza, "The romantext format: A flexible and standard method for representing roman numeral analyses," in *Int. Society of Music Information Retrieval Conf.*, 2019.

[23] G. Micchi, M. Gotham, and M. Giraud, "Not all roads lead to rome: Pitch representation and model architecture for automatic harmonic analysis," *Trans. of the International Society for Music Information Retrieval*, vol. 3, no. 1, pp. 42–54, 2020.

[24] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Łukasz Kaiser, and I. Polosukhin, "Attention is all you need," in *Conf. on Neural Information Processing Systems*, 2017.

[25] T.-P. Chen and L. Su, "Functional harmony recognition of symbolic music data with multi-task recurrent neural networks," in *Int. Society of Music Information Retrieval Conf.*, 2018, pp. 90–97.

[26] 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 *Int. Society of Music Information Retrieval Conf.*, 2015.

[27] N. N. López, *Automatic harmonic analysis of classical string quartets from symbolic score*. Doctoral dissertation, Universitat Pompeu Fabra, 2017.

[28] G. Micchi, K. Kosta, G. Medeot, and P. Chanquion, "A deep learning method for enforcing coherence in automatic chord recognition," in *Int. Society of Music Information Retrieval Conf.*, 2017, pp. 443–451.

[29] H.-W. Dong, K. Chen, J. McAuley, and T. Berg-Kirkpatrick, "Muspy: A toolkit for symbolic music generation," in *Int. Society of Music Information Retrieval Conf.*, 2020.

[30] G. Huang, Z. Liu, L. V. D. Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Conf. on Computer Vision and Pattern Recognition*, 2017, pp. 4700–4708.

[31] 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 *Int. Society of Music Information Retrieval Conf.*, 2014, pp. 367–372.

[32] C. Harte, *Towards automatic extraction of harmony information from music signals*. Doctoral dissertation, Queen Mary University of London, 2010.

[33] J. Clarysse, J. Hörrmann, and F. Yang, "Why adversarial training can hurt robust accuracy," in *Int. Conf. on Learning Representations*, 2023.

A DATASET AND BASELINE FOR AUTOMATED ASSESSMENT OF TIMBRE QUALITY IN TRUMPET SOUND

Alberto Acquilino* **Ninad Puranik*** **Ichiro Fujinaga** **Gary Scavone**

Department of Music Research, Schulich School of Music, McGill University
Centre for Interdisciplinary Research in Music Media and Technology (CIRMMT)
555 Sherbrooke St W., Montreal, Canada H3A 1E3

alberto.acquilino@mail.mcgill.ca, ninad.puranik@mail.mcgill.ca

ABSTRACT

Music Performance Analysis is based on the evaluation of performance parameters such as pitch, dynamics, timbre, tempo and timing. While timbre is the least specific parameter among these and is often only implicitly understood, prominent brass pedagogues have reported that the presence of excessive muscle tension and inefficiency in playing by a musician is reflected in the timbre quality of the sound produced. In this work, we explore the application of machine learning to automatically assess timbre quality in trumpet playing, given both its educational value and connection to performance quality. An extensive dataset consisting of more than 19,000 tones played by 110 trumpet players of different expertise has been collected. A subset of 1,481 tones from this dataset was labeled by eight professional graders on a scale of 1 to 4 based on the perceived efficiency of sound production. Statistical analysis is performed to identify the correlation among the assigned ratings by the expert graders. A Random Forest classifier is trained using the mode of the ratings and its accuracy and variability is assessed with respect to the variability in human graders as a reference. An analysis of the important discriminatory features identifies stability of spectral peaks as a critical factor in trumpet timbre quality.

1. INTRODUCTION

The significance of tone quality in brass musical instruments has attracted considerable attention due to its relevance in areas such as pedagogy and musical performance. Teaching aural discrimination skills of tone quality is indeed a major component of music training [1]. The emphasis placed on the development of good tone quality can be attributed to its close relationship with sound production efficiency. In brass instrument pedagogy, there is a widely held belief that the most efficient sounds are perceived as rich and round, while less efficiently produced tones tend to sound strained and shrill [2–4]. This implies that a method that can accurately and consistently distinguish the perceived tone quality in a brass instrument may hold significant potential in pedagogical applications, providing guidance to beginning students on how to achieve greater efficiency in sound production. However, understanding factors that contribute to the timbral quality of trumpet sound remains an unsolved challenge thus far.

Playing a trumpet tone involves a complex interplay between the musician's embouchure, oral cavity, and airflow [5]. It is a delicate balance in which even the slightest alteration in any component contributing to the creation of a tone can result in changes to the overall timbre [6]. The multi-variable interaction that contributes to the characterization of timbre makes defining its quality a challenging task [7].

Helmoltz was among the first to attempt providing insight into the audio properties related to the quality of a musical tone by proposing a direct relationship to the quantity and to the relative intensity of its constituent partials [8]. In an exploratory study using the trumpet as a case study, Madsen and Geringer identified the amplitude of the first overtone as a discriminatory feature between tones of differing sound quality [9]. Building on this finding, a subsequent perceptual study by Geringer and Worthy analyzed the tonal quality of the trumpet by altering the content of partials in the sound [10].

In recent years, the investigation of trumpet tone quality has emerged as an area of inquiry within the field of Music Information Retrieval. A pioneering study conducted by Knight et al. examined the potential of a model classifier to categorize trumpet tones into two, three, and seven classes [11]. This research assessed 56 single- and multidimensional audio features, as well as their correlations with human judgments, utilizing a dataset comprised of 239 individual sounds. Despite the relatively low accuracy of the resultant model, this foundational work has paved the way for subsequent advancements in the automatic assessment of brass tone quality, highlighting its potential in pedagogical applications.

A subsequent collaborative research project between

* Equal contribution

© A. Acquilino, N. Puranik, I. Fujinaga, and G. Scavone. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** A. Acquilino, N. Puranik, I. Fujinaga, and G. Scavone, "A Dataset and Baseline for Automated Assessment of Timbre Quality in Trumpet Sound", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

the Music Technology Group of Pompeu Fabra University (MTG-UPF) and KORG Inc. employed machine learning algorithms to evaluate various musical parameters of trumpet sounds, including timbre quality [12, 13]. To the best of our knowledge, this represents the most recent investigation in this domain. The researchers collected and analyzed a publicly accessible dataset containing 738 trumpet sounds. However, the findings revealed a weak correlation between the scores generated by the trained model and the rankings assigned by human evaluators, indicating significant room for improvement in the model's performance. Limitations were also identified in relation to the reference dataset, which lacked diversity by utilizing sounds from only two graduated trumpet players, and in the proposed interface for implementation in pedagogical contexts [14].

The current study aims to provide a comprehensive exploration of this subject, incorporating a complete dataset of sampled sounds and expert-generated labels. [1] Section 2 describes dataset collection and preprocessing, while Section 3 presents the machine learning training, results and visualization based on the most important feature.

2. MATERIALS

The dataset employed for training the proposed model comprises auditory samples gathered by the first author at various music institutions and master classes throughout Europe before the start of his academic program at the host institution. In total, 110 distinct trumpet performers were recorded under varying acoustic conditions. To encompass the complete spectrum of sound production efficiency levels, individuals from diverse backgrounds were recorded, including students and instructors from amateur music schools, arts universities, orchestral musicians, and international jazz and classical soloists.

The same recording system was utilized across all data acquisition sessions, specifically the IM69D130 Shield2Go evaluation board developed by Infineon Technologies, which is equipped with two Infineon IM69D130 Micro-Electro-Mechanical Systems microphones. Such a microphone exhibits an Acoustic Overload Point of 130 dB, allowing it to capture loud audio signals such as those produced by a trumpet without distortion or saturation. Moreover, the microphone offers a sufficiently flat and extensive frequency response ranging from 20 Hz to 20 kHz, thereby covering the entire audible spectrum.

The selected evaluation board was connected to a Raspberry Pi 4 Model B and a Raspberry Pi Model 3B+ for recording. A sampling rate of 48 kHz and 32-bit depth were used for the acquisition of audio data. The subsequent section provides a detailed account of the recording methodology employed for audio data collection.

2.1 Dataset acquisition methodology

The data acquisition process involved inviting each musician into a room with a fairly low ambient noise level. A microphone was positioned approximately 50 cm in front of the trumpet bell and 10 cm from its longitudinal axis. In most instances, a set of two microphones was employed concurrently to ensure data redundancy, mitigating the risk of data loss should a device malfunction occur during the recording session.

Participants were instructed to play isolated tones of approximately one-second duration over a chromatic scale ranging from E3 to B♭5 at three distinct dynamic levels: *piano*, *mezzoforte*, and *forte*, in their preferred sequence. Musicians utilized their personal instruments and mouthpieces and were not required to adhere to a reference pitch (e.g., A4 at 440 Hz) as timbral quality concerning sound production efficiency is anticipated to be independent of a reference pitch.

The inclusion of various dynamic levels aimed to enhance the dataset's variability, as the timbre of brass instruments is significantly influenced by loudness [15]. A digital sound level meter was positioned adjacent to the microphone, providing real-time decibel level readings during the recording. Trumpet players were given indicative reference levels of 85 dB, 105 dB, and 115 dB, corresponding to the *piano*, *mezzoforte*, and *forte* dynamic levels, respectively.

Despite the specified guidelines, the dataset exhibits several inherent variabilities:

- The sustain duration of the tones ranged from 0.7 to 4 seconds.
- The chromatic scale's range was contingent upon the performer's skill level. Generally, less proficient musicians struggled to produce tones in the high register, in which case they were instructed to play up to their highest achievable note.
- For beginner musicians, playing a chromatic scale in front of a microphone proved challenging at times. Some participants opted to perform legato notes rather than separate tones.
- Less skilled musicians often experience difficulty in controlling the instrument's dynamic range, resulting in the recommended dynamic levels being primarily adhered to by more proficient players.

During the recording sessions, the first author, who holds a degree in trumpet performance and has professional experience as a musician and instructor, assigned a preliminary grade of the overall sound production efficiency on a scale of 1 to 100 to each player. Figure 1 illustrates the distribution of assigned grades divided into four ranges (i.e., 0–25, 26–50, 51–75, and 75–100), demonstrating that a substantial number of players are represented in each category.

The dataset under examination was partitioned into discrete trumpet tones utilizing the *pyin vamp* plugin developed by Mauch and Dixon [16], yielding a collection of over 19,000 tones. Although the segmentation process demonstrated a degree of inaccuracy, with certain audio

[1] The dataset can be accessed at: `https://github.com/PNinad/ISMIR2023`

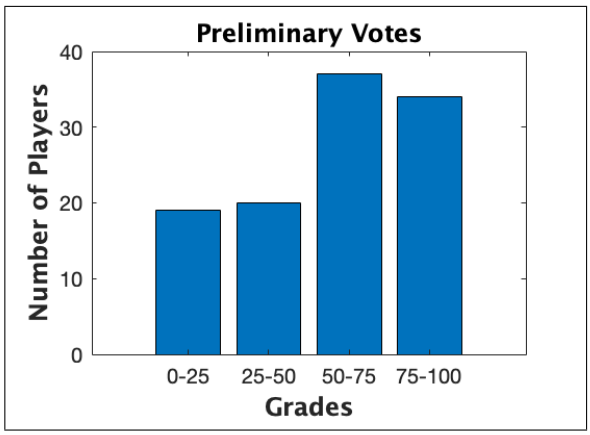


Figure 1. Distribution of recorded players according to the level of tone quality noted at the time of recording.

segments containing noise rather than trumpet tones, it nevertheless provided a satisfactory initial categorization of the data.

The following section outlines the methodology employed to prepare the dataset for label assignment by chosen evaluators.

2.2 Dataset preparation

Considering the approximate accuracy of the segmentation algorithm and the extensive nature of the overall dataset, it was decided to select a representative subset of the dataset for the manual examination of audio samples. To ensure that the whole range of tone quality is sufficiently represented, the subset was constructed of seventeen trumpet players such that five individuals had received a preliminary vote between 0–25 and four individuals with a grade between the other 3 ranges 26–50, 51–75, and 76–100 respectively. The first category was assigned one player more as the less experienced participants only partially cover the required chromatic scale, thus compensating for the lower representation of tones within this class. The selected subset encompassed 1,712 distinct trumpet tones.

It was decided to classify each tone into four categories based on their sound production efficiency, resulting in four classification levels: 1:poor, 2:fair, 3:good, and 4:excellent. This classification into four levels was employed with the intention of simplifying the label assignment process while retaining sufficient variability, as suggested by Wesolowski [17] and employed by Köktürk-Güzel et al. in a related research study [18].

The web interface shown in Figure 2 was subsequently developed to facilitate blind listening (i.e., without revealing the player's identity) and label assignment for each tone. The first author listened to all 1,712 sounds in the subset under analysis through the interface and assigned a label to each tone. The "Not a note" button enabled tagging of erroneously segmented sound samples which were filtered out to yield a dataset 1,481 clean samples.

The assignment of sound production efficiency class through anonymous listening to the audio samples in random order facilitated the allocation of a grade on a note-by-note basis, as opposed to providing an overall grade to the performance. This allowed for different grades to be assigned depending on the note if the level of sound efficiency varied along the chromatic scale. Additionally, the reliability of unbiased judgment could be assessed through a comparison with the preliminary grades assigned during the recording. The Spearman correlation coefficient between the two sets of grades was found to be 0.873 (P value<0.001), indicating the consistency of the author in assigning grades over time. This further indicates that players in general exhibit a consistent level of sound production efficiency along the chromatic scale.

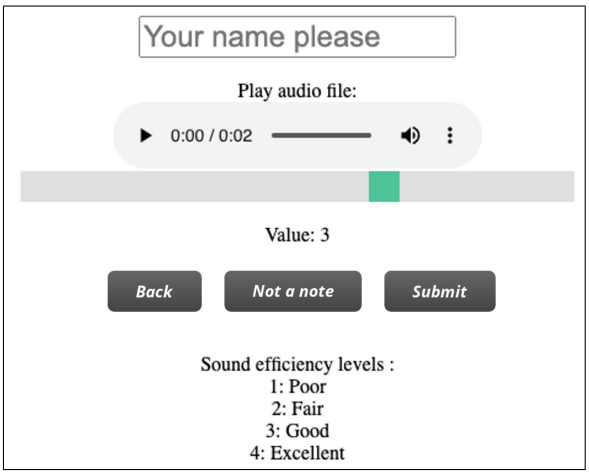


Figure 2. Interface for blind grading the trumpet tones.

2.3 Assessment labels

The cleaned dataset with 1,481 samples was subsequently presented to a panel of expert raters for evaluation via the described interface. A total of seven experts from different schools across Europe, North America and South America were chosen for the task. Among the raters, six were trumpet players, and one was a bass trombone player. All raters have professional experience as performers and/or teachers. This exploratory perceptual study was conducted online, with raters instructed to complete the task in a low-noise environment using professional headphones.

The rating sessions started with an introduction to the concept expressed by renowned brass instrument pedagogues, which asserts that rigidities in a trumpet player's body result in inefficiencies in playing, manifesting as a forced and strained sound. In contrast, a high-quality sound indicates efficiency of the embouchure and breathing muscles. Audio samples demonstrating extreme cases of this idea were presented and each rater confirmed their understanding of the concept and their ability to discern sound production efficiency in trumpet sounds based solely on audio information.

The dataset of 1,481 samples was split into two parts with 100 and 1,381 tones respectively. The raters first graded each of the 100 samples in approximately 15 minutes. After a 5 minute break, additional samples, randomly selected from the remaining 1,381 samples were presented

for evaluation. The raters continued to assess the trumpet tones until they experienced fatigue or until 90 minutes had elapsed from the beginning of the experiment. Table 1 displays the number of audio samples rated by each grader. Grader 1 corresponds to the first author who assigned the ratings manually by listening to all 1,481 samples in the subdataset, as described in the previous section. The set of 100 sounds were chosen such that they were equally distributed across the four classes, as determined from the labels by the author, and were used to ascertain the level of inter-rater reliability.

The next section describes the statistical analysis implemented on the data thus collected.

Grader ID	Graded tones
Grader 1	100 + 1381
Grader 2	100 + 401
Grader 3	100 + 206
Grader 4	100 + 312
Grader 5	100 + 383
Grader 6	100 + 366
Grader 7	100 + 564
Grader 8	100 + 491

Table 1. Number of individual tones evaluated by each grader.

2.4 Data analysis

The inter-rater reliability was assessed using the subdataset containing 100 tones graded by all the experts. Table 2 presents the Spearman ρ correlation coefficients with the corresponding P values for each pair of evaluators. As depicted in the table, all P values, representing the likelihood of obtaining the same results by chance, are less than 0.05.

The reported Spearman correlation coefficients range from 0.237 to 0.701. Notably, pairs including Grader 8 (the sole non-trumpet-playing expert) exhibited significantly lower correlation coefficients than all other pairs, potentially suggesting the significance of employing experts whose primary instrument aligns with the instrument under analysis for tasks of this nature. Due to the substantial differences in the ratings relative to the other raters, Grader 8 was deemed an outlier, and their results were excluded from further consideration. This adjustment increased Spearman ρ coefficients from 0.496 to 0.701, indicating fairly strong agreement among the judges [19].

Subsequently, a confusion matrix was computed for each evaluator, comparing the ratings assigned by that specific grader to the most frequently occurring (i.e., statistical mode) value in the ratings assigned by the seven evaluators for that specific tone. Cases where the mode was uncertain on one value were eliminated, resulting in 87 overall tones. The first seven subplots of Figure 3 display the resulting confusion matrices for each grader and their respective accuracy values (average f1 scores).

The next section describes the description of a model trained on the data obtained with reference to the variability of human assessment.

3. METHODOLOGY AND RESULTS

3.1 Audio Preprocessing and Model Training

The dataset preparation process described in Section 2.2 yielded a clean dataset with the audio samples of 1,481 tones. As a preprocessing step, the sound samples were first normalized to have a maximum signal amplitude equal to one. White noise at -60 dB was then added to the normalized audio to overcome the numerical errors (division by zero) encountered during feature extraction, without significantly altering the original signal. The audio features for each tone were then extracted using the Extractor algorithm from the Essentia library [20]. To reduce the computational complexity, only the statistical aggregates of the audio features (e.g., mean, variance, and mean of derivative) were utilized. Rhythm-based features were excluded since they were not deemed suitable for a timbre classification task. A total of 1,230 features were thus extracted to represent each audio sample.

As a first step, a Random Forest (RF) Classifier [21] was trained using the extracted audio features and labels provided by Grader 1, since Grader 1 had annotated each of 1,481 samples in the dataset. When the model was trained using the full set of audio features, a mean accuracy score of 78% was obtained in the 10-fold cross-validation. Using the model based feature selection in scikit-learn, the top 256 features were identified from an RF-classifier model trained using a 75%-25% train-test split of the dataset. Using just the top 256 features for training, the mean accuracy for the 10-fold cross-validation improved to 81.37%. The model thus obtained was implemented in a pedagogical application in a concurrent publication by the authors [22].

To eliminate the bias introduced by using a single grader, it was assumed that the most frequent label given by the expert graders is the true label. Only samples with at least two votes were used and samples which had equal number of votes for two labels by the expert graders were assumed to be ambiguous and were discarded from the dataset. With this approach, out of the 1,381 samples, 871 samples were deemed unambiguous. Similarly, 87 out of the 100 samples were unambiguous. An accuracy score of 59% was obtained on the test set of 87 samples for the RF model trained using the 871 samples as training set. The confusion matrices on the test-set for the different graders and the RF classifier can be seen in the bottom right subplot of Figure 3. It can be observed that most of the confusion is between the adjacent classes. Since the audio samples in the adjacent classes are in fact more similar to each other than the other classes, the errors seem to be reasonable, for both the graders and the model. While an accuracy score of 59% appears low, it is within the range of accuracy scores (53%–72%) of the human expert graders and it demonstrates that the extracted audio features could be used to classify the audio samples based on timbre quality.

The trained model was tested in real time by trumpet players and on labeled datasets different from the one in this study [12] showing promising generalisability.

Grader Pair	Grader						
	2	3	4	5	6	7	8
1	0.691*	0.668*	0.654*	0.645*	0.523*	0.638*	0.247***
2	-	0.701*	0.628*	0.650*	0.589*	0.650*	0.279**
3		-	0.599*	0.594*	0.496*	0.667*	0.237***
4			-	0.696*	0.650*	0.567*	0.349*
5				-	0.502*	0.637*	0.275**
6					-	0.524*	0.264**
7						-	0.353*

Table 2. Spearman ρ correlation coefficients between each pair of graders. Legend: * *p<.001*, ** *p<.01*, *** *p<.05*

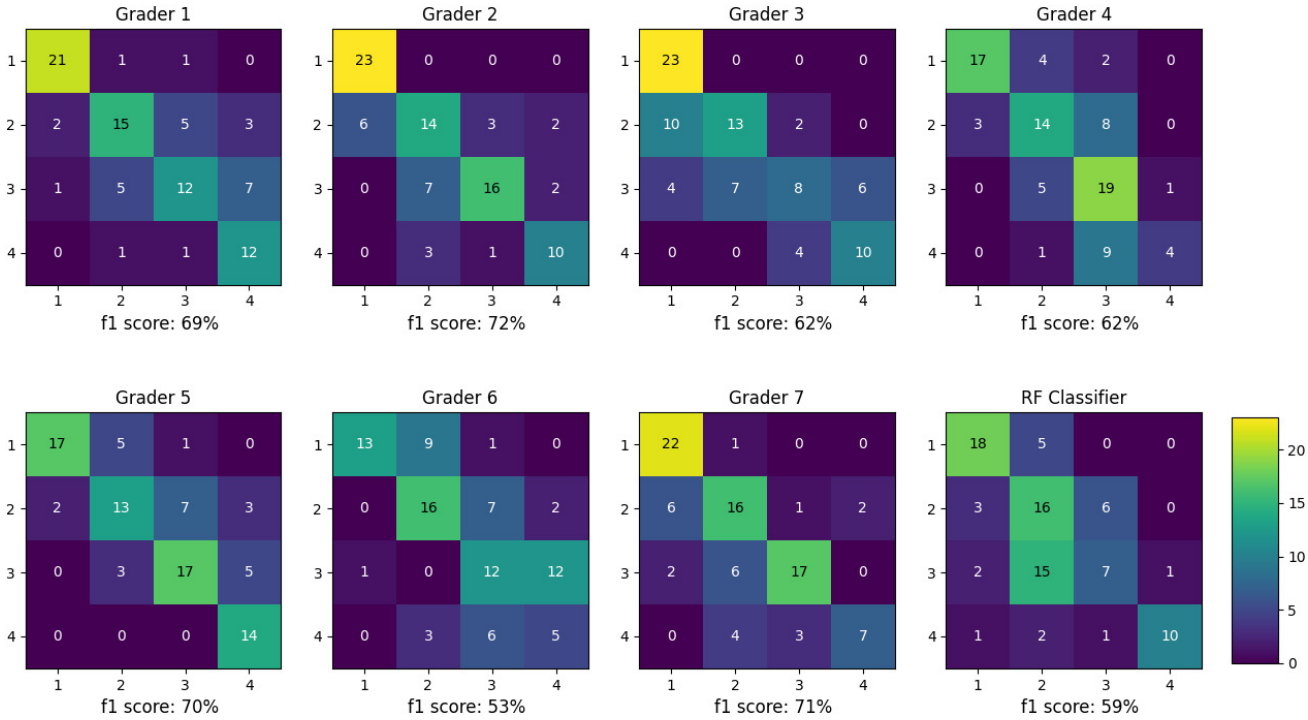


Figure 3. Confusion matrices with the predicted labels of each grader and of the trained RF classifier (horizontal axis) with respect to the true label as the mode of the assigned grade (vertical axis) and the corresponding f1 scores.

3.2 Feature importance

Due to a slightly subjective nature of the problem, there is considerable variability in the labels by human experts. Hence, very high classification accuracy scores cannot be achieved even with sophisticated machine learning models. However, even with a moderately accurate classifier, analysis of the most important features could help to develop an intuitive understanding of good quality timbre in trumpet sounds.

One of the main reasons to choose the Random Forest Classifier algorithm was that it gives access to the importance of each feature in the classification task. The feature importance scores for the classification are available as a model property in the scikit-learn implementation of the Random Forest algorithm [23]. The top 20 observed features are listed in Table 3.

Many of the top features are based on the mean of the derivative 'dmean' and the mean of the double derivative 'dmean2', suggesting that the change in the spectrum accross time is a crucial factor in the perception of the timbre quality. Notably three of the top features namely lowLevel.spectral_complexity.dmean, lowLevel.spectral_complexity.dmean2 and lowLevel.spectral_complexity.dvar are related to the time varying properties of the same underlying feature of spectral complexity.

A scatter plot of the lowLevel.spectral_complexity.dmean and lowLevel.spectral_complexity.dmean2 features considering only the best and worst class samples is shown in Figure 4. It is apparent that just this pair of features is quite successful in discriminating between the best and worst samples. Since both features are statistical aggregates of the spectral complexity feature, the raw feature was explored to develop a visualization of the sound production efficiency as described in the following subsection.

3.3 Visualization based on Spectral Complexity

Spectral complexity is based on the number of peaks in the spectrum of a time window [24]. The Essentia implementation of this feature considers the spectral peaks only up

Audio feature	Score (%)
lowLevel.spectral_complexity.dmean	1.381
lowLevel.scvalleys.mean_5	1.182
lowLevel.spectral_complexity.dmean2	1.049
lowLevel.spectral_complexity.dvar	0.897
lowLevel.sccoeffs.var_5	0.648
lowLevel.scvalleys.mean_3	0.636
lowLevel.sccoeffs.stdev_5	0.622
lowLevel.scvalleys.median_5	0.594
lowLevel.spectral_spread.dmean	0.570
sfx.tristimulus.dmean2_2	0.561
lowLevel.sccoeffs.median_4	0.531
lowLevel.sccoeffs.dmean2_3	0.496
lowLevel.scvalleys.median_3	0.492
lowLevel.barkbands.dmean_25	0.478
lowLevel.pitch_ instantaneous_confidence.dmean2	0.465
lowLevel.spectral_flux.dmean	0.465
lowLevel.spectral_complexity.dvar2	0.425
lowLevel.sccoeffs.mean_4	0.424
lowLevel.scvalleys.mean_2	0.412
lowLevel.spectral_complexity.stdev	0.402

Table 3. Top 20 features ranked by importance in the Random Forest Classifier.

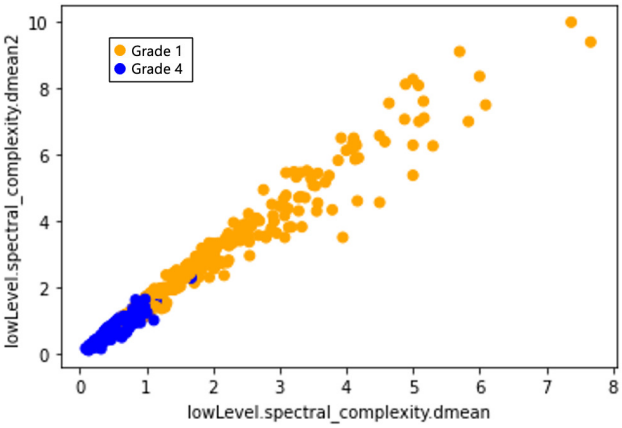


Figure 4. Scatter plot depicting the spectral complexity based features for best (blue) and worst (orange) class samples.

to 5 kHz. From the spectra of the collected dataset, the presence of harmonic peaks at frequencies higher than 5 kHz was evident. It was therefore decided to implement the spectral complexity considering the entire audible frequency range. To enhance peak detection accuracy, prior knowledge of the fundamental frequency 'f0' of the tone was utilized to search for spectral peaks exclusively in the vicinity of the integer multiples of the f0 frequency. For a normalized audio, peaks with signal energy less than -40dB were discarded to reduce noise. An FFT-bin mask was generated by assigning the value of one to the FFT bin if a peak was detected in it while all other bins were assigned a value of zero, thus generating a visualization to track the peaks across the analysis time windows.

Figure 5 shows the visualization for two representative sounds. It is evident that for sounds rated as excellent quality, the spectral peaks consistently lie in the same FFT-bin across time, leading to flat horizontal lines in the visualization. Whereas for sounds rated as poor quality, the spectral peaks show unsteadiness, particularly at the higher harmonics, which leads to broken and wavy lines in the visualization. The total number of peaks could be more or less depending on the f0 frequency of the note and the loudness. However, it appears that the perception of timbre quality is correlated to the steadiness of the peaks rather than their total number. A real-time implementation of this visualization could offer invaluable feedback on the efficiency of sound production, greatly benefiting new trumpet students who are still developing their auditory skills.

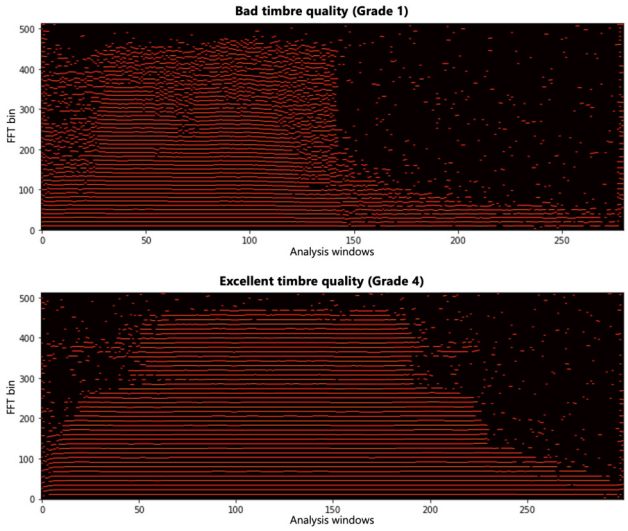


Figure 5. Visualization of the temporal evolution of spectral peaks for trumpet sounds rated as low-quality (top) and high-quality (bottom) timbre.

4. CONCLUSIONS

In this paper, we introduced the importance of timbre quality in trumpet performance and pedagogy. With an aim to develop an automated tool for the assessment and visualization of trumpet tone quality, an extensive dataset of trumpet tones was collected and manually graded with the help of experts. Through the inter-grader analysis presented, it was shown that while there are some differences in timbre preferences, most experts generally concur in differentiating the different levels of trumpet tone quality.

Random Forest Classifier models trained using extracted audio features were found to have accuracy scores comparable to the accuracy scores of human experts. Features based on spectral complexity were observed to have very high importance in the models trained for the task of trumpet timbre discrimination.

A representation based on the harmonic peaks in the spectrum was developed to visualize the timbre quality. The proposed visualization suggests that the stability over time of spectral partials plays an important role in discriminating the timbre quality of trumpet sounds.

Future research aims to incorporate the developed model and visualization in a pedagogical application and assess its efficacy in music classrooms.

5. ETHICS STATEMENT

Ethical approval for the study, including consenting procedures, was granted by the Research Ethics Board Office of McGill University following the guidelines of the Canadian Tri-Council Policy Statement.

Acknowledgments

This work was made possible with the support of a CIRMMT Student Award and a Tomlinson Doctoral Fellowship.

The authors thank the foundational contribution of Mirko d'Andrea and Emanuela Bussino for the audio data collection stages, as well as all the volunteers and colleagues who supported this research.

6. REFERENCES

[1] A. L. Simmons, "The relationship between prospective teachers' tone quality evaluations and their knowledge of wind instrument pedagogy," *Applications of Research in Music Education*, vol. 23, no. 2, pp. 42–51, 2005.

[2] A. Jacobs and B. Nelson, *Also Sprach Arnold Jacobs: A Developmental Guide for Brass Wind Musicians*. Polymnia Press, 2006.

[3] J. Thompson, *The Buzzing Book Complete Method; Trumpet or Other Brass Instruments*. Editions BIM, 2003.

[4] K. Steenstrup, *Teaching Brass*. Der Jyske Musikkonservatorium, 2007.

[5] F. G. Campos, *Trumpet Technique*. Oxford: Oxford University Press, 2005.

[6] S. Levarie and E. Levy, *Tone : A Study in Musical Acoustics. 2d ed.* Kent, Ohio: Kent State University Press, 1980.

[7] S. McAdams, S. Winsberg, S. Donnadieu, G. Soete, and J. Krimphoff, "Perceptual scaling of synthesized musical timbres: Common dimensions, specificities, and latent subject classes," *Psychological Research*, vol. 58, pp. 177–912, Dec. 1995.

[8] H. von Helmholtz, *On the Sensations of Tone as a Physiological Basis for the Theory of Music*. New York: Dover: Trans. By A. Ellis, 1977.

[9] C. Madsen and J. Geringer, "Preferences for trumpet tone quality versus intonation," *Bulletin for the Council for Research in Music*, vol. 46, pp. 13–22, 1976.

[10] J. M. Geringer and M. D. Worthy, "Effects of tone-quality changes on intonation and tone-quality ratings of high school and college instrumentalists," *Journal of Research in Music Education*, vol. 47, no. 2, pp. 135–149, 1999.

[11] T. Knight, T. Upham, and I. Fujinaga, "The potential for automatic assessment of trumpet tone quality," in *Proceedings of the International Society for Music Information Retrieval Conference*, 2011, pp. 573–578.

[12] G. Bandiera, O. Romani, H. Tokuda, W. Hariya, K. Oishi, and X. Serra, "Good-sounds.org: A framework to explore goodness in instrumental sounds," in *Proceedings of the International Society for Music Information Retrieval Conference*, New York, 2016.

[13] O. Romani, H. Parra, D. Dabiri, H. Tokuda, W. Hariya, K. Oishi, and X. Serra, "A real-time system for measuring sound goodness in instrumental sounds," in *138th Audio Engineering Society Convention, AES*, Warsaw, Poland, 2015, pp. 1106–1111.

[14] A. Acquilino and G. Scavone, "Current state and future directions of technologies for music instrument pedagogy," *Frontiers in Psychology*, vol. 13, 2022.

[15] D. Luce and M. J. Clark, "Physical correlates of brass-instrument tones," *The Journal of the Acoustical Society of America*, vol. 42, pp. 1232–1243, 1967.

[16] M. Mauch and S. Dixon, "Pyin: A fundamental frequency estimator using probabilistic threshold distributions," in *Proceedings of the 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 659–663.

[17] B. C. Wesolowski, "Understanding and developing rubrics for music performance assessment," *Music Educators Journal*, pp. 36–42, 2012.

[18] B. E. Köktürk-Güzel, O. Büyük, B. Bozkurt, and O. Baysal, "Automatic assessment of student rhythmic pattern imitation performances," *Digital Signal Processing*, vol. 133, 2023.

[19] A. Williamon, J. Ginsborg, R. Perkins, and G. Waddell, *Performing Music Research: Methods in Music Education, Psychology, and Performance Science*. Oxford: Oxford University Press, 2021.

[20] D. Bogdanov, N. Wack, E. Gómez, S. Gulati, P. Herrera, O. Mayor, G. Roma, J. Salamon, J. Zapata, and X. Serra, "Essentia: An audio analysis library for music information retrieval," in *Proceedings of the International Society for Music Information Retrieval Conference*, 2013, pp. 493–498.

[21] T. K. Ho, "Random decision forests," in *Proceedings of 3rd International Conference on Document Analysis and Recognition*, vol. 1. IEEE, 1995, pp. 278–282.

[22] A. Acquilino, N. Puranik, I. Fujinaga, and G. Scavone, "Detecting efficiency in trumpet sound production: proposed methodology and pedagogical implications," in *Proceedings of the 5th Stockholm Music Acoustic Conference*, Stockholm, Sweden, 2023.

[23] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, A. Müller, J. Nothman, G. Louppe, 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, pp. 2825–2830, 2011.

[24] C. Laurier, O. Meyers, J. Serrà, M. Blech, P. Herrera, and X. Serra, "Indexing music by mood: design and integration of an automatic content-based annotator," *Multimedia Tools and Applications*, vol. 48, no. 1, pp. 161–184, 2009.

VISUAL OVERVIEWS FOR SHEET MUSIC STRUCTURE

Frank Heyen **Quynh Quang Ngo** **Michael Sedlmair**

VISUS, University of Stuttgart, Germany

{frank.heyen, quynh.ngo, michael.sedlmair}@visus.uni-stuttgart.de

ABSTRACT

We propose different methods for alternative representation and visual augmentation of sheet music that help users gain an overview of general structure, repeating patterns, and the similarity of segments. To this end, we explored mapping the overall similarity between sections or bars to colors. For these mappings, we use dimensionality reduction or clustering to assign similar segments to similar colors and vice versa. To provide a better overview, we further designed simplified music notation representations, including hierarchical and compressed encodings. These overviews allow users to display whole pieces more compactly on a single screen without clutter and to find and navigate to distant segments more quickly. Our preliminary evaluation with guitarists and tablature shows that our design supports users in tasks such as analyzing structure, finding repetitions, and determining the similarity of specific segments to others.

1. INTRODUCTION

Common music notation can be considered as a special visual encoding to convey music, including instructions on how to perform it. Despite its compactness and detailed information, a music sheet is hard to analyze for novice musicians [1]. Moreover, it contains lots of information that is hard to display at once without visual clutter or getting too small – getting an overview is tricky. When pieces contain repeating sections such as a chorus, much information is redundant. Even with abbreviations that denote repetitions, such as a double bar with colon, da capo, or al segno, a complex structure can lead to tedious navigation.

Recent work [1–3] strove to enrich notation to better convey music-theoretical information and patterns. In this paper, we instead focus on quickly gaining an overview of structures such as similarities and repetitions. This overview is meant to support learning, or teaching a music piece, as musicians often have to remember which segment of a piece they have to play when and how often, information that can be obscured in classical sheet music notation. According to the visualization principle "eyes beat memory" [4], we argue that the current notation leaves room for further optimization.

Toward this goal, we propose to visually enrich sheet music by mapping calculated similarities among segments of the sheet music, such as sections or bars (measures), to colors. To fit the notes of a whole piece onto a screen while remaining legible, we propose compact alternative encodings to common music notation that allow for overview and easier navigation without having to scroll or change pages. This work focuses on guitar tablature of Western music, which is easier to represent compactly and often features more repeating parts than other kinds of sheet music. However, we argue that our general method of mapping similarities to color can also help with other kinds of music. We conducted a preliminary qualitative evaluation through pair analytics with four guitarists. The results indicate that our design supports tasks such as summarizing structure, finding repetitions, and analyzing similarity.

In summary, we contribute 1) the exploration of novel representations of sheet music for easier overviews, specifically a method for mapping similarity among components of a music sheet to color, and 2) a preliminary pair analytics study with four guitarists. We further provide source code and a web app where users can try their own sheet music in MusicXML [5] at visvar.github.io/sheetmusic-overviews.

2. RELATED WORK

Similarity in music concerns many dimensions such as cognition, perception, tempo, pitch, and more. Therefore, existing metrics use different approaches, including a continuous representation of notes [6], a geometrical metric [7], shapes of curves [8], and a graph-based metric for harmony [9]. Janssen et al. [10] evaluated melodic similarity metrics using human annotators and a survey [11] defined eight criteria for symbolic melodic similarity. The overall aim of the above work is to query pieces in a database. In contrast, our work focuses on supporting the structural analysis of a single piece, by visualizing similarities within it. While our design allows integrating any metric, its main purpose is demonstrating how visualization can support sheet music analysis. We thus use a simpler symbolic metric to instantiate our design.

There is a broad range of music-related visualization [12, 13], including structure [14–22]. However, some visual encodings, such as one based on Tonnetz [21], re-

© F. Heyen, Q. Q. Ngo, and M. Sedlmair. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).
Attribution: F. Heyen, Q. Q. Ngo, and M. Sedlmair, "Visual Overviews for Sheet Music Structure", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

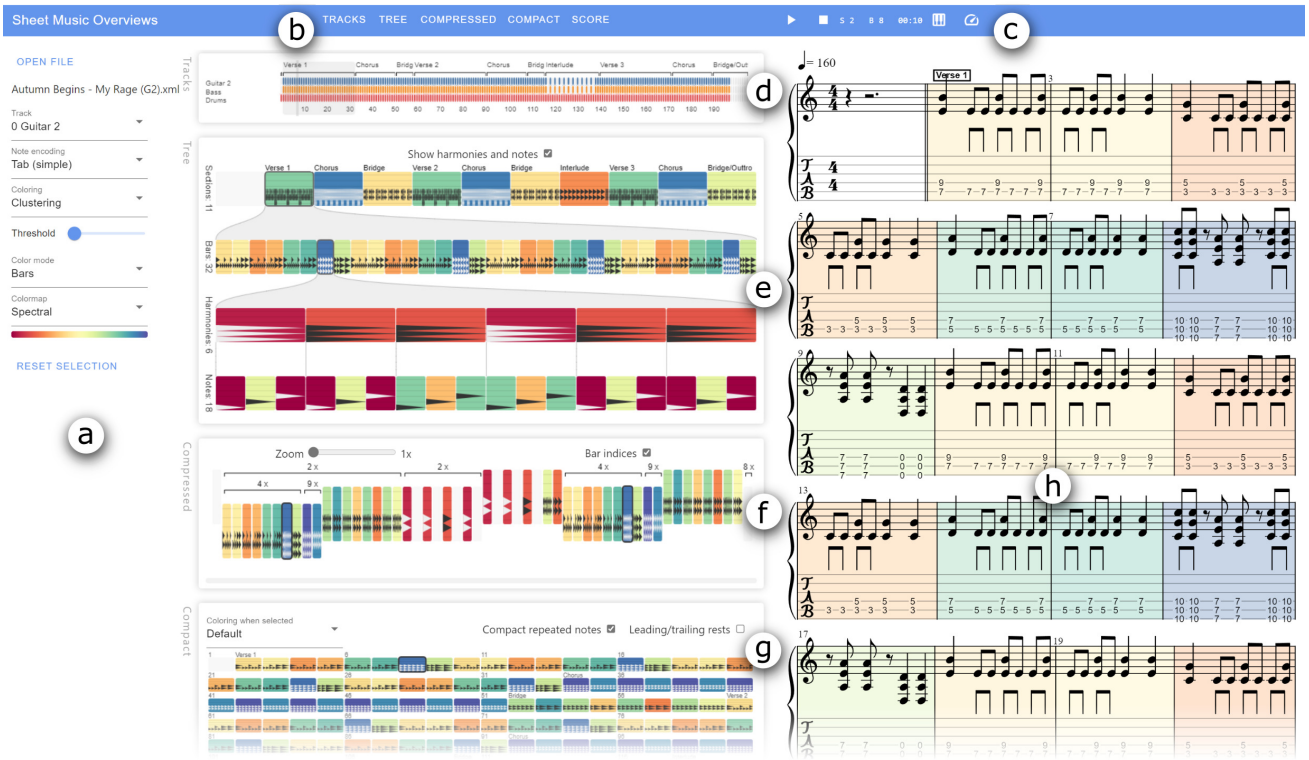


Figure 1: Screenshot of our design with all views (cut): a) data and visualization options, b) view selection, c) player, d) instrument/track overview, e) structure hierarchy, f) compressed repetitions, g) compact sheet, and h) complete score.

quire knowledge of music theory. Similar to our approach, *MoshViz* [17] focuses on visual analysis in an overview-detail fashion, but without considering perceptual or cognitive aspects. Our design allows encoding any similarity metric (mathematical or perceptual) with colors, to make sheet music easier to understand. Closest to our work is a structure visualization that uses dimensionality reduction (DR) to map audio features to color [18], which inspired us to design a similar mapping for sheet music.

Augmented sheet music adds visual components to common music notation to increase expressiveness. Related work augments a music piece with radial note histograms, to facilitate analyzing harmonic patterns [2], or visualizes rhythm through color-coded sunbursts [1]. Miller et al. [3] combined both approaches, but do not address supporting performance preparation tasks. Only little research supports instrument learning and composition. Bunks et al. [23] use color for reference keys on a tabular layout to support jazz improvisation. Others augment sheet music with lines and ellipses to support error detection in composition [24] or pianists in identifying mistakes while practicing [25, 26]. In this work, we also support learning by aiding music reading before and during practice.

3. DESIGN

We first introduce the tasks we want to support with our approach. Then, we describe how we compute similarities and map them to colors and how we represent sheet music visually (Figure 1).

3.1 User Tasks

Our overall goal is to improve the efficiency of reading sheet music, by sparing users the need to search for certain segments or memorize patterns. We want to reveal potentially interesting patterns that are hard to infer from the bare sheet music itself but could be helpful for better understanding or practicing a piece. More specifically, we want to support the following tasks: ($\mathbf{T_1}$) understand the *overall structure* of a piece, ($\mathbf{T_2}$) detect repetitions, which means to spot *where* something repeats *how often*, and detect *repetitions nested within* repetitions, ($\mathbf{T_3}$) *compare* multiple segments regarding their similarity.

3.2 Color Mappings

Similarity metrics. Our approach works with any metric that takes notes and returns a scalar similarity score. We compare non-overlapping segments of the piece, which can be bars, pre-defined sections read from the MusicXML file, or the result of an automatic segmentation (the latter is not implemented).

In our related work section, we discussed existing similarity metrics for symbolic music. Some of these metrics do not support polyphony or require complete scores or additional annotations (such as chords) or assumptions on musical meaning. Metrics that are based on western tonal harmony [27, 28] would also not generalize to various cultures and genres. Therefore, we designed the following simple but robust algorithm: First, the notes of a segment

are sorted by their start time and those with equal start time by pitch ascending. Mapping each note to its pitch then results in one sequence of integers for each segment. We then compute a similarity matrix by calculating the *Levenshtein distance* [29] for all possible pairs of segments, equals the minimum number of pitches one would have to insert, delete, or replace, to transform the first sequence into the second.

We further compute similarities between all sets of notes that have the same start time, which we refer to as harmonies. For these sets, we only use the notes' pitch class (disregarding octaves) and compute the *Jaccard index* [30], which equals the ratio of intersection over union.

Mapping. Once we have a similarity matrix, we can create a color mapping that respects these similarities. We explored three alternative methods that use either a one-to-many comparison, dimensionality reduction, or hierarchical clustering. Figure 2 shows an example for our similarity-based color mapping, Figure 3 summarizes our different mapping strategies.

The first method colors bars by their similarity to a selected bar. This selection is made by the user or automatically when playing the piece, where the currently played bar is selected. To obtain colors, we linearly map the similarities to a color scale. Another mode only colors bars that the metric considers identical to the selected one, allowing users to quickly spot where and how often it repeats.

Our second method uses dimensionality reduction (DR), a method commonly used to transform data from a high-dimensional space to a lower-dimensional one. Usually, the target space is two- or three-dimensional, such that data points can be shown on a screen. We instead project onto a one-dimensional space that we can then linearly map to a color scale. As we do not have concrete positions in a space, but only the similarities between them, we chose *multi-dimensional scaling* (MDS) [31] that accepts a similarity matrix as input. Furthermore, MDS optimizes the computed projection to preserve these similarities, leading to a coloring optimized for these.

As an alternative to MDS, we designed a method that clusters similar segments together and then assigns each cluster one color such that similar clusters have similar colors. Using our similarity matrix, we compute *hierarchical agglomerative clustering*, which gives us a binary tree. We then sort the tree's leaves from left to right, as leaves that are closer together are more similar, and map them in this order to a color scale. Compared to the above method using MDS, the resulting colors are easier to distinguish but represent similarities less accurately. Using a similarity threshold, users can steer the number of clusters and therefore colors, to choose a trade-off between detail and overview.

Color scales. Research on perception proposed a range of color scales specifically designed for visualization. Since there are different irreconcilable goals, no scale is appropriate for all tasks. While multi-hue scales such as rainbows have been criticized [32], they have been shown to work well for some circumstances [33–35]. For users

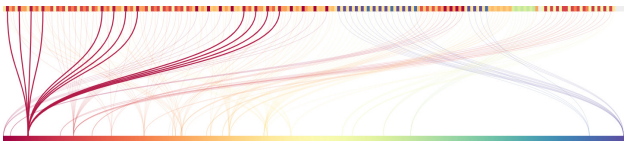


Figure 2: Example of similarity-based color mappings. *Top:* rectangles represent all bars of a piece, from left to right in the order they occur. *Bottom:* a color scale. Curves connect each bar to its color. In this figure, the curves of a single color are highlighted through a stronger opacity to show how they connect to identical bars.

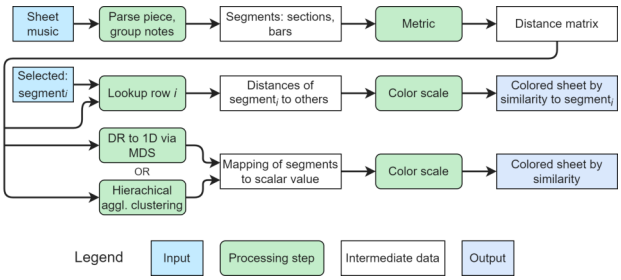


Figure 3: We compute similarity-based colors for extracted segments via direct comparison, dimensionality reduction, or clustering. A *segment* can be any sequence of notes in the piece, such as a bar or a pre-defined section. The *selected segment* is chosen by the user to compare it to all others.

with a color vision deficiency, scales with fewer hues and, therefore, less discernible colors can be used, such as cividis [36]. When color is used to compare different values or intervals, a color scale needs to accurately represent similarities between values. For this task, single-hue scales or interpolations between two hues are appropriate but further reduce the number of discernible colors. Although the number of distinguishable colors is small, there are enough for our use case, as the number of different segments in a piece is limited. Since colors are distributed by similarity, indistinguishable colors should only be assigned to very similar segments. To accommodate different user needs, we choose a broad multi-hue scale (spectral) as default but also provide more accessible ones; for direct comparison, we choose a single hue scale (blues) (Figure 4).

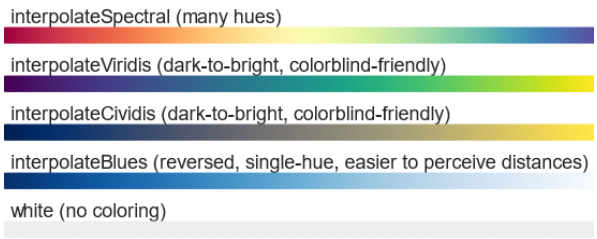


Figure 4: Some of the included color scales, taken from D^3 [37]. The choice depends on the current task and individual limitations of the user's color vision.

Figure 5: A pattern of repetitions with different endings. Bar 16 is colored blue, 24 and 40 are the same yellow, and bar 32 is green.

3.3 Visual Encodings

Layout enrichment for common music notation. We designed several visual encodings to address different tasks and reveal different kinds of patterns. The most straightforward encoding is to display the full sheet music as the common notation that is familiar to musicians and represents the complete information. We enrich this display by adding colored, semi-transparent rectangles on top of the segments, for example, one for each bar (Figure 5). The reduced opacity makes colors brighter than in other views but improves the readability of the notation. This coloring helps more quickly see where a bar repeats ($\mathbf{T_2}$), as the user only has to compare bars with similar colors ($\mathbf{T_3}$). Even when two different bars were assigned similar colors, this process allows for ruling out many others. This encoding suffers from the same limitations as non-colored sheet music. Due to its highly detailed nature, fitting the complete piece on the screen at once would lead to small and illegible visuals. Therefore, we designed simplified, filtered, and compressed alternatives, which we explain in the following paragraphs.

Note display. In most views, we represent notes by blocks that are positioned horizontally by start time and have a width proportional to the note's duration, to visually indicate timing and rhythm (Figure 6). The first mode displays the notes as triangles in a piano roll, allowing it to represent music for any instrument, but less readable than other encodings. A second mode displays guitar tablature, where each row stands for one string, and the third adds fret numbers for more detail. We focus primarily on guitar tablature in this work, but new encodings resembling other instruments' common notations could extend our approach. Depending on their size, our encodings become hard to read but still reveal coarse patterns more clearly than detailed notation. In order to show the whole piece at once ($\mathbf{T_1}$), without filtering or compression, we display the full score with the above encodings (Figure 1g).

Hierarchy. Most music pieces have a hierarchical structure in the form of sections such as verse and chorus, spanning multiple bars, each with none to a few notes, which might be grouped in harmonies (notes played at the same time). We visualize this structure as a tree, where users can select a node to show only its children in the level below (Figure 1e). This representation supports gaining an overview ($\mathbf{T_1}$) and allows navigating the sheet music more conveniently. The colors are level-specific, such that they only represent similarities within, not between, the levels. Notes have their own color map that is not based on similarity but still allows to spot repetitions or patterns such as sequences of increasing pitch ($\mathbf{T_2}$).

(a) Piano roll. (b) Tab (simple). (c) Tab with frets.

Figure 6: Note display: a) Piano rolls can represent any music but lack additional information. b, c) Tablature either simplified or with fret numbers. Notes are drawn in black or white depending on the background's luminosity.

Compressed multi-level repetitions. Music pieces might have another hierarchical structure regarding *repetitions* when a repeating segment contains repeating sub-segments ($\mathbf{T_{1,2}}$). Similar to data compression, this allows us to create a more compact representation, by only displaying a repeating segment once and annotating it with its number of repetitions. Doing this recursively results in a tree where each leaf is a bar of the music piece, and each inner node contains the following information: A pre-fix child, a repeated child with its repetition count, and a post-fix child, where pre- and post-fix can be empty. The visual encoding we chose for this data structure uses our compact note encoding (Figure 6) for the leaves and brackets for the inner nodes (Figure 7). Numbers above the bars denote the index of their first occurrence, allowing to spot recurring ones that are farther apart.

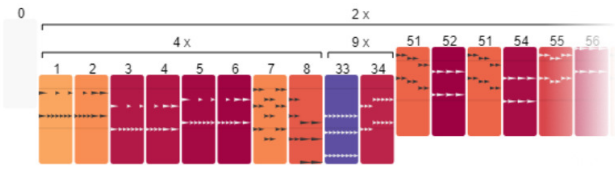


Figure 7: Our compressed view shows repetitions as nested blocks (cut). Note, that bar 51 appears two times, as it equals bar 53.

Workflow and interaction. We envision musicians using our interface primarily while learning a new piece, where they first get an overview and then take a closer look at the detailed sheet music. During navigation, reading, and playing, they could use overviews as 'minimaps' that provide context for what they are currently focusing on. As all views are linked, clicking on a bar in any view allows users to highlight or jump to a certain bar in all other views.

4. EVALUATION

We chose a pair analytics setup over a comparative user study, as most related work focuses on different tasks or data, which does not allow fair comparison. Furthermore,

instead of quantitative usability ratings and time measurements, we were more interested in qualitative feedback on *how* guitarists would use our design and *what limitations* they encounter. In pair analytics [38], designers and participants collaboratively analyze the participants' data, allowing designers to evaluate a design without needing to teach participants how to use it, saving them time and ensuring they use all features appropriately.

Our participants (P1–P4) have experience with reading guitar tablature. P1 has played guitar for 15 years and regularly teaches it and P2 has played drums for 5 years and guitar occasionally. P3 and P4 have played guitar for 16 and 12 years. P1 and P3 have full color vision, P2 and P4 have a slight red-green deficiency. All but P3 were familiar with visualization. We met with each participant for roughly 1.5 hours. After an introduction to our interface, we looked at guitar tablature of their choice together, encouraging them to use different features and think aloud.

Our participants found the coloring generally helpful: *"I have played classical pieces with 8 or 12 pages ... you searched, with your teacher, made annotations with a pen ... 'here it's that part again' ... if it's only black-on-white, you're blind at some point" (P3)*. They were able to detect various patterns: *"the color indicates a new segment" (P1)* ($\mathbf{T_1}$), *"always two bars one note, two bars the other note, ..." (P1)* (Figure 8). One interesting example was a pattern where the same segment was repeated four times, with a different ending each time, except for the fourth that equals the second (Figure 5, $\mathbf{T_{2,3}}$).

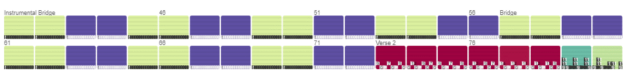


Figure 8: A simple alternating pattern with bars that repeat as *AABB* multiple times.

In some cases, our current similarity metric did not work well: *"here, I'm sure it's different, but the color is not quite different ... if you check the color carefully, I think you can see it" (P1)*. We proposed coloring annotated sections of the sheet music by their similarity: *"That is already very useful, because ... when I'm [teaching] and want to show segments, then I always have to mark them by hand. This is doing it for me" (P1)*. While we currently colorize either by section or bar, P3 suggested alternatively coloring by sequences of bars that occur together multiple times, such as riffs or motifs. Interestingly, our coloring allows users to quickly guess the effort needed to learn a piece: *"The colors show what you already learned" (P3)*, *"For me it looks like I practice this purple bar ... and then I practice these yellow and green bars and then I can play 90 percent of the song" (P1)* ($\mathbf{T_{1,2}}$). P4 suggested ignoring bars with only a single note or chord when coloring, as these are less interesting. Instead, they proposed coloring differently transposed versions of the same pattern more similarly and allowing users to manually adjust the color of a set of identical bars.

We also compared coloring via DR versus via clustering. When trying clustering, P3 first pondered *"I think [coloring via clustering] is easier to understand ... as you can really see that it's different"* but then concluded that *"it's difficult, both have pros and cons ... now if I would search [by color], it would be more difficult to spot"* ($\mathbf{T_{2,3}}$). P4 suggested to additionally vary the color's brightness for different segments in the same cluster, to also reveal similarity within clusters ($\mathbf{T_3}$).

Most patterns were visible with all color scales, although less clearly with those using fewer hues, so users with color vision deficiencies can also benefit from our design. Even though P4 has a slight red-green deficiency, they wanted to use the default *spectral* scale for most of the study. When trying out other scales, they told us that it indeed makes a good default, as it has fewer hues than rainbow scales. P4 further preferred *viridis* over *cividis*: *"here I see better"*. When turning of colors, P3 was astonished: *"here you see how white it is! When looking at colors for so long, you see for the first time how ugly white it is"*.

The simplified and full tablature encodings still reveal characteristics: *"These are power chords, right?" (P2)*, *"This is a power chord on the second fret ... and that should be A minor" (P3)*. For our hierarchy view, P4 found that *"the tab encoding doesn't help much, the simplified tab and piano roll work much better"*. Especially, since with the piano roll *"you can see well which bars have similar notes" (P4)* ($\mathbf{T_3}$).

In our hierarchy view, clicking on different sections allows users to quickly compare them: *"Main riff and verse is almost the same, it's labeled as 'main riff' because it is ... without singing" (P1)* ($\mathbf{T_3}$). This also shows a drawback of sheet music, where repetition signs often apply for all instruments at once, so if one repeats and another does not, the first instrument's notes will show up redundant. During the comparison, we found that P1 labeled the sections incorrectly, as one had a few more bars that actually belonged to the following section: *"we found that we labeled it wrong, that's good!" (P1)* ($\mathbf{T_3}$). Switching between different sections allows comparing them: *"here's again a verse, but a little different ... back there, this bar is repeated ... this chorus is much longer" (P3)* ($\mathbf{T_{2,3}}$). During our study with P4, we saw that a whole bar consisted of almost only the note E, as indicated by identical color, except for a single note with different color – *"I also wouldn't have noticed that [without color]" (P4)*. As we only support highlighting sections and bars for now, P3 missed being able to click on single harmonies and notes to highlight them in other views.

The compressed encoding (Figure 7) was P1's favorite: *"This is the feature I'm most excited about for showing people the song structure because this is something I just can't do with a [music notation software]"* ($\mathbf{T_1}$). For playing a song with students where each plays a different instrument of the piece, it *"would be great if everyone would have something like this"* to have a compact summary of the part they should play. Our participants proposed features we could add to this view, such as reducing or disabling nested repetitions: P1 found *"it would*

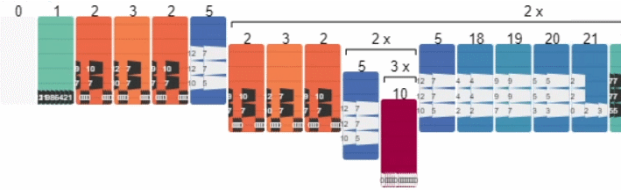


Figure 9: Our algorithm might not create the same structure as a musician: The sequence $2, 3, 2, 5$ appears twice in a row, but since the following bars of the second appearance form a longer repetition, the sequence was included there instead.

be great if you had the option to simplify it" and P2 suggested adding a slider for the compression level. They also told us that this view could be extended to support annotation: *"That would be great, to be able to go here and annotate some things" (P1)*. Our compression sometimes leads to unexpected results as it tries to find the longest repetitions, which might not be how musicians would compress a piece: *"I would expect that this is in here two times, but somehow it's here – it makes sense, it's just two different ways of describing it" (P1)* (Figure 9). P3 and P4 did not consider the compressed view useful (*"I would not use compressed much, maybe once when first looking at a new song" (P3)*) and P4 proposed merging it with the compact view by optionally drawing repeated segments only once with the brackets used in this view.

As our compact sheet music view shows the whole piece at once, it allows users to spot global patterns, such as how often a bar appears ($\mathbf{T_{1,2}}$): *"The compact view shows how things repeat" (P4)*, *"If you go here and go for [the color mode] 'Identical', you can see that there is a lot of this" (P1)*. P4 wished for an alignment, to have repeating patterns exactly below each other: *"it could auto-align it for me"*. We then asked what they think about manually inserting line breaks, whereto P4 responded: *"this would be the most important to me – if this should help me learn or read or play, I have to be able to customize and save it"*.

When asked for general feedback, P1 told us they *"like it a lot, because it's always hard to* see *what is similar to something else ... I think it's very important that you [know] not just what is played, but also if there is a connection to other segments"* ($\mathbf{T_{1,2,3}}$). P1 wished to be able to directly compare bars or sections ($\mathbf{T_3}$): *"That would be great if you could select two and then see the difference because I always click [back and forth]"*.

When asked for use cases, our participants told us they would use our interface for learning, for example by playing along. Both P3 and P4 further imagined using our compact view as a cheat sheet during a performance: *"[For songs with chords and simpler rhythm] you could print this and give it to someone ... and they could play the song ... or you use a tablet"" (P3)*. P4 hid all views but tracks and compact, thereby maximizing the latter: *"like this, I see the whole song at once ... convenient as a memory aid. Assuming I know the song already ... I see how often I have to play everything"* ($\mathbf{T_{1,2}}$). As another use case, P4 wished to be able to see multiple instruments at once to be able to compare them visually.

5. LIMITATIONS AND DISCUSSION

We mainly focus on guitar tablature, which is easier to represent compactly and often features more repeating segments than other kinds of sheet music. However, we argue that our general method of mapping similarities to color can also help with other kinds of music. With new, specialized similarity metrics and note encodings, our approach could support non-western kinds and even music without discrete notes, as long as a piece can be segmented. Our example design for guitar tablature and with a simple similarity metric allowed us to stay within a reasonable scope and matched our own musical expertise.

Human color vision is limited, even more with color vision deficiencies. Our approach can add value compared to non-colored sheet music for everyone, although accessible color scales reveal less detail. In our study, some patterns were clearly visible while some were harder to spot – still, they were easier to spot than without any color. Coloring by similarity works well for pieces with a few different segments that are repeated, as fewer colors are necessary, but will not work as well for others.

As our current approach depends on dimensionality reduction and clustering, it inherits the limitation of these techniques, such as distortion and artifacts. We chose MDS and hierarchical agglomerative clustering to preserve similarities as well as possible, but other algorithms or approaches might further reduce these limitations.

In our current interface, the participants missed being able to directly compare two selections of bars, align bars automatically or through line breaks, and assign custom colors and labels. Our evaluation only included four participants. While such a number is typical for pair analytics, real-world acceptance can only be evaluated through longitudinal field studies, where a larger number of users regularly use a product in their daily life.

6. CONCLUSION

We designed multiple methods to ease the detection of repeating structures in sheet music. Our evaluation provided a first qualitative indication of the effectiveness of our approach. Therefore, we are confident that extensions to our design can turn our work into a helpful tool for musicians.

Future work includes further similarity metrics and visual encodings better suited for different tasks, sheet music characteristics, instruments, and music genres. Adding labels and exporting them would allow musicians and teachers to save and share their results. Showing multiple instruments of a piece at once would allow comparing them, for example, to quickly see where two guitars play similar notes. We plan to let more musicians actively use our design during learning, playing, and teaching over months to test real-world usage and acceptance longitudinally.

7. ACKNOWLEDGMENTS

This work was funded by the Cyber Valley Research Fund and by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project-ID 251654672 – SFB TRR 161, project A08.

8. REFERENCES

[1] D. Fürst, M. Miller, D. A. Keim *et al.*, "Augmenting sheet music with rhythmic fingerprints," in *5th Workshop Visualization for the Digital Humanities (VIS4DH)*. IEEE, 2020, pp. 14–23.

[2] M. Miller, A. Bonnici, and M. El-Assady, "Augmenting music sheets with harmonic fingerprints," in *Proc. ACM Symp. Document Engineering (DocEng)*. ACM, 2019.

[3] M. Miller, D. Fürst, H. Hauptmann *et al.*, "Augmenting digital sheet music through visual analytics," *Computer Graphics Forum (CGF)*, vol. 41, no. 1, pp. 301–316, 2022.

[4] T. Munzner, *Visualization analysis and design*. CRC press, 2014.

[5] M. Good *et al.*, "MusicXML: An internet-friendly format for sheet music," in *XML Conf. and expo*. Citeseer, 2001, pp. 03–04.

[6] R. Valle and A. Freed, "Symbolic music similarity using neuronal periodicity and dynamic programming," in *Mathematics and Computation in Music (MCM)*, T. Collins, D. Meredith, and A. Volk, Eds. Cham: Springer, 2015, pp. 199–204.

[7] W. B. de Haas, F. Wiering, and R. C. Veltkamp, "A geometrical distance measure for determining the similarity of musical harmony," *Int. Journal of Multimedia Information Retrieval (IJMIR)*, vol. 2, no. 3, pp. 189–202, 2013.

[8] J. Urbano, J. Lloréns, J. Morato *et al.*, "Using the Shape of Music to Compute the Similarity between Symbolic Musical Pieces," in *Int. Symposium on Computer Music Modeling and Retrieval (CMMR)*, 2010, pp. 385–396.

[9] F. Simonetta, F. Carnovalini, N. Orio *et al.*, "Symbolic music similarity through a graph-based representation," in *Proc. of the Audio Mostly 2018 on Sound in Immersion and Emotion*. ACM, 2018.

[10] B. Janssen, P. van Kranenburg, and A. Volk, "A comparison of symbolic similarity measures for finding occurrences of melodic segments," in *16th Int. Society for Music Information Retrieval Conf. (ISMIR)*, 2015.

[11] V. Velardo, M. Vallati, and S. Jan, "Symbolic melodic similarity: State of the art and future challenges," *Computer Music Journal (CMJ)*, vol. 40, no. 2, pp. 70–83, 2016.

[12] R. Khulusi, J. Kusnick, C. Meinecke *et al.*, "A survey on visualizations for musical data," *Computer Graphics Forum (CGF)*, 2020.

[13] H. B. Lima, C. G. R. D. Santos, and B. S. Meiguins, "A survey of music visualization techniques," *ACM Comput. Surv.*, vol. 54, no. 7, 2021.

[14] F. Watanabe, R. Hiraga, and I. Fujishiro, "Brass: Visualizing scores for assisting music learning," in *Proc. International Computer Music Conference (ICMC)*, 2003.

[15] M. Wattenberg, "Arc Diagrams: Visualizing structure in strings," in *IEEE Symp. Information Visualization (INFOVIS)*. IEEE, 2002, pp. 110–116.

[16] J. Li, "Music analysis through visualization," in *Proc. of the Int. Conf. on Technologies for Music Notation and Representation*, 2016, pp. 220–225.

[17] G. D. Cantareira, L. G. Nonato, and F. V. Paulovich, "MoshViz: A detail+overview approach to visualize music elements," *IEEE Trans. on Multimedia*, vol. 18, no. 11, pp. 2238–2246, 2016.

[18] J. Savelsberg, "Visualizing music structure using Spotify data," in *Extended Abstracts for the Late-Breaking Demo Session of the 22nd Int. Society for Music Information Retrieval Conf. (ISMIR EA)*, 2021.

[19] M. Müller and N. Jiang, "A scape plot representation for visualizing repetitive structures of music recordings." in *13th Int. Society for Music Information Retrieval Conf. (ISMIR)*, 2012.

[20] A. Hayashi, T. Itoh, and M. Matsubara, "Colorscore – visualization and condensation of structure of classical music," in *2011 15th Int. Conf. on Information Visualisation (IV)*, 2011, pp. 420–425.

[21] T. Bergstrom, K. Karahalios, and J. C. Hart, "Isochords: Visualizing structure in music," in *Proc. Graphics Interface*. ACM, 2007, p. 297–304.

[22] J. Snydal and M. Hearst, "ImproViz: Visual explorations of Jazz improvisations," in *Extended Abstracts on Human Factors in Computing Systems (CHI EA)*, ser. CHI EA '05. ACM, 2005, p. 1805–1808.

[23] C. Bunks, T. Weyde, A. Slingsby, and J. Wood, "Visualization of tonal harmony for jazz lead sheets," in *EuroVis 2022 - Short Papers*. The Eurographics Association, 2022, pp. 109–113. [Online]. Available: https://doi.org/10.2312/evs20221102

[24] R. D. Prisco, D. Malandrino, D. Pirozzi *et al.*, "Understanding the structure of musical compositions: Is visualization an effective approach?" *Information Visualization*, vol. 16, no. 2, pp. 139–152, 2017.

[25] S. Asahi, S. Tamura, Y. Sugiyama *et al.*, "Toward a high performance piano practice support system for beginners," in *Asia-Pacific Signal and Information Processing Association Annual Summit and Conf. (APSIPA ASC)*, 2018, pp. 73–79.

[26] M. Hori, C. M. Wilk, and S. Sagayama, "Piano practice evaluation and visualization by HMM for arbitrary jumps and mistakes," in *2019 53rd Annual Conf. Information Sciences and Systems (CISS)*, 2019, pp. 1–5.

[27] W. B. De Haas, M. Rohrmeier, R. C. Veltkamp *et al.*, "Modeling harmonic similarity using a generative grammar of tonal harmony," in *Proc. of the Tenth Int. Conf. on Music Information Retrieval (ISMIR)*, 2009.

[28] W. B. De Haas, J. Rodrigues Magalhães, R. C. Veltkamp *et al.*, "HarmTrace: Improving harmonic similarity estimation using functional harmony analysis," in *Proc. of the 12th Int. Conf. on Music Information Retrieval (ISMIR)*, 2011.

[29] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," in *Soviet Physics Doklady*, vol. 10, no. 8. Soviet Union, 1966, pp. 707–710.

[30] P. Jaccard, "The distribution of the flora in the alpine zone," *New Phytologist*, vol. 11, no. 2, pp. 37–50, 1912.

[31] J. B. Kruskal, "Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis," *Psychometrika*, vol. 29, no. 1, pp. 1–27, 1964.

[32] D. Borland and R. M. Taylor Ii, "Rainbow color map (still) considered harmful," *IEEE Computer Graphics and Applications (CG&A)*, vol. 27, no. 2, pp. 14–17, 2007.

[33] K. Reda and D. A. Szafir, "Rainbows revisited: Modeling effective colormap design for graphical inference," *IEEE Trans. Visualization and Computer Graphics (TVCG)*, vol. 27, no. 2, pp. 1032–1042, 2021.

[34] K. Reda, A. A. Salvi, J. Gray *et al.*, "Color nameability predicts inference accuracy in spatial visualizations," *Computer Graphics Forum (CGF)*, vol. 40, no. 3, pp. 49–60, 2021.

[35] M. Wattenberg, F. B. Viégas, and K. Hollenbach, "Visualizing activity on Wikipedia with chromograms," in *Human-Computer Interaction – INTERACT 2007*. Springer, 2007, pp. 272–287.

[36] J. R. Nuñez, C. R. Anderton, and R. S. Renslow, "Optimizing colormaps with consideration for color vision deficiency to enable accurate interpretation of scientific data," *PLOS ONE*, vol. 13, no. 7, pp. 1–14, 2018.

[37] M. Bostock, V. Ogievetsky, and J. Heer, "D^3: Data-driven documents," *IEEE Trans. on Visualization and Computer Graphics (TVCG)*, vol. 17, no. 12, pp. 2301–2309, 2011.

[38] R. Arias-Hernandez, L. T. Kaastra, T. M. Green *et al.*, "Pair analytics: Capturing reasoning processes in collaborative visual analytics," in *44th Hawaii Int. Conf. System Sciences (HICSS)*, 2011, pp. 1–10.

PASSAGE SUMMARIZATION WITH RECURRENT MODELS FOR AUDIO – SHEET MUSIC RETRIEVAL

Luís Carvalho[1] **Gerhard Widmer**[1,2]
[1]Institute of Computational Perception & [2]LIT Artificial Intelligence Lab
Johannes Kepler University Linz, Austria
{luis.carvalho, gerhard.widmer}@jku.at

ABSTRACT

Many applications of cross-modal music retrieval are related to connecting sheet music images to audio recordings. A typical and recent approach to this is to learn, via deep neural networks, a joint embedding space that correlates short fixed-size snippets of audio and sheet music by means of an appropriate similarity structure. However, two challenges that arise out of this strategy are the requirement of strongly aligned data to train the networks, and the inherent discrepancies of musical content between audio and sheet music snippets caused by local and global tempo differences. In this paper, we address these two shortcomings by designing a cross-modal recurrent network that learns joint embeddings that can summarize longer passages of corresponding audio and sheet music. The benefits of our method are that it only requires weakly aligned audio – sheet music pairs, as well as that the recurrent network handles the non-linearities caused by tempo variations between audio and sheet music. We conduct a number of experiments on synthetic and real piano data and scores, showing that our proposed recurrent method leads to more accurate retrieval in all possible configurations.

1. INTRODUCTION

The abundance of music-related content in various digital formats, including studio and live audio recordings, scanned sheet music, and metadata, among others, calls for efficient technologies for cross-linking between documents of different modalities. In this work, we explore a cross-modal task referred to as audio – sheet music passage retrieval. We define it as follows: given an audio fragment as a query, search within an image database and retrieve the corresponding sheet music passage; or vice versa, find the appropriate recording fragment given a query in the form of some snippet of (scanned) sheet music.

A fundamental step in audio–sheet music retrieval concerns defining a suitable shared representation that permits the comparison between items of different modalities in a convenient and effective way. The conventional approaches for linking audio recordings to their respective printed scores are based on handcrafted mid-level representations [1, 2]. These are usually pitch-class profiles, like chroma-based features [3,4], symbolic fingerprints [5], or the bootleg score [6, 7], which is a coarse mid-level codification of the main note-heads in a sheet music image. However extracting such representations requires a series of pre-processing stages that are prone to errors, for example optical music recognition on the sheet music side [8–10], and automatic music transcription on the audio part [11–13].

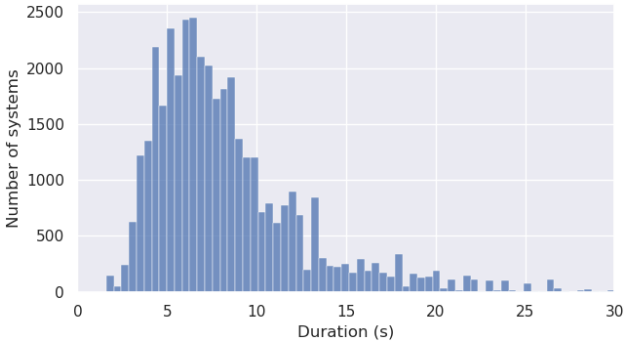


Figure 1: Distribution of system durations in around 40,000 examples from the MSMD. More than 25% of the passages are longer than ten seconds.

A promising approach [14, 15] has been proposed to eliminate these problematic pre-processing steps by learning a shared low-dimensional embedding space directly from audio recordings and printed scores. This is achieved by optimizing a cross-modal convolutional network (CNN) to project short snippets of audio and sheet music onto a latent space, in which the cosine distances between semantically related snippets are minimized, whereas non-related items of either modality are projected far from each other. Then the retrieval procedure is reduced to simple nearest-neighbour search in the shared embedding space, which is a simple and fast algorithm.

A first limitation of this strategy relates to its supervised nature: it requires strongly-aligned data in order to generate matching audio–sheet snippet pairs for training, which means fine-grained mappings between note onsets and corresponding note positions in the score. Obtaining such annotations is tedious and time-consuming, and also

© L. Carvalho and G. Widmer. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** L. Carvalho and G. Widmer, "Passage summarization with recurrent models for Audio – Sheet Music Retrieval", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

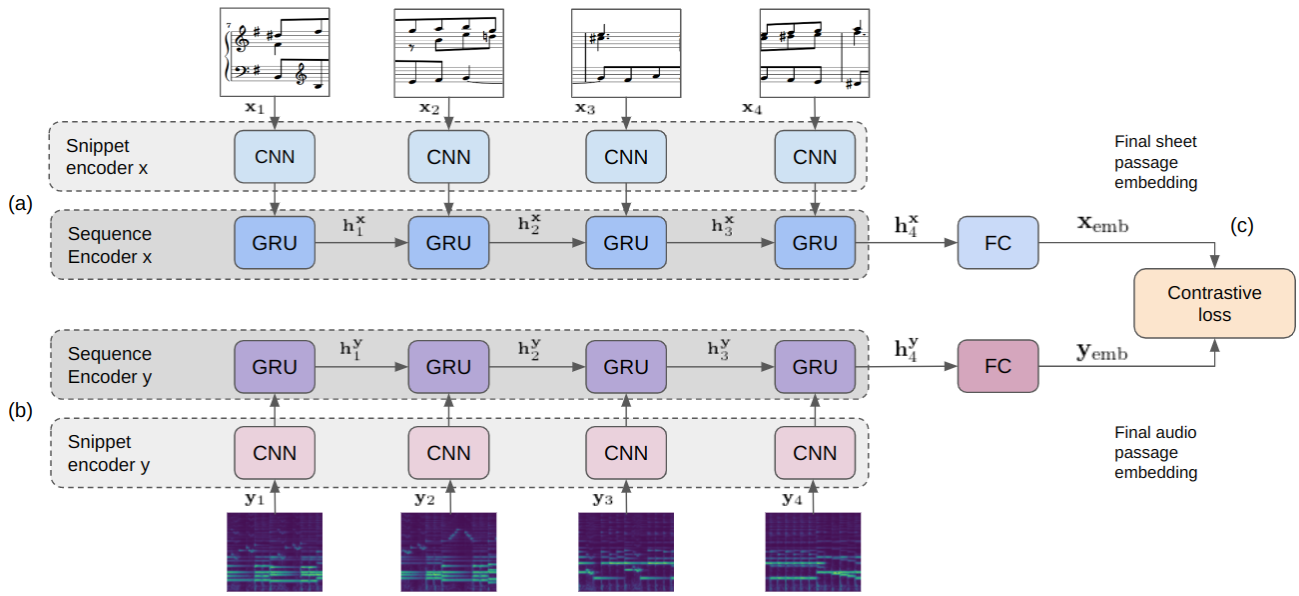


Figure 2: Diagram of the proposed network. Two independent pathways are trained to encode sheet music (a) and audio (b) passages by minimizing a contrastive loss function (c).

requires specialized annotators with musical training. As a result, embedding learning approaches have been trained with synthetic data, in which recordings, sheet music images, and their respective alignments are rendered from symbolic scores. This leads to poor generalization in scenarios with real music data, as shown in [16].

Moreover, the snippets in both modalities have to be fixed in size, meaning that the amount of actual musical content in the fragments can vary considerably depending on note durations and the tempo in which the piece is played. For example, a sheet excerpt with longer notes played slowly would correspond to a considerably larger duration in audio than one with short notes and a faster tempo. This leads to generalization problems caused by differences between what the model sees during training and test time; [17] attempted to address this limitation by introducing a soft-attention mechanism to the network.

In this paper we address the two aforementioned limitations by proposing a recurrent cross-modal network that learns compact, fixed-size representations from longer variable-length fragments of audio and sheet music. By removing the fixed-size fragment constraint, we can adjust the lengths of fragments during training so that cross-modal pairs can span the same music content, leading to a more robust representation. Moreover, by operating with longer music passages, it is possible to rely solely on weakly-annotated data for training, since we now require only the starting and ending positions of longer-context music fragments within music documents, in order to extract audio–sheet passages to prepare a train set. This is a remarkable advantage compared for example to other approaches based on [14], where fine-detailed alignments are indispensable to generate short audio–sheet snippet pairs.

The rest of the paper is structured as follows. In Section 2 we describe the model proposed to learn joint representations from cross-modal passages. Section 3 presents a series of experiments on artificial and real data and Section 4 summarizes and concludes the work.

2. AUDIO–SHEET PASSAGE RETRIEVAL

For the purposes of this paper, and in order to be able to use our annotated corpora for the experiments, we define a "passage" as the musical content corresponding to one line of sheet music (also known as a "system"). System-level annotation of scores are much easier to come by than note-precise score-recording alignments, making it relatively easy to compile large collections of training data for our approach. Our definition of passages resembles that of "musical themes", which has been used under a cross-modal retrieval scenario with symbolic queries in a number of previous works [18, 19]. To illustrate the temporal discrepancies between passages, we show in Figure 1 the distribution of time duration of the systems from all pieces of the MSMD dataset [14] (later we will elaborate more on this database). In this dataset, we observe that systems can cover from less than five to more than 25 seconds of musical audio.

This important temporal aspect motivates us to propose the network depicted in Figure 2 to learn a common latent representation from pairs of audio–sheet passages. The architecture has two independent recurrent-convolutional pathways, which are responsible for encoding sheet music (Figure 2a) and audio (Figure 2b) passages. The key component of this approach is the introduction of two recurrent layers that, inspired by traditional sequence-to-sequence models [22], are trained to summarize a variable-length sequences into context vectors, that we conveniently refer to as embedding vectors.

Defining a pair of corresponding passages in the form of image (sheet music) and log-magnitude spectro-

Audio CNN encoder input: 92×20	**Sheet-Image CNN encoder** input: 160×180
2x Conv(3, pad-1)-24 - BN	2x Conv(3, pad-1)-24 - BN
MaxPooling(2)	MaxPooling(2)
2x Conv(3, pad-1)-48 - BN	2x Conv(3, pad-1)-48 - BN
MaxPooling(2)	MaxPooling(2)
2x Conv(3, pad-1)-96 - BN	2x Conv(3, pad-1)-96 - BN
MaxPooling(2)	MaxPooling(2)
2x Conv(3, pad-1)-96 - BN	2x Conv(3, pad-1)-96 - BN
MaxPooling(2)	MaxPooling(2)
Conv(1, pad-0)-32 - BN	Conv(1, pad-0)-32 - BN
FC(32)	FC(32)

Table 1: Overview of the two convolutinal encoders. Each side is responsible for their respective modality. Conv(3, pad-1)-24: 3×3 convolution, 24 feature maps and zero-padding of 1. BN: Batch normalization [20]. We use ELU activation functions [21] after all convolutional and fully-connected layers.

gram (audio) as $\mathbf{X}$ and $\mathbf{Y}$, respectively, two sequences $(\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_N})$ and $(\mathbf{y_1}, \mathbf{y_2}, \ldots, \mathbf{y_M})$ are generated by sequentially cutting out short snippets from $\mathbf{X}$ and $\mathbf{Y}$. The shapes of the short sheet and audio snippets are respectively 160×180 (pixels) [1] and 92×20 (frequency bins $\times$ frames), which corresponds to one second of audio. After that, each individual snippet is encoded by a VGG-style CNN [23] into a 32-dimensional vector, as shown in Figure 2, generating two sequences of encoded snippets, one for the audio passage, and the other for the sheet passage (note that each modality has its own dedicated CNN encoder). The architecture of the CNN encoders are detailed in Table 1.

Then each sequence is fed to a recurrent layer in order to learn the spatial and temporal relations between subsequent snippets, which are inherent in music. After experimenting with two typical simple recurrent layers, namely long short-term memory cells (LSTM) [24] and gated recurrent units (GRU) [25], we observed on average better results with GRUs, and we decided for the latter for our architecture. Each of the two GRUs is designed with 128 hidden units, where the hidden state of each GRU after the last step is the context vector that summarizes the passages. Finally a fully connected layer (FC) is applied over each context vector, in order to encode the final passage embeddings $(\mathbf{x}_{\text{emb}}, \mathbf{y}_{\text{emb}})$ with the desired dimension.

During training, a triplet (contrastive) loss function [26] is used to minimize the distances between embeddings from corresponding passages of audio and sheet music and maximize the distance between non-corresponding ones. Defining $\mathrm{d}(\cdot)$ as the cosine distance, the loss function is given by:

$$\mathcal{L} = \sum_{k=1}^{K} \max\Big\{0, \alpha + \mathrm{d}\Big(\mathbf{x}_{\text{emb}}, \mathbf{y}_{\text{emb}}\Big) - \mathrm{d}\Big(\mathbf{x}_{\text{emb}}, \mathbf{y}^{k}_{\text{emb}}\Big)\Big\}, \tag{1}$$

where $\mathbf{y}^{k}_{\text{emb}}$ for $k \in 1, 2, \ldots, K$ are contrastive (negative) examples from K non-matching passages in the same training mini-batch. This contrastive loss is applied to all $(\mathbf{x}_{\text{emb}}, \mathbf{y}_{\text{emb}})$ pairs within each mini-batch iteration. The margin parameter $\alpha \in \mathbb{R}_{+}$, in combination with the $\max\{\cdot\}$ function, penalizes matching snippets that were poorly embedded.

For the sake of simplicity, we leave the remaining details concerning the design of the networks, such as learning hyper-parameters, to our repository where our method will be made publicly available, [2] as well as the trained models derived in this work.

3. EXPERIMENTS

In this section we conduct experiments on different audio–sheet music scenarios. We first elaborate on the main dataset used for training and evaluation and define the steps of the passage retrieval task. Then we select four experiment setups and present the results.

We train our models with the Multi-Modal Sheet Music Dataset (MSMD) [14], which is a collection of classical piano pieces with multifaceted data, including score sheets (PDF) engraved via Lilypond [3] and corresponding audio recordings rendered from MIDI with several types of piano soundfonts. With over 400 pieces from over 50 composers, including Bach, Beethoven and Schubert, and covering more than 15 hours of audio, the MSMD has audio–sheet music alignments which allow us to obtain corresponding cross-modal pairs of musical passages. From the MSMD we were able to derive roughly 5,000 audio–sheet passages for training, which is scaled up to around 40,000 different pairs after data augmentation: audios are re-rendered with different soundfonts and have their tempo changed between 90% and 110%. Then we generate a test set of 534 pairs from a separate set of music pieces, that were rendered with a soundfont that was not seen during training. Later, in 3.2, we will also consider real scanned scores and real audio recordings.

To perform cross-modal passage retrieval, we first embed all audio–sheet pairs in the shared space using our trained model depicted in Figure 2. Then the retrieval is conducted by using the cosine distance and nearest-neighbor search within the space. For example, in case of using an audio passage as a query to find the appropriate sheet music fragment, the pairwise cosine distances between the query embedding and all the sheet music passage embeddings are computed. Finally, the retrieval results are obtained by means of a ranked list through sorting the distances in ascending order.

As for evaluation metrics, we look at the *Recall@k* (R@k), *Mean Reciprocal Rank* (MRR) and the *Median Rank* (MR). The R@k measures the ratio of queries which were correctly retrieved within the top k results. The MRR is defined as the average value of the reciprocal rank over all queries. MR is the median position of the correct match in the ranked list.

[1] In our approach, all sheet music pages are initially re-scaled to a 1181×835 resolution

[2] `https://github.com/luisfvc/lcasr`

[3] `http://www.lilypond.org`

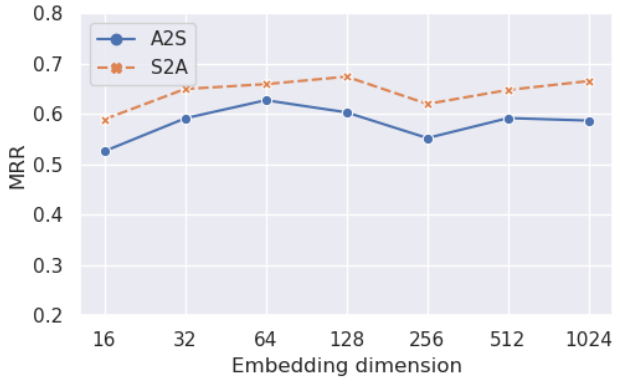


Figure 3: Mean Reciprocal Rank (MRR) for different embedding dimensions, evaluated in both search directions.

3.1 Experiment 1: Embedding dimension

In the first round of experiments, we investigate the effect of the final embedding dimension in the retrieval task. We consider the values in $\{16, 32, 64, 128, 256, 512, 1024\}$ and train the model of Figure 2 with the same hyperparameters. Then we perform the retrieval task in both search directions: audio-to-sheet music (A2S) and sheet music-to-audio (S2A).

Figure 3 presents the MRR of the snippet retrieval results evaluated on the 534 audio–sheet music passage pairs of the MSMD testset. A first and straightforward observation is that in all cases the S2A direction indicates better retrieval quality. We observe the performance increasing together with the embedding dimensionality until it stagnates at 64-D, and the MRR does not improve on average for higher-dimensional embeddings. For this reason, we select the model that generates 64-dimensional embeddings as the best one, which will be evaluated more thoroughly in the next experiments.

3.2 Experiment 2: Real data and improved models

In this section, we conduct an extensive series of experiments comparing our proposed recurrent network and some improved models thereof with baseline methods, and extend the evaluation to real-world piano data.

Given that our training data are entirely synthetic, we wish to investigate the generalization of our models from synthetic to real data. To this end, we evaluate on three datasets: on a (1) fully artificial one, and on datasets consisting (2) partially and (3) entirely of real data. For (1) we use the test split of MSMD and for (2) and (3) we combine the Zeilinger and Magaloff Corpora [27] with a collection of commercial recordings and scanned scores that we have access to. These data account for more than a thousand pages of sheet music scans with mappings to both MIDI files and over 20 hours of classical piano recordings. Then, besides the MSMD (I), we define two additional evaluation sets: (II) *RealScores_Synth*: a partially real set, with *scanned* (real) scores of around 300 pieces aligned to *synthesized* MIDI recordings. And (III) *RealScores_Rec*: an entirely real set, with *scanned* (real) scores of around 200 pieces and their corresponding *real audio* recordings.

As a baseline (BL), we implement the method from [14] and adapt their short-snippet-voting strategy to identify and retrieve entire music recordings and printed scores so it can operate with passages. [4] In essence, short snippets are sequentially cut out from a passage query and embedded, and are compared to all embedded snippets which were selected from passages in a search dataset of the counterpart modality, resulting in a ranked list based on the cosine distance for each passage snippet. Then the individual ranked lists are combined into a single ranking, in which the passage with most similar snippets is retrieved as the best match.

Additionally, we investigate whether our models can benefit from pre-trained cross-modal embeddings. Since both CNN encoders of our proposed network architecture (see Figure 2) are the same as in [14], we re-designed the baseline cross-modal network to accommodate our snippet dimensions (160×180 and 92×20, for sheet and audio, respectively) and trained a short-snippet embedding model also with the MSMD, as a pre-training step, and then loaded the two CNN encoders of our recurrent network with their respective pre-trained weights before training. Our hypothesis is that, by initializing the CNN encoders with parameters that were optimized to project short pairs of matching audio–sheet snippets close together onto a common latent space, models with better embedding capacity can be obtained. After loading the two CNNs with pre-trained weights, we can either freeze (FZ) them during training or just fine-tune (FT) on them. Therefore, in our experiments, we refer to these modifications of our proposed vanilla recurrent network (RNN) as RNN-FZ and RNN-FT, respectively.

Moreover, an additional CCA (canonical correlation analysis) layer [28] is used in [14] to increase the correlation of corresponding pairs in the embedding space. This CCA layer is refined in a post-training step, and we investigate whether this refinement process is beneficial to our network. In our experiments we refer to models that were initialized with pre-trained parameters from networks that had their CCA layer refined as RNN-FZ-CCA and RNN-FT-CCA.

Table 2 presents the results for all data configurations and models defined previously. To keep our experiments consistent and the comparison fair, we randomly select 534 passage pairs from sets (II) and (III) to create the retrieval scenario for their respective experiments.

An evident observation from the table is the considerable performance drop as we transition from synthetic to real music data. For all the models, the MRR drops at least

[4] The reasons we did not use the attention-based method from [17] as a baseline comparison are twofold. First we intend to compare the exact original snippet embedding architecture with and without a recurrent encoder, and adding the attention mechanism to a baseline model would introduce a significant number of additional trainable parameters, making the comparison unfair. Second, the purpose of the attention model is to compensate the musical content discrepancy between audio and sheet snippets, which is not the case for musical passages as defined here: pairs of audio–sheet music passages comprise the exact musical content (that is the reason why fragments are not fixed in time).

Table 2: Results of audio–sheet music passage retrieval, performed in both search directions, and evaluated in three types of data: (I) fully synthetic, (II) partially real and (III) entirely real. Boldfaced rows represent the best performing model per dataset.

	Audio-to-Score (A2S)					**Score-to-Audio (S2A)**				
	R@1	**R@10**	**R@25**	**MRR**	**MR**	**R@1**	**R@10**	**R@25**	**MRR**	**MR**
I MSMD (Fully synthetic)										
BL	47.56	81.68	90.80	0.592	1	51.37	83.51	92.59	0.628	1
RNN	51.12	84.46	92.88	0.627	1	54.30	85.95	94.94	0.670	1
RNN-FT	55.27	87.98	95.02	0.651	1	56.32	87.12	96.44	0.697	1
RNN-FT-CCA	**60.04**	**89.66**	**97.73**	**0.692**	**1**	**62.11**	**91.44**	**98.41**	**0.734**	**1**
RNN-FZ	50.76	84.20	92.11	0.619	1	52.90	85.21	94.12	0.658	1
RNN-FZ-CCA	52.67	86.46	92.88	0.635	1	55.67	86.30	95.34	0.682	1
II RealScores_Synth (Sheet music scans and synthetic recordings)										
BL	20.19	55.47	74.99	0.343	7	25.15	70.27	83.11	0.391	5
RNN	25.09	61.24	78.27	0.374	5	30.15	72.47	86.89	0.439	3
RNN-FT	28.87	66.41	81.32	0.447	4	33.98	75.47	88.51	0.462	2
RNN-FT-CCA	**33.36**	**69.49**	**83.88**	**0.481**	**3**	**37.35**	**79.22**	**89.95**	**0.538**	**1**
RNN-FZ	25.83	62.02	79.74	0.376	5	31.45	74.87	87.26	0.442	3
RNN-FZ-CCA	26.82	63.33	80.19	0.391	5	33.55	75.71	88.79	0.467	2
III RealScores_Rec (Sheet music scans and real recordings)										
BL	15.67	31.46	48.12	0.226	29	18.30	36.71	54.94	0.266	18
RNN	19.11	35.98	53.65	0.278	21	22.76	39.95	57.47	0.303	15
RNN-FT	22.39	39.53	57.19	0.338	18	26.76	42.77	59.38	0.371	7
RNN-FT-CCA	**26.62**	**44.81**	**60.01**	**0.362**	**7**	**29.84**	**46.71**	**60.88**	**0.435**	**4**
RNN-FZ	17.65	33.12	52.98	0.252	22	19.13	37.51	55.57	0.277	17
RNN-FZ-CCA	18.38	35.81	54.51	0.279	21	22.30	38.95	58.82	0.285	16

0.2 points to a partially real test set, and drops more than 0.3 points when moving to the entirely real data. Moreover, as mentioned in Subsection 3.1, the passage retrieval metrics of the S2A direction are better than those of A2S for all models and scenarios.

Our recurrent model RNN and its variants outperform the baseline approach in all retrieval scenarios for all evaluation metrics. In our findings, we did not see noticeable improvements when the pre-loaded encoders were frozen during training. In fact, for some configurations (scenarios I and III) the evaluation metrics were slightly worse than those from the vanilla RNN model. When the CNN encoders are pre-loaded and enabled for fine-tuning, we observe the largest improvements over RNN and subsequently over BL. Moreover, the models initialized with pre-trained weights from CCA-refined networks (RNN-FT-CCA) achieved the best overall results, for all test datasets and search directions.

In addition to the overall absolute improvements, we observe that the performance drop between synthetic and real datasets shrinks with our proposed models, specially with RNN-FT-CCA. In comparison with the baseline, the I-to-III MRR gap is reduced by 0.036 and 0.06 points in the directions A2S and S2A, respectively.

The results we obtained and summarized in Table 2 indicate that introducing a recurrent layer to learn longer contexts of musical content is beneficial in our cross-modal retrieval problem. However the real-data generalization problem is still evident, and in Section 4 we discuss potential solutions to address such issues.

3.3 Experiment 3: Global tempo variations

In this experiment, we investigate the robustness of our system to global tempo changes. To this end, the pieces of the MSMD test dataset are re-rendered with different tempo ratios $\rho \in \{0.5, 0.66, 1, 1.33, 2\}$ ($\rho = 0.5$ means the tempo was halved and $\rho = 2$ stands for doubling the original tempo). A similar study was conducted in [17] for retrieval of short audio–sheet snippets.

Table 3 summarizes the MRR values obtained for each tempo re-rendering, where the baseline method is compared with our proposed recurrent model. We notice the general trend that the MRR gets worse as the tempo ratio is farther from $\rho = 1$ (original tempo). This behavior is somehow expected because the new tempo renditions are more extreme than the tempo changes the model has seen during training.

Besides the better MRR values of the proposed network, an important improvement concerns the performance drop when changing from $\rho = 1$ to $\rho = 0.5$ (slower renditions). The MRR gap between these tempo ratios drops from 0.12 to 0.1 and from 0.09 to 0.07 points for the A2S and S2A directions, respectively, when comparing our net-

Model	$\rho = 0.5$	$\rho = 0.66$	$\rho = 1$	$\rho = 1.33$	$\rho = 2$
BL	0.47	0.54	0.59	0.52	0.40
RNN	0.53	0.59	0.63	0.58	0.43

(a) A2S search direction.

Model	$\rho = 0.5$	$\rho = 0.66$	$\rho = 1$	$\rho = 1.33$	$\rho = 2$
BL	0.54	0.59	0.63	0.56	0.48
RNN	0.60	0.64	0.67	0.61	0.50

(b) S2A search direction.

Table 3: MRR for different tempo renderings of the test pieces of MSMD in both (a) audio-to-sheet and (b) sheet-to-audio retrieval directions. We evaluate both baseline and RNN models.

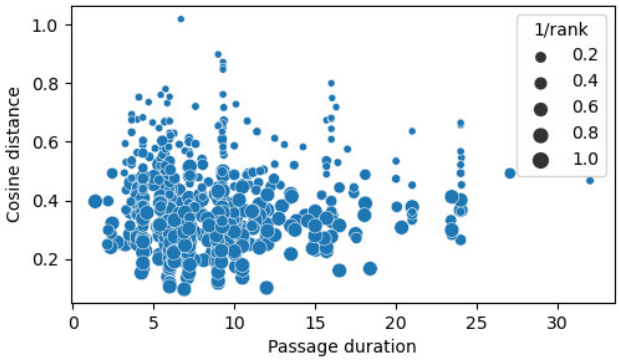


Figure 4: Cosine distance in the embedding space in relation to the respective audio passage duration of 534 pairs from the MSMD test set. The cosine distances were computed with the RNN model.

work with the baseline. This indicates that the recurrent model is more robust to global tempo variations and can operate well with longer audio passages.

3.4 Experiment 4: Qualitative analysis

To get a better understanding of the behavior of our proposed network, in this last experiment we take a closer look at the shared embedding space properties. Figure 4 shows the distribution of the pairwise cosine distances between the passage pairs from the MSMD test set, in relation to the duration (in seconds) of their respective audio passages. Moreover, we scale the point sizes in the plot so they are proportional to their individual precision values (inverse of the rank values), when considering the S2A experimental setup.

An interesting behavior in this visualization is the size of the points increasing as the cosine distance decreases. It is expected that passage pairs with smaller distances between them, meaning that they are closer together in the embedding space, would be lead to better retrieval ranks.

Another interesting aspect of this distribution concerns the proportion of larger cosine distances as the audio duration of the passages increases. For example, between five and ten seconds, there are more large points observed than smaller ones, while between 20 and 25 seconds, the proportion is roughly equal. This indicates that, in our test set, embeddings from shorter passages of audio are still located closer to their sheet counterparts in comparison with longer audio passages, despite our efforts to design a recurrent networks that learns from longer temporal contexts.

4. CONCLUSION AND FUTURE WORK

We have presented a novel cross-modal recurrent network for learning correspondences between audio and sheet music passages. Besides requiring only weakly-aligned music data for training, this approach overcomes the problems of intrinsic global and local tempo mismatches of previous works that operate on short and fixed-size fragments. Our proposed models were validated in a series of experiments under different retrieval scenarios and generated better results when comparing with baseline methods, for all possible configurations.

On the other hand, a serious generalization gap to real music data was observed, which points us to the next stages of our research. A natural step towards making deep-learning-based cross-modal audio–sheet music retrieval more robust would be to include real and diverse data that can be used for training models. However such data with suitable annotations are scarce, and recent advances in end-to-end full-page optical music recognition [29] can be a possible solution to learn correspondences on the score page level. Moreover, the powerful transformers [30] are potential architectures to learn correspondences from even longer audio recordings, accommodating typical structural differences between audio and sheet music, such as jumps and repetitions.

5. ACKNOWLEDGMENTS

This work is supported by the European Research Council (ERC) under the EU's Horizon 2020 research and innovation programme, grant agreement No. 101019375 (*Whither Music?*), and the Federal State of Upper Austria (LIT AI Lab).

6. REFERENCES

[1] M. Müller, A. Arzt, S. Balke, M. Dorfer, and G. Widmer, "Cross-modal music retrieval and applications: An overview of key methodologies," *IEEE Signal Processing Magazine*, vol. 36, no. 1, pp. 52–62, 2019.

[2] Ö. Izmirli and G. Sharma, "Bridging printed music and audio through alignment using a mid-level score representation," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Porto, Portugal, 2012, pp. 61–66.

[3] C. Fremerey, M. Clausen, S. Ewert, and M. Müller, "Sheet music-audio identification," in *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, Kobe, Japan, Oct. 2009, pp. 645–650.

[4] F. Kurth, M. Müller, C. Fremerey, Y. ha Chang, and M. Clausen, "Automated synchronization of scanned

sheet music with audio recordings," in *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, Vienna, Austria, Sep. 2007, pp. 261–266.

[5] A. Arzt, S. Böck, and G. Widmer, "Fast identification of piece and score position via symbolic fingerprinting," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Porto, Portugal, 2012, pp. 433–438.

[6] T. J. Tsai, "Towards linking the Lakh and IMSLP datasets," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2020, pp. 546–550.

[7] D. Yang, T. Tanprasert, T. Jenrungrot, M. Shan, and T. J. Tsai, "MIDI passage retrieval using cell phone pictures of sheet music," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2019, pp. 916–923.

[8] J. Calvo-Zaragoza, J. H. Jr., and A. Pacha, "Understanding optical music recognition," *ACM Computing Surveys*, vol. 53, no. 77, 2021.

[9] J. C. López-Gutiérrez, J. J. Valero-Mas, F. J. Castellanos, and J. Calvo-Zaragoza, "Data augmentation for end-to-end optical music recognition," in *Proceedings of the 14th IAPR International Workshop on Graphics Recognition (GREC)*. Springer, 2021, pp. 59–73.

[10] E. van der Wel and K. Ullrich, "Optical music recognition with convolutional sequence-to-sequence models," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Suzhou, China, 2017, pp. 731–737.

[11] 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.

[12] 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 (ISMIR)*, Paris, France, 2018, pp. 50–57.

[13] 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.

[14] M. Dorfer, J. Hajič 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*, vol. 1, no. 1, 2018.

[15] M. Dorfer, A. Arzt, and G. Widmer, "Learning audio-sheet music correspondences for score identification and offline alignment," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Suzhou, China, 2017, pp. 115–122.

[16] L. Carvalho, T. Washüttl, and G. Widmer, "Self-supervised contrastive learning for robust audio–sheet music retrieval systems," in *Proceedings of the ACM International Conference on Multimedia Systems (ACM-MMSys)*, Vancouver, Canada, 2023.

[17] S. Balke, M. Dorfer, L. Carvalho, A. Arzt, and G. Widmer, "Learning soft-attention models for tempo-invariant audio-sheet music retrieval," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Delft, Netherlands, 2019, pp. 216–222.

[18] F. Zalkow and M. Müller, "Using weakly aligned score–audio pairs to train deep chroma models for cross-modal music retrieval," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Montréal, Canada, 2020, pp. 184–191.

[19] S. Balke, V. Arifi-Müller, L. Lamprecht, and M. Müller, "Retrieving audio recordings using musical themes," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Shanghai, China, 2016, pp. 281–285.

[20] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on International Conference on Machine Learning (ICML)*, Lille, France, 2015, pp. 448–456.

[21] D. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (ELUs)," in *International Conference on Learning Representations, (ICLR)*, 2016.

[22] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proceedings of the 27th International Conference on Neural Information Processing Systems*, 2014, pp. 3104–3112.

[23] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations (ICLR)*, 2015.

[24] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[25] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha,

Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1724–1734.

[26] R. Kiros, R. Salakhutdinov, and R. S. Zemel, "Unifying visual-semantic embeddings with multimodal neural language models," *arXiv preprint (arXiv:1411.2539)*, 2014. [Online]. Available: http://arxiv.org/abs/1411.2539

[27] 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.

[28] M. Dorfer, J. Schlüter, A. Vall, F. Korzeniowski, and G. Widmer, "End-to-end cross-modality retrieval with CCA projections and pairwise ranking loss," *International Journal of Multimedia Information Retrieval*, vol. 7, no. 2, pp. 117–128, Jun 2018. [Online]. Available: https://doi.org/10.1007/s13735-018-0151-5

[29] A. Ríos-Vila, J. M. Iñesta, and J. Calvo-Zaragoza, "End-to-end full-page optical music recognition for mensural notation," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Bengaluru, India, 2022, pp. 226–232.

[30] 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*, vol. 30. Curran Associates, Inc., 2017.

PREDICTING PERFORMANCE DIFFICULTY FROM PIANO SHEET MUSIC IMAGES

Pedro Ramoneda[1] **Jose J. Valero-Mas**[1]
Dasaem Jeong[2] **Xavier Serra**[1]
[1] Music Technology Group, Universitat Pompeu Fabra, Barcelona
{pedro.ramoneda, josejavier.valero, xavier.serra}@upf.edu
[2] MALer Lab, Sogang University, Seoul
dasaemj@sogang.ac.kr

ABSTRACT

Estimating the performance difficulty of a musical score is crucial in music education for adequately designing the learning curriculum of the students. Although the Music Information Retrieval community has recently shown interest in this task, existing approaches mainly use machine-readable scores, leaving the broader case of sheet music images unaddressed. Based on previous works involving sheet music images, we use a mid-level representation, bootleg score, describing notehead positions relative to staff lines coupled with a transformer model. This architecture is adapted to our task by introducing an encoding scheme that reduces the encoded sequence length to one-eighth of the original size. In terms of evaluation, we consider five datasets—more than 7500 scores with up to 9 difficulty levels—, two of them particularly compiled for this work. The results obtained when pretraining the scheme on the IMSLP corpus and fine-tuning it on the considered datasets prove the proposal's validity, achieving the best-performing model with a balanced accuracy of 40.34% and a mean square error of 1.33. Finally, we provide access to our code, data, and models for transparency and reproducibility.

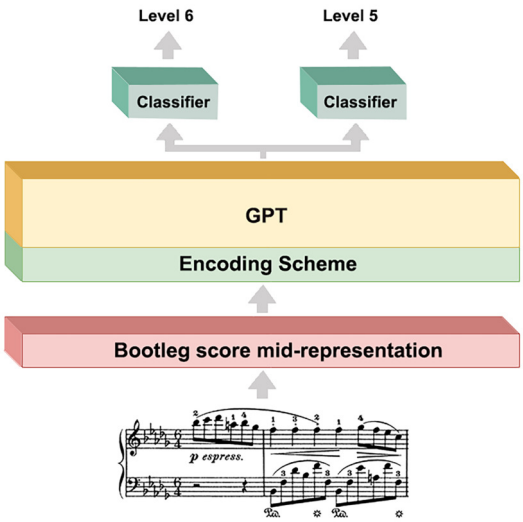


Figure 1. We consider the bootleg score mid-representation with a multi-task GPT-based recognition framework to predict the performance difficulty associated to a piano score directly from sheet images from multiple annotated collections with varied difficulty levels.

1. INTRODUCTION

Estimating the difficulty of a piece is crucial for music education, as it enables the effective structuring of music collections to attend to the student's needs. This has led to a growing research interest [1–4], as well as the development of automatic systems for exploring difficulties by major industry players such as Muse Group [5,6] and Yousician [7].

Previous research on predicting piano difficulty has primarily focused on symbolic machine-readable scores [1, 2, 4, 8–10]. Early studies explored feature engineering descriptors [1,2] and the relationship between piano fingering and difficulty [8–10]. A recent study [4] used stacked recurrent neural networks and context attention for difficulty classification on machine-readable scores, employing embeddings from automatic piano fingering, piano expressive generation [11], and score information. This study found that modeling the score difficulty classification task as an ordinal regression problem [12] was advantageous, and using entire pieces for training, rather than fragments, was essential to avoid degraded performance.

Although symbolic machine-readable scores offer more interpretability [10], with all the music information completely accessible, their limited availability compared to sheet music images restricts the practical use of difficulty prediction tools for librarians, teachers, and students. Focusing on sheet music image analysis expands the range of available music, has the potential to preserve the cultural heritage of symbolic-untranscribed scores, and addresses the lack of diversity in Western classical piano curricula. By analyzing image-based sheet music, we aim

© P. Ramoneda, J. J. Valero-Mas, D. Jeong and X. Serra. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** P. Ramoneda, J. J. Valero-Mas, D. Jeong and X. Serra, "Predicting performance difficulty from piano sheet music images", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.*, Milan, Italy, 2023.

to create technology for highlighting historically underrepresented communities like female composers [13, 14] and promoting diversity in piano education. This promotion is crucial since the piano teaching repertoire has remained mostly unchanged for decades [15], containing around 3,300 pieces [16], while projects such as IMSLP house remarkably larger databases.

One of the main challenges in working with sheet music is attaining a symbolic music-based representation for direct analysis. Although Optical Music Recognition (OMR) literature has considerably improved in creating such representations over the past 30 years, it remains an unsolved task [17]. Bootleg score [18] is an alternative to symbolic scores obtained with OMR. This mid-level symbolic representation keeps the most relevant primitives of the music content in a music sheet, which has shown remarkable success in several tasks [19–22], especially in classification, such as piano composer classification [19, 23, 24] or instrument recognition [25].

We build on this literature, employing the GPT model [26] and bootleg score in our analysis. More precisely, we consider the approach by Tsai et al. [18], in which a GPT model pretrained on the IMSLP piano collection is finetuned for specific recognition tasks. With adequate adaptations, we hypothesize that this framework may also succeed in estimating performance difficulty on music sheet images.

As aforementioned, difficulty estimation benefits from the use of entire music pieces rather than excerpts to obtain adequate success rates. However, processing large sequence stands as a remarkable challenge in music processing, especially when addressing bootleg representations due its considerable verbosity. While some recent mechanisms address this issue in general learning frameworks (*e.g.*, Flash Attention [27]), we extend the original proposal by Tsai et al. [18] with a multi-hot optimization target for GPT pretraining, and replace the categorical encoding with causal convolutional or feedforward projection layers to enhance performance and reduce costs.

Moreover, addressing data scarcity is crucial for promoting and establishing this task within the Music Information Retrieval community. As of now, the *Mikrokosmos-difficulty* (MK) [10] and *Can I Play It?* (CIPI) [4] symbolic datasets stand for the only available annotated collections, out of which music sheet images can be obtained by engraving mechanisms. To enhance data availability and encourage further research, we have collected additional datasets from existing collections, namely *Pianostreet-difficulty* (PS), *Freescore-difficulty* (FS), and black female composers collection Hidden Voices (HV). This results in more than 7500 music pieces, spanning up to 9 difficulty levels and each annotated with a difficulty classification system. Although difficulty prediction contains a subjective element, global trends may emerge when examining multiple difficulty classification systems simultaneously. To our knowledge, no previous research has explored this aspect. Consequently, we propose a multitask approach to training simultaneously on CIPI, PS, and FS datasets. Finally, we also analyze the generalization of our proposed methodologies with the MK and HV benchmark datasets.

Considering all above, our precise contributions are: (i) we adopt the previous bootleg-representation literature [23, 24], pretraining a GPT model on IMSLP and finetuning it for our task, adapting the encoding scheme accordingly, as presented in Figure 1; (ii) we evaluate our proposal using a novel sheet music image collection of five datasets with more than 7,500 pieces with difficulty levels ranging up to 9; (iii) we propose a multi-task strategy for combining multiple difficulty classification systems from the datasets; (iv) we conduct extensive experiments to assess the proposed methodologies, including a zero-shot scenario for testing generalization and comparisons with previous proposals on the CIPI dataset; and (v) to promote the task, code, and models [1], and datasets [2] are publicly available.

2. MUSIC SHEET IMAGE DATASETS

Due to the relative recentness of the field, the lack of annotated corpora has severely constrained the performance difficulty assessment. The earliest data assortments may be found in the works by Sebastian et al. [1] and Chiu et al. [2], which respectively collected 50 and 300 MIDI scores from different score repositories. However, these datasets were never publicly released.

To our best knowledge, the *Mikrokosmos difficulty* (MK) set by Ramoneda et al. [10], which comprises 147 piano pieces by Béla Bartók in a symbolic format graded by the actual composer, represents the first publicly available collection for the task at hand. More recently, the authors introduced the *Can I Play It?* (CIPI) dataset [4], a collection of 652 piano works in different symbolic formats annotated after 9 different difficulty levels. Note that, while sheet music scores can be obtained by resorting to engraving mechanisms, the insights obtained may not apply to real-world scenarios.

Dataset	Pieces	Classes	AIR	Noteheads	Composers
MK [10]	147	147	.78	49.2k	1
CIPI [4]	652	9	.33	1.1M	29
PS	2816	9	.24	7.2M	92
FS	4193	5	.37	5.8M	747
HV	17	4	1	21.5k	10

Table 1. Description of existing collections for performance difficulty estimation based on the number of pieces, classes, average imbalance ratio (AIR), noteheads, and composers. The dashed line differentiates the datasets based on symbolic (above) and image (below) sheet music.

To address this limitation, we compiled a set of real sheet music images of piano works together with their performance difficulty annotations from different music education and score-sharing platforms on the Internet. More

[1] https://github.com/PRamoneda/pdf-difficulty
[2] https://zenodo.com/record/8126801

precisely, we arranged three different collections attending to the source: (i) the *Pianostreet-difficulty* (PS) set retrieved from [28] that depicts 2,816 works with 9 difficulty levels annotated by the Pianostreet team; (ii) the *Freescores-difficulty* (FS) assortment from [29] that contains 4,193 pieces with 5 difficulty levels comprising a variety of compositions and annotations by the users of the platform; and (iii) the *Hidden Voices* (HV) collection [30, 31], a set of 17 pieces by black female composers annotated with 4-level difficulty labels by musicologists of the Colorado Boulder Music Department.

Table 1 summarizes the main characteristics of commented publicly-available collections. The *average imbalance ratio* (AIR), measured as the mean of the individual ratios between each difficulty class and the majority label in each collection, is also provided for reference purposes.

3. METHODOLOGY

Based on its success when addressing classification tasks from sheet music images [23, 25], our proposal considers the use of the so-called bootleg score representation coupled with a GPT-based recognition model to estimate the performance difficulty of a piece.

Introduced by [18], bootleg scores stand as a simple—yet effective—representation to encode the content of a sheet music image for certain recognition tasks. Formally, a bootleg score is a binary matrix of length w and $h = 62$ vertical positions—*i.e.*, $\mathcal{X} \in \{0,1\}^{w \times 62}$—that respectively denote the temporal and pitch dimensions. Note that the w value represents the number of note heads detected by the bootleg extraction process. Our work resorts to this representation, being the use of alternative codifications posed as a future line to address.

The GPT recognition framework undergoes an unsupervised pretraining step on the IMSLP piano collection, which was originally used by [18]. Eventually, considering a set of labeled data $\mathcal{T} \subset \mathcal{X} \times \mathcal{C}$ where $\mathcal{C} = \{c_1, \ldots, c_{|\mathcal{C}|}\}$ denotes the possible difficulty levels, the model is finetuned to retrieve the recognition function $\hat{f} : \mathcal{X} \rightarrow \mathcal{C}$ that relates a bootleg representation to a particular difficulty level. Based on previous work addressing this task [4], we consider an ordinal classification framework [12] as the difficulty grading scales naturally fit this formulation.

Despite being capable of addressing the task, the framework was noticeably affected by two factors: (i) the excessive length of the input sequences when pretraining the model; and (ii) the inconsistent definition of difficulty levels among corpora. Consequently, we introduce two mechanisms specifically devised to address these limitations.

3.1 Sequence length in pretraining

One of the main drawbacks related to bootleg representations is their verbosity, as it depicts $h = 62$ elements per frame. To address this issue, Tsai et al. [23] proposed subdividing each column into groups of 8 elements and encoding each according to a vocabulary of $|\sigma| = 2^8$ elements. In this regard, the initial bootleg score $x \in \{0,1\}^{w \times 62}$ is mapped to a novel space defined as $\Sigma^{w \times 8}$. This representation is then flattened to undergo a categorical embedding process that maps it to a feature-based space denoted as $\mathbb{R}^{8w \times 768}$, which is eventually used for pretraining the GPT model with 768-dim hidden states. Note that this process reduces the vocabulary size and remarkably increases the sequence length.

To address this issue, we propose substituting this tokenization process with an embedding layer that directly maps the bootleg score into a suitable representation, avoiding the extension of the initial length of the sequence. In this sense, the initial bootleg representation $x \in \{0,1\}^{w \times 62}$ is mapped to a space defined as $\mathbb{R}^{w \times 768}$ that serves as input to the GPT model with a fraction of the length of the encoding used by Tsai et al. [23]. Besides reducing the length of the sequences to process, we hypothesize that such an embedding may benefit the recognition model as a suitable representation is inferred for the task. In this regard, our experiments will compare two types of embedding approaches—more precisely, a fully-connected layer and a convolutional one, respectively denoted as FC and CNN—to quantitatively assess this claim.

Figure 2 graphically describes the approach by Tsai et al. [23] and the presented proposal. In opposition to the reference work, the proposal considers multi-hot encoding instead of discrete categorical index as the output of the GPT recognition framework, by using binary cross-entropy loss instead of negative log-likelihood loss.

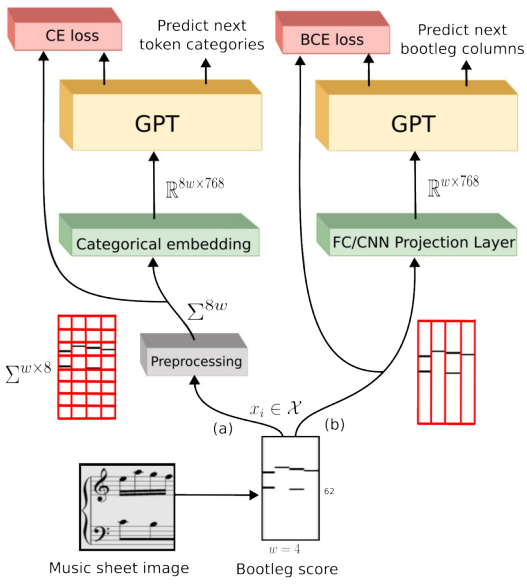


Figure 2. Comparison between the proposal by Tsai et al. [23]—denoted as (a)—and the presented proposal—highlighted as (b)—for a case of toy example with a duration of $w = 4$.

3.2 Multi-task learning of multiple difficulty classification systems

The pretrained GPT model can be simply finetuned for a performance difficulty classification task by adding a projection layer and a learnable classification token, as de-

picted in Figure 3. However, the actual definition of the performance difficulty of a piece is a highly subjective problem that may bias—and, hence, remarkably hinder—the goodness of a recognition model. In this regard, we hypothesize that using a multi-task approach that attends different definitions of difficulty—*i.e.*, a labeled assortment of data from multiple annotators—may benefit the generalization capabilities of the approach.

In this regard, we modify the reference architecture for the downstream task to include an additional classification layer for each training collection. While simple, such a proposal is expected to improve the overall recognition performance given the wider variety of data provided during the training process. Figure 3 graphically describes this proposal.

Finally, no pre-processing is done in relation to the label distribution of the corpora to avoid inducing any type of bias. In this regard, the sampling protocol of the model has been forced to maintain its original distributions.

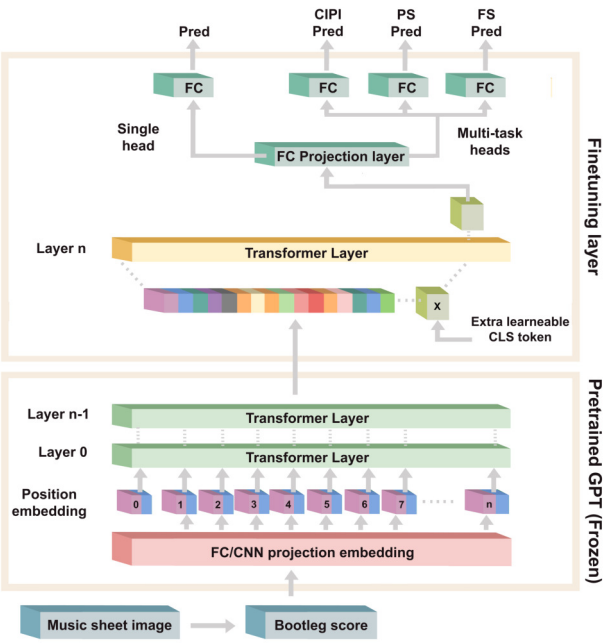


Figure 3. Graphical description of the downstream architecture depicting the classification heads for the multi-task proposals as well as the single-head case of the reference work.

4. EXPERIMENTAL SETUP

4.1 Data collections and assessment metrics

To validate the proposal, we have considered the five publicly-available data collections presented in Section 2, *i.e.*, *Mikrokosmos difficulty* (MK) [10], *Can I Play It?* (CIPI) [4], *Pianostreet-difficulty* (PS) [28], *Freescores-difficulty* (FS) [29], and *Hidden Voices* (HV) [30, 31]. While MK and CIPI exclusively comprise symbolic scores, we engraved them into music sheets and included them due to the commented scarcity of annotated data.

We considered a 5-fold cross-validation scheme with a data partitioning of 60% for the finetuning phase after the pretraining stage with IMSLP together with two equal-size splits of the remaining data for validation and testing. Note that, since MK and HV are exclusively used for benchmark purposes, no partitioning is applied to them.

In terms of performance evaluation, we resort to two assessment criteria typically used in ordinal classification [32]: *accuracy within* n (Acc_n) and *mean squared error* (MSE). To adequately described them, let $\mathcal{S} \subset \mathcal{X} \times C$ denote a set of test data and let $\mathcal{S}_c = \{(x_i, y_i) \in \mathcal{S} : y_i = c\}$ with $1 \leq i \leq |\mathcal{S}|$ be the subset of elements in $\mathcal{S}$ with class c.

Based on this, Acc_n is defined as:

$$\text{Acc}_n = \frac{1}{|\mathcal{C}|} \sum_{\forall c \in \mathcal{C}} \frac{\left|\left\{ y \in \mathcal{S}_c : \left|\hat{f}(x) - c\right| \leq n \right\}\right|}{|\mathcal{S}_c|} \quad (1)$$

where $\hat{f}(\cdot)$ represents the trained recognition model and $n \in \mathbb{N}_0$ denotes the tolerance or class-boundary relaxation that allows for errors in adjacent labels. In our experiments we consider the values of $n = 0$ (no tolerance) and $n = 1$ (smallest adjacency tolerance), respectively denoted as Acc_0 and Acc_1 in the rest of the work.

Regarding MSE, this figure of merit is defined as:

$$\text{MSE} = \frac{1}{|\mathcal{C}|} \sum_{\forall c \in \mathcal{C}} \frac{\sum_{\forall x \in \mathcal{S}_c} \left(\hat{f}(x) - c\right)^2}{|\mathcal{S}_c|} \quad (2)$$

Finally, note that all these metrics are macro-averaged to account for the unbalanced nature of the data collections used in the work.

4.2 Training procedure

As commented, the recognition model undergoes an initial pretraining stage considering the IMSLP corpus. During this stage, the model considers sequences of 256 tokens, each with a binary cross-entropy as a loss function. To speed up this process, the Flash Attention framework by [27] is also considered. For comparative purposes, all other parameters remain unaltered from the reference works [23].

After that, the model is finetuned on the downstream difficulty estimation task, considering an Adam optimizer [33] with a learning rate of 10^{-5} and early stopping based on the Acc_0 and MSE metrics on the validation set. Moreover, a balanced sampler is considered to tackle the issue of unbalanced data collections. Ordinal Loss [12] is applied to train the difficulty prediction as an ordinal classification problem, while no loss weighting considered in the multi-task framework. For regularization and stable training, gradient clipping is set to 10^{-4}, with a batch size of 64 and L2 regularization. This optimization process is carried out exclusively on the last layer of the model, resorting the remaining parts to the weights obtained during the pretraining phase of the procedure.

Note that while these processes may be further studied to account for the optimal solution that retrieves the best-performing results, such a study is out of the scope of the work and is left as future work to address.

5. EXPERIMENTS AND RESULTS

This section presents the results obtained with the introduced experimental scheme. To adequately provide insights about the task, the section provides a series of individual experiments devoted to analyzing one aspect of the proposal: Section 5.1 analyzes the influence of the encoding scheme; Section 5.2 evaluates the influence of the multitask architecture; Section 5.3 delves on the ranking generalization in a zero-shot scenario; finally, Section 5.4 compares the attainable results when addressing the task from the symbolic versus the sheet-image domains.

5.1 Encoding schemes experiment

This first experiment compares the performance of the two encoding schemes presented in Section 3.1, *i.e.*, GPT_{FC} and GPT_{CNN}. Table 2 presents the results obtained for the CIPI, FS, and PS collections for the three figures of merit considered.

Encoding	Acc_0 (%)	Acc_1 (%)	MSE
Can I Play it?			
GPT_{FC}	34.3(6.1)	78.1(4.6)	1.6(0.3)
GPT_{CNN}	**36.2(8.2)**	**81.7(1.5)**	**1.4(0.1)**
PianoStreet			
GPT_{FC}	30.9(3.8)	71.1(9.6)	2.1(0.4)
GPT_{CNN}	**31.8(1.6)**	**78.8(1.8)**	**1.9(0.1)**
FreeScores			
GPT_{FC}	46.6(1.9)	**92.5(1.0)**	**0.8(0.1)**
GPT_{CNN}	**47.3(3.4)**	92.4(0.6)	0.8(0.1)

Table 2. Results of comparing the encoding schemes GPT_{FC} and GPT_{CNN}. Bold values highlight the best results per collection and metric.

As it may be observed, the GPT_{CNN} experiment outperformed the GPT_{FC} experiment in most evaluation metrics across the three datasets. More precisely, the GPT_{CNN} consistently achieved the best performance in the Acc_0 metric for all data collections, showing an average improvement of 1% concerning the GPT_{CNN} case. This trend remains for the rest of the figures of merit except for the case in the FS assortment, in which the results of the FC-based model outperform those of the CNN case.

Nevertheless, attending to the high standard deviations, the performance results of the two models show a remarkable overlap in performance, hence suggesting that both schemes are equally capable of performing the posed task of score difficulty analysis from sheet music images. In this regard, further work should explore other encoding alternatives to assess whether this performance stagnation is due to the representation capabilities of the considered embedding layers or due to the recognition framework.

5.2 Multi-task learning experiment

In this second study, we assess the capabilities of the multi-task framework proposed in Section 3.2 trained simultaneously on the CIPI, PS, and FS datasets for the two $\text{GPT}_{\text{FC}}^{\text{multi}}$ and $\text{GPT}_{\text{CNN}}^{\text{multi}}$ encoding schemes. Table 3 provides the results obtained.

Encoding	Acc_0 (%)	Acc_1 (%)	MSE
$\text{GPT}_{\text{FC}}^{\text{multi}}$			
CIPI	**40.3(4.3)**	**82.0(1.4)**	**1.3(0.1)**
PS	35.9(3.1)	**78.2(3.4)**	**1.9(0.2)**
FS	45.8(2.5)	92.0(1.4)	**0.8(0.1)**
$\text{GPT}_{\text{CNN}}^{\text{multi}}$			
CIPI	34.9(5.0)	81.4(1.3)	1.4(0.1)
PS	**35.9(2.8)**	74.5(3.4)	2.7(0.2)
FS	**45.9(1.2)**	**92.4(2.1)**	0.8(0.1)

Table 3. Results of multi-task learning experiment when evaluated on different test collections for the two encoding schemes. Bold values highlight the best results per collection and metric.

Overall, the $\text{GPT}_{\text{FC}}^{\text{multi}}$ method had higher results than the $\text{GPT}_{\text{CNN}}^{\text{multi}}$ method on the CIPI and PS datasets, especially on Acc_0 and Acc_1. For CIPI, $\text{GPT}_{\text{FC}}^{\text{multi}}$ surpassed $\text{GPT}_{\text{CNN}}^{\text{multi}}$ with gains of 5.4% in Acc_0, 0.6% in Acc_1, and 0.1 in MSE. For PS, $\text{GPT}_{\text{FC}}^{\text{multi}}$ slightly exceeded $\text{GPT}_{\text{CNN}}^{\text{multi}}$ with a 3.7% improvement in Acc_1 and a 0.6-point reduction in MSE, while Acc_0 was nearly equal for both methods, although $\text{GPT}_{\text{CNN}}^{\text{multi}}$ had a smaller standard deviation. Both methods displayed similar performance on the FS dataset, with less than a 1% difference across all metrics. As a result, subsequent experiments will reference the $\text{GPT}_{\text{FC}}^{\text{multi}}$ model.

The comparison between Tables 2 and 3 shows a trend change with better results performed with the FC version of the models. The other major difference is the relative improvement between the $\text{GPT}_{\text{FC}}^{\text{multi}}$ method and the best previous model GPT_{CNN} in the CIPI and slightly in the PS dataset. In contrast, the FS dataset results remain comparable. In CIPI, Acc_0 is 11.3% higher in $\text{GPT}_{\text{FC}}^{\text{multi}}$, and in PS, there is a relative improvement of 12.8%. For CIPI, Acc_1 sees a minor increase of 0.4%. MSE exhibits a small improvement of 3.6% for CIPI and 0.5% for PS. Possible reasons include label quality differences—CIPI annotated by a musicology team, PS labels provided by the platform, and FS crowdsourced by users—or the impact of dataset sizes—CIPI being the smallest and FS the largest.

5.3 Ranking generalization experiment

In this experiment, we assess the ranking capabilities of the proposal in a zero-shot setting by utilizing the embeddings of the projection layer of the model (check Figure 3). We reduce the 768-dimensional embeddings to a single dimension using Principal Component Analysis (PCA) and employ the resulting values to rank the target pieces.

Table 4 shows the results obtained resorting to the

Kendall rank correlation coefficient, τ_c, for all data collections discussed in the experiment, considering both the single-task and multi-task frameworks posed. Note that MK and HV are only used for benchmarking purposes.

Train	Evaluation				
	CIPI	PS	FS	MK	HV
CIPI	.67 (.01)	.56 (.02)	.56 (.01)	.67 (.05)	.50 (.05)
PS	.67 (.01)	.58 (.02)	.56 (.01)	.68 (.01)	.43 (.04)
FS	.64 (.04)	.55 (.01)	.56 (.02)	**.71 (.02)**	**.56 (.07)**
MULTI	**.68 (.02)**	**.59 (.02)**	**.56 (.01)**	.63 (.02)	.51 (.07)

Table 4. Zero-shot ranking results. Bold values denote the best-performing result on each evaluation dataset.

In the three training datasets, the multi-task architecture $\text{GPT}_{\text{FC}}^{\text{multi}}$ achieves the best performance with CIPI ($\tau_c = 0.68$), PS ($\tau_c = 0.59$), and FS ($\tau_c = 0.56$). Unexpectedly, the FS method outperforms others in the datasets of the MK ($\tau_c = 0.61$) and HV ($\tau_c = 0.56$). This outcome may suggest that simultaneous training on all three datasets could limit generalizability. Alternatively, the presence of license-free pieces composed after 1900 in the FS dataset, which users have uploaded, might explain the difference.

The HV dataset displays notably lower generalizability, possibly due to the smaller number of pieces, resulting in higher standard deviations. Potential bias similar to MK could also arise from the predominance of pre-20th-century data in CIPI and PS. These factors might affect the zero-shot experiment's performance. However, we must also acknowledge that most composers used for training are white males, and the HV results are significantly worse than the rest of the datasets. Therefore, future research should investigate and minimize the potential gender gap in difficulty prediction tasks.

5.4 Comparison with previous approaches

This last experiment compares the goodness of the proposed methodology in sheet music scores against other image-based approaches and with the symbolic-oriented methods domain. Regarding sheet image methods, we consider the reference method by Tsai et al. [23] based on bootleg mid-representation, denoted as GPT_{EMB}. Concerning the symbolic baseline, we reproduce the approach in [4] that proposes to describe the symbolic score in terms of piano fingering information, expressive annotations, and pitch descriptors to feed a recurrent model based on Gated Recurrent Units with attention layers (referred to as GRU+Att). Table 5 provides the results obtained. For comparative purposes, we only consider the CIPI dataset as the reference symbolic work accounted for that collection.

Examining the experiments, the $\text{GPT}_{\text{FC}}^{\text{multi}}$ model may be observed to outperform the other cases in the Acc_0 figure of merit. However, for the rest of the metrics, the reference symbolic case—denoted as GRU+Att—outperforms all image-oriented recognition models. Such a fact suggests that, while a bootleg score somehow suits this difficulty estimation task, a performance gap between this representation and pure symbolic notation needs to be addressed.

Case		Acc_0 (%)	Acc_1 (%)	MSE
Symbolic [4]				
	GRU+Att	39.5(3.4)	**87.3(2.2)**	**1.1(0.2)**
Tsat et al. [23]				
	GPT_{EMB}	19.7(4.0)	58.1(7.2)	3.3(0.8)
Proposal				
	GPT_{FC}	34.3(6.1)	78.1(4.6)	1.6(0.3)
	GPT_{CNN}	36.2(8.2)	81.7(1.5)	1.4(0.1)
	$\text{GPT}_{\text{FC}}^{\text{multi}}$	**40.3(4.3)**	82.0(1.4)	1.3(0.1)

Table 5. Performance results for the symbolic [4] and Tsai et al. [23] methods as well as the proposed approach for the CIPI dataset. Bold values highlight the best result per figure of merit.

Finally, the GPT_{EMB} model achieves the lowest performance of all alternatives, with remarkably lower accuracy rates than our proposal. Note that such a fact emphasizes the relevance of our work as a more suitable approach for performing difficulty estimation in sheet music images.

6. CONCLUSIONS

Estimating the performance difficulty of a music piece is a crucial need in music education to structure the learning curriculum of the students adequately. This task has recently gathered attention in the Music Information Retrieval field, given the scarce existing research works devoted to symbolic machine-readable scores. However, due to the limited availability of this type of data, there is a need to devise methods capable of addressing this task with image-based sheet music.

Attending to its success in related classification tasks, this work considers the use of a mid-level representation—namely, bootleg score—that encodes the content of a sheet music image with a GPT-based recognition framework for predicting the difficulty of the piece. Instead of directly applying this methodology, we propose using specific embedding mechanisms and multi-task learning to reduce the task complexity and improve its recognition capabilities. The results obtained with five different data collections—three of them specifically compiled for this work—prove the validity of the proposal as it yields recognition rates comparable to those attained in symbolic machine-readable scores.

Further work comprises assessing and proposing alternative representations to the bootleg scores (*e.g.*, solutions based on Optical Music Recognition). Also, we consider that using smaller training sequences using hierarchical attention models or weak labels for varying-length piece fragments may report benefits in the process. Finally, the practical deployment of this proposal in real-world scenarios involving real users may report some additional insights about the validity of the proposal.

7. ACKNOWLEDGMENT

We want to thank T.J. Tsai and all his students, especially Daniel Yang, for having conducted the prior research on the bootleg score and, above all, for sharing all their work in the interest of Open Science. We are also grateful to Pedro D'Avila for bringing to our attention the work of Alejandro Cremaschi related to the Hidden Voices project. Lastly, we thank Alejandro Cremaschi and the University of Colorado Boulder Libraries team, David M. Hays and Jessica Quah, for providing us with the scores.

This work is funded by the Spanish Ministerio de Ciencia, Innovación y Universidades (MCIU) and the Agencia Estatal de Investigación (AEI) within the Musical AI Project – PID2019-111403GB-I00/AEI/10.13039/501100011033 and the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Korea Government (MSIT) (NRF-2022R1F1A1074566).

8. REFERENCES

[1] V. Sébastien, H. Ralambondrainy, O. Sébastien, and N. Conruyt, "Score analyzer: Automatically determining scores difficulty level for instrumental e-learning," in *Proceedings of 13th International Society for Music Information Retrieval Conference, ISMIR*, Porto, Portugal, 2012.

[2] S.-C. Chiu and M.-S. Chen, "A study on difficulty level recognition of piano sheet music," in *Proceedings of the IEEE International Symposium on Multimedia*. IEEE, 2012, pp. 17–23.

[3] E. Nakamura and K. Yoshii, "Statistical piano reduction controlling performance difficulty," *APSIPA Transactions on Signal and Information Processing*, vol. 7, 2018.

[4] P. Ramoneda, D. Jeong, V. Eremenko, N. C. Tamer, M. Miron, and X. Serra, "Combining piano performance dimensions for score difficulty classification," *arXiv preprint arXiv:2306.08480*, 2023.

[5] "Musescore have automatic difficulty categories from year 2022," https://musescore.com/, accessed on April 11, 2023.

[6] "Ultimate guitar have automatic difficulty categories from year 2022," https://www.ultimate-guitar.com/, accessed on April 11, 2023.

[7] "System for estimating user's skill in playing a music instrument and determining virtual exercises thereof," Patent US9 767 705B1, 2017.

[8] E. Nakamura, N. Ono, and S. Sagayama, "Merged-output hmm for piano fingering of both hands." in *Proceedings of the 13th International Society for Music Information Retrieval Conference, ISMIR*, Taipei, Taiwan, 2014, pp. 531–536.

[9] E. Nakamura and S. Sagayama, "Automatic piano reduction from ensemble scores based on merged-output hidden markov model," in *Proceedings of the 41st International Computer Music Conference, ICMC*, Denton, USA, 2015.

[10] P. Ramoneda, N. C. Tamer, V. Eremenko, M. Miron, and X. Serra, "Score difficulty analysis for piano performance education," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, Singapore, Singapore, 2022, pp. 201–205.

[11] 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*, 2019, pp. 908–915.

[12] J. Cheng, Z. Wang, and G. Pollastri, "A neural network approach to ordinal regression," in *Proceedings of the IEEE International Joint Conference on Neural Networks, IJCNN*. Hong Kong, China: IEEE, 2008, pp. 1279–1284.

[13] D. Bennett, S. Macarthur, C. Hope, T. Goh, and S. Hennekam, "Creating a career as a woman composer: Implications for music in higher education," *British Journal of Music Education*, vol. 35, no. 3, pp. 237–253, 2018.

[14] J. Halstead, *The woman composer: Creativity and the gendered politics of musical composition*. Routledge, 2017.

[15] R. Cutietta, "Content for music teacher education in this century," *Arts Education Policy Review*, vol. 108, no. 6, pp. 11–18, 2007.

[16] J. Magrath, *Pianists guide to standard teaching and performance literature*. Alfred Music, 1995.

[17] J. Calvo-Zaragoza, J. H. Jr, and A. Pacha, "Understanding optical music recognition," *ACM Computing Surveys (CSUR)*, vol. 53, no. 4, pp. 1–35, 2020.

[18] D. Yang, T. Tanprasert, T. Jenrungrot, M. Shan, and T. Tsai, "MIDI passage retrieval using cell phone pictures of sheet music," in *Proceedings of the 20th International Society for Music Information Retrieval Conference, ISMIR*, Delft, The Netherlands, 2019, pp. 916–923.

[19] D. Yang, A. Goutam, K. Ji, and T. J. Tsai, "Large-scale multimodal piano music identification using marketplace fingerprinting," *Algorithms*, vol. 15, no. 5, p. 146, 2022.

[20] D. Yang, K. Ji, and T. Tsai, "Aligning unsynchronized part recordings to a full mix using iterative subtractive alignment," in *Proceedings of the 22nd International Society for Music Information Retrieval Conference, ISMIR*, Online, 2021, pp. 810–817.

[21] K. Ji, D. Yang, and T. Tsai, "Piano sheet music identification using marketplace fingerprinting," in *Proceedings of the 22nd International Society for Music Information Retrieval Conference, ISMIR*, Online, 2021, pp. 326–333.

[22] D. Yang and T. J. Tsai, "Piano sheet music identification using dynamic n-gram fingerprinting," *Transactions of the International Society for Music Information Retrieval*, vol. 4, no. 1, pp. 42–51, 2021.

[23] T. Tsai and K. Ji, "Composer style classification of piano sheet music images using language model pretraining," in *Proceedings of the 21th International Society for Music Information Retrieval Conference, ISMIR*, Montreal, Canada, 2020, pp. 176–183.

[24] D. Yang and T. Tsai, "Composer classification with cross-modal transfer learning and musically-informed augmentation," in *Proceedings of the 22nd International Society for Music Information Retrieval Conference, ISMIR*, Online, 2021, pp. 802–809.

[25] K. Ji, D. Yang, and T. J. Tsai, "Instrument classification of solo sheet music images," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, Toronto, ON, Canada, 2021, pp. 546–550.

[26] 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.

[27] T. Dao, D. Y. Fu, S. Ermon, A. Rudra, and C. Ré, "FlashAttention: Fast and memory-efficient exact attention with IO-awareness," in *Advances in Neural Information Processing Systems*, 2022.

[28] "Piano street," https://www.pianostreet.com/, accessed on April 11, 2023.

[29] "Free-scores," https://www.free-scores.com/, accessed on April 11, 2023.

[30] University of Colorado, "Hidden voices project," https://www.colorado.edu/project/hidden-voices/, accessed on April 11, 2023.

[31] H. Walker-Hill, *Piano Music by Black Women Composers: A Catalog of Solo and Ensemble Works*, ser. Music Reference Collection. Greenwood Press, 1992.

[32] L. Gaudette and N. Japkowicz, "Evaluation methods for ordinal classification," in *Proceedings of the 22nd Canadian Conference on Advances in Artificial Intelligence*, Kelowna, Canada, 2009, pp. 207–210.

[33] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

SELF-REFINING OF PSEUDO LABELS FOR MUSIC SOURCE SEPARATION WITH NOISY LABELED DATA

*Junghyun Koo[1] *Yunkee Chae[2] Chang-Bin Jeon[1] Kyogu Lee[1,2,3]
[1]Department of Intelligence and Information, [2]Interdisciplinary Program in Artificial Intelligence,
[3]Artificial Intelligence Institute, Seoul National University
{dg22302, yunkimo95, vinyne, kglee}@snu.ac.kr

ABSTRACT

Music source separation (MSS) faces challenges due to the limited availability of correctly-labeled individual instrument tracks. With the push to acquire larger datasets to improve MSS performance, the inevitability of encountering mislabeled individual instrument tracks becomes a significant challenge to address. This paper introduces an automated technique for refining the labels in a partially mislabeled dataset. Our proposed self-refining technique, employed with a noisy-labeled dataset, results in only a 1% accuracy degradation in multi-label instrument recognition compared to a classifier trained on a clean-labeled dataset. The study demonstrates the importance of refining noisy-labeled data in MSS model training and shows that utilizing the refined dataset leads to comparable results derived from a clean-labeled dataset. Notably, upon only access to a noisy dataset, MSS models trained on a self-refined dataset even outperform those trained on a dataset refined with a classifier trained on clean labels.

1. INTRODUCTION

Music source separation (MSS) is a critical task in the field of music information retrieval (MIR), with applications ranging from remixing [1–3] to transcription [4–6] and music education [7, 8]. To train high-performing MSS models, it is essential to have clean single-stem music recordings for guidance, which serve as the ground truth for model training. However, obtaining clean, large-scale datasets of single instrument tracks remains a challenging task.

With the increasing availability of music data on the internet, platforms such as YouTube provide a vast pool of potential single-instrument tracks. Although these sources offer an opportunity for performance gains through larger training datasets, collecting single instrument tracks from such platforms inevitably leads to encountering tracks with incorrect labels. For example, a query aimed at obtaining drum recordings might yield results that contain other types of instruments or noise, causing discrepancies between the expected and actual content of the collected recordings.

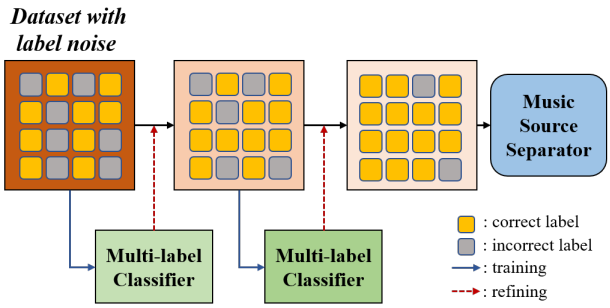


Figure 1. Overview of self-refining procedure on a noisy-labeled dataset for music source separation.

Label noise in datasets can arise from various factors, such as bleeding between instrument tracks, mislabeling due to human error, or the ambiguous timbre of instruments that resemble other instrument categories [9]. These factors make it challenging to assign a single definitive instrument label to a given recording. Such label noise is detrimental to the performance of MSS models, and there is a pressing need for an approach that can effectively train MSS models using partially corrupted datasets.

In response to this challenge, we propose an automated approach for refining mislabeled instrument tracks in a partially noisy-labeled dataset. Our *self-refining* technique, which leverages noisy-labeled data, results in only a 1% accuracy degradation for multi-label instrument recognition compared to a classifier trained with a clean-labeled dataset. The study highlights the importance of refining noisy-labeled data for training MSS models and demonstrates that utilizing the refined dataset for MSS yields results comparable to those obtained using a clean-labeled dataset. Notably, when only a noisy dataset is available, MSS models trained on self-refined datasets even outperform those trained on datasets refined with a classifier trained on clean labels. This paper presents a comprehensive analysis of our proposed method and its impact on the performance of MSS models.

*Equal contribution

© J. Koo, Y. Chae, C.-B. Jeon, and K. Lee. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).
Attribution: J. Koo, Y. Chae, C.-B. Jeon, and K. Lee, "Self-refining of Pseudo Labels for Music Source Separation with Noisy Labeled Data", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.*, Milan, Italy, 2023.

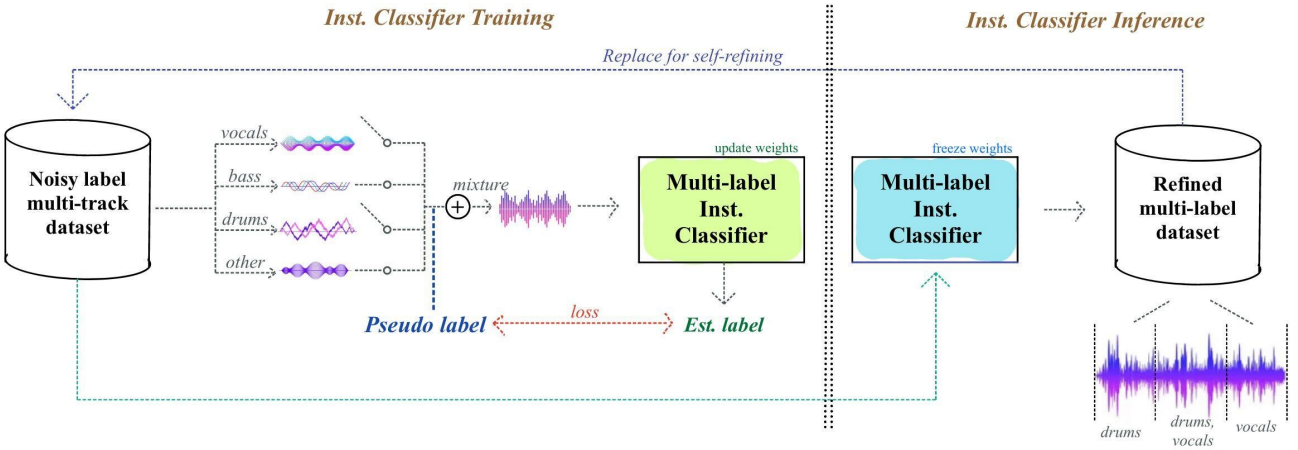


Figure 2. Overall training procedure of the Instrument Classifier Ψ. The classifier is trained to perform instrument recognition with mixtures that are synthesized by randomly selecting each stem from the noisy labeled dataset. After this training procedure, we refine the original noisy dataset and then use this new dataset to train the final Ψ.

2. RELATED WORKS

Self-training of machine learning models has been studied in various literatures, where a teacher model is first trained with clean labeled data and is used as a label predictor of unlabeled data, then a student model is trained with clean and pseudo-labeled data [10, 11]. Recently, Xie et al. proposed a noisy student method for self-training [12], which uses an iterative training of teacher-student models and noise injection methods for training student models. Thanks to their usefulness, these self-training methods have been used in diverse MIR tasks, such as singing voice detection [13, 14] and vocal melody extraction [15].

Instrument recognition or classification has been researched in various literatures, both in single-instrument [16–19] or multi-source settings [20–27]. Although such research has been focused on single or predominant-label prediction, Zhong, et al. [28] recently proposed the hierarchical approach for multi-label music instrument classification.

Our self-refining method for training of instrument classifier shares similar attributes with noisy student training [12] and the previous multi-label instrument classification [28] but differs from some perspectives. *i)* We train all our models only with partially noisy-labeled data, without access to clean-labeled data. *ii)* We train the classifiers for direct prediction of labels used in standard music source separation, e.g., vocals, bass, drums, and others, instead of the hierarchical approach. *iii)* We train multi-label classifiers with mixtures of randomly selected instruments, which are based on the characteristic of musical audio. If there exist two different instruments in one audio signal, that can be classified into two instruments. This random mixing of different instrumental tracks has been used in music source separation as well [29]. Note that the mixup method [30], which is also a mixing method of two different images, also shares a similar attribute with our method but is used for regularization of training single-label classifiers, not like our multi-label classifiers.

3. METHODOLOGY

Given a real-world scenario where the available multi-track dataset for MSS is partially incorrect with its instrument labels, a possible naive approach is first to rectify mislabeled tracks and then train an MSS model using stems with the revised labels. In this section, we introduce an effective training technique that first performs instrument recognition by only utilizing data with noisy labels and then leverages the refined dataset inferred with the trained multi-label instrument classifier to train the MSS model. With this two-stage approach, we explore the impact of the refined noisy datasets on the performance of MSS models.

3.1 Multi-label Instrument Recognition

Figure 2 summarizes the proposed training procedure of the Instrument Classifier Ψ. Similar yet different from self-training, our approach learns directly from noisy labeled data and re-labels the training data to train the final Ψ using this refined dataset. We call this training procedure *self-refining*, and this is possible by *random mixing*, a method to synthesize a mixture of multiple instruments with pseudo labels. The *random mixing* technique takes advantage of the acoustic music domain in that mixing sources of different instrument tracks still leads to natural output mixture, whereas naively combining different images in the image domain is likely to produce unrealistic results. We further discuss about the benefits and the detailed process of *random mixing* at 3.1.1.

The network architecture of Ψ is that of the ConvNeXt model [31], where it has shown great performance on a multi-instrument retrieval in [27]. The input of the network is a stereo-channeled magnitude linear spectrogram. Followed by a sigmoid layer, the model outputs four labels indicating the presence of each stem. The objective function for instrument recognition $\mathcal{L}_\Psi$ is a mean absolute loss between the estimated and synthesized pseudo labels. Preliminary experiments showed no significant difference in performance when employing mean absolute loss as compared to binary cross-entropy loss. This is likely due to

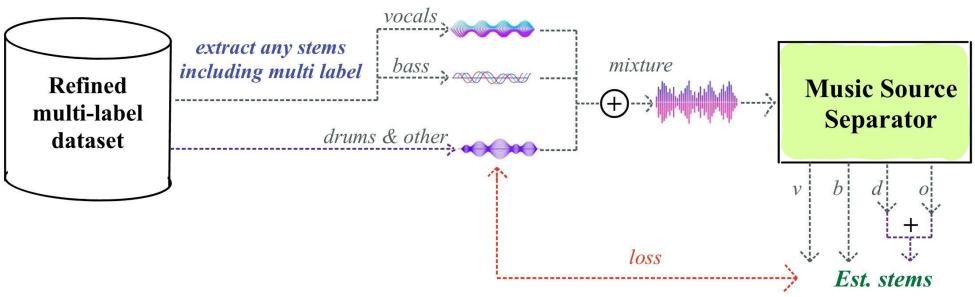


Figure 3. Music source separation training. Similar to the training procedure of the instrument classifier, we randomly mix each stem from the refined dataset to synthesize a mixture and use it as a network input. When a multi-labeled segment is selected for synthesis, the corresponding estimated stems are summed for loss computation.

the random mixing sampling that ensures similar occurrences of positive and negative labels of each instrument class during the training procedure.

3.1.1 Random Mixing

Randomly mixing stems with label noise not only creates various combinations of multi-labeled mixtures for training the instrument classifier but also brings the chance to generate a correct pseudo label from mislabeled stems. For instance, if we randomly select one correctly labeled drum track and a track that contains both sources of drums and vocals but is mislabeled as vocals, the mixing process synthesizes a correctly labeled mixture. Thanks to these fortunate chances, the random mixing technique assists the instrument classifier's accuracy in refining the label noise dataset by utilizing mislabeled stems.

To synthesize a random mixture and its pseudo label, each stem is first selected with a chance rate from the noisy dataset. The audio effects manipulation is then applied to each chosen track by simulating the music mixing process for data augmentation [3]. The order of applying audio effects with random parameters is 1. *dynamic range compression*, 2. *algorithmic reverberation*, 3. *stereo imaging*, and 4. *loudness manipulation*. Labels corresponding to randomly selected stems are used as a multi-label objective for the instrument classifier.

3.2 Music Source Separation

In this section, we describe the training procedure of MSS model employing a multi-labeled refined dataset curated by the classifier trained in Section 3.1. The majority of MSS research has focused on estimating each of the four instrument groups (*vocals*, *bass*, *drums*, and *other*) [32–35]. However, our refined dataset contains sources labeled with multiple stems, which are unsuitable for use as distinct target instruments. To utilize multi-labeled sources, we propose an appropriate MSS training framework tailored to our refined dataset.

First, we determine whether to include the multi-stem source for each input mixture sample by considering the probability p. If we decide not to include the multi-labeled source, we can train the MSS model in a conventional manner, computing the losses for each stem. Otherwise, we select a multi-labeled source from the refined dataset. Subsequently, we choose the remaining stems that do not correspond to the selected multi-labeled source from a pool of single-labeled sources and combine them to simulate a mixture. For example, when selecting a multi-labeled source *bass+drums*, we opt for single sources labeled as *vocals* and *others* to synthesize the mixture. After conducting inference with the MSS model, we add the estimated stems corresponding to the multi-stem source of the input mixture and assess the loss between them. Figure 3 illustrates our training procedure when a multi-labeled source is selected. We compute the losses for each stem, treating the multi-labeled source as an individual stem, and subsequently sum these losses to derive the final loss value.

4. EXPERIMENTS

4.1 Dataset

We use the label noise dataset provided by the Music Demixing Challenge 2023 (MDX2023) [36], which consists of 203 songs, licensed by *Moises.AI* [1]. Similar to MUSDB18 [37], the provided dataset contains mixtures of music recordings segregated into four different instrumental stems: *vocals*, *bass*, *drums*, and *other*. Each stem and its corresponding label are intentionally altered to produce a corrupted dataset to simulate mislabeling such as bleeding or human mistakes. That is, for instance, `drums.wav` may contain drum sounds and singing voices simultaneously, which is likely to be caused by bleeding. For another example, a kick-drum sound might be mislabeled as `bass.wav` when the pitch of the kick drum is melodic enough to trick a human labeler. Due to the nature of the MDX2023 challenge, the dataset does not contain the actual ground truth labels. Hence, we use all 203 songs of the MDX2023 dataset only as training data.

To validate our system trained with noisy labeled data, we employed the MUSDB18 [37] as the clean dataset for comparison and evaluation. MUSDB18 comprises 150 songs, with 100 songs for the training and 50 songs for the test set. We adopt the test subset for evaluating all systems, while the training subset is used to train the upper bound system for observation.

Data preprocessing. To prevent models from mislabels caused by silence, we remove all silent sections through-

[1] https://moises.ai/

Label Type	Training Data	Accuracy / F1 Score Precision / Recall				
		vocals	*bass*	*drums*	*other*	*avg*
Single-Label	*clean*	97.8% / 0.947 0.91 / 0.98	94.4% / 0.891 0.84 / 0.94	95.1% / 0.914 0.85 / 0.98	93.2% / 0.880 0.90 / 0.85	95.1% / 0.906 0.87 / 0.93
	noisy	93.6% / 0.860 0.76 / 0.97	**90.0% / 0.821** **0.73** / 0.93	**93.7% / 0.893** **0.81 / 0.98**	**92.6% / 0.865** **0.92** / 0.81	92.5% / 0.860 **0.80** / 0.92
	refined	**96.1% / 0.911** **0.84 / 0.98**	89.6% / 0.818 0.71 / **0.96**	93.1% / 0.884 0.79 / **0.98**	92.3% / 0.862 0.90 / **0.82**	**92.8% / 0.866** **0.80 / 0.93**
Multi-Label	*clean*	92.4% / 0.929 0.92 / 0.93	89.6% / 0.905 0.89 / 0.92	90.5% / 0.913 0.87 / 0.95	88.1% / 0.878 0.90 / 0.85	90.2% / 0.907 0.90 / 0.91
	noisy	87.9% / 0.895 0.83 / 0.96	87.5% / 0.888 **0.86** / 0.93	87.7% / 0.891 0.82 / **0.96**	87.3% / 0.872 **0.88 / 0.87**	87.6% / 0.887 0.85 / 0.93
	refined	**91.9% / 0.928** **0.88 / 0.97**	**87.8% / 0.894** 0.84 / **0.95**	**89.6% / 0.906** **0.85 / 0.96**	**87.4% / 0.874** **0.88 / 0.87**	**89.2% / 0.901** **0.86 / 0.94**

Table 1. Instrument recognition performance on single and multi-label instrument classifiers trained with different datasets. The training data of *clean*, *noisy*, and *refined* each represents the training subset of MUSDB18, MDX2023, and MDX2023 refined with the instrument classifier trained with MDX2023 Ψ_{noisy}, respectively.

out both datasets. The preprocessing procedure for silence removal is as follows:

1. For each song, detect silent areas that are below 30 dB relative to the maximum peak amplitude.
2. Remove all detected areas then merge them into one single long audio track.
3. Repeat 1. (with the threshold of 60 dB) and 2. based on the merged audio track, in case of stems that are almost silent.

After trimming silent regions, the total durations for each stem in the respective order of *vocals*, *bass*, *drums*, and *other* are [7.2, 7.8, 9.2, 10.3] hours for the MDX2023 dataset, and [2.2, 2.7, 2.9, 3.3] hours for the test subset of MUSDB18. Note that for evaluating MSS performance, we instead follow the original convention of processing entire songs from the test subset without any silence removal. We use the original audio specifications of both datasets where all audio tracks are stereo-channeled and have a sampling rate of 44.1 KHz.

4.2 Experimental Setups

For multi-label instrument recognition, the network architecture of Ψ is ConvNeXt's tiny version [31], which consists of 27.8M parameters. We feed the network with stereo-channeled mixtures of instruments that are of 2.97 seconds, which are transformed into a time-frequency domain linear magnitude spectrogram with an FFT size of 2048 and a hop size of 512. We train all Ψ for 100 epochs. During inference, Ψ performs classification by processing the entire input audio in windows of a size equivalent to the network input size, with a hop size of one-fourth of this window size. The output labels from these windows are then averaged to yield the final decision, based on a threshold value of 0.9. We utilized this inference procedure to refine the noisy dataset, which was then used to train our MSS models. Our final version of the instrument classifier trained on the refined dataset $\Psi_{refined}$ only uses stems inferred as a single-labeled for better performance based on our preliminary experiments.

We employed two MSS models, Hybrid Demucs (Demucs v3) [38] and CrossNet-Open-Unmix (X-UMX) [39], to evaluate their performance when trained on the processed datasets. Multi-labeled sources were selected with a probability of 0.4, and input loudness normalization (-14 LUFS) was applied for both training and inference in accordance with [40]. `pyloudnorm` [41] was used for loudness calculation [42].

For Demucs, the input duration was set to 3 seconds, and optimization was performed using Adam optimizer [43] and L1 loss on the time domain. The model was trained for 21,000 iterations with a batch size of 160.

For X-UMX, the input duration was set to 6 seconds, and optimization was performed using AdamW optimizer [44] and mean squared error loss on the time-frequency domain. For the sake of simplicity, we omit the multi-domain and combination loss proposed in [39]. The model was trained for 56,400 iterations with a batch size of 32. For the + *finetune w/ multi-labeled* model in Table 3, we first train the model with only single-labeled data for 20,680 iterations, then finetune it with multi-labeled data for another 35,720 iterations.

5. RESULTS

5.1 Instrument Recognition

Table 1 presents the instrument recognition performance of the multi-instrument classifier on single-labeled and multi-labeled data. As ground-truth labels are not available for the MDX2023 dataset, we validate the classification performance according to single and multi-labeled data with the MUSDB18 test set for evaluation. For the multi-label evaluation, we synthesized 3,941 mixtures from the test set with the random mixing technique described in 3.1.1. We observe the performance of Ψ trained with MUSDB18 (*clean*), MDX2023 (*noisy*), and MDX2023 once refined

Network	Training Data	SDR [dB]				
		vocals	*bass*	*drums*	*other*	*avg*
Demucs [38]	*clean*	5.92	6.16	5.58	4.43	5.52
	noisy	3.37	1.92	0.70	0.86	1.71
	w/ Ψ_{clean}	5.31	**5.12**	1.32	2.16	3.48
	w/ Ψ_{noisy}	4.15	4.58	1.62	2.85	3.30
	w/ $\Psi_{refined}$	**5.36**	5.04	**3.09**	**3.13**	**4.16**
X-UMX [39]	*clean*	5.76	4.44	5.47	3.65	4.83
	noisy	3.39	1.78	1.52	0.96	1.91
	w/ Ψ_{clean}	4.50	3.22	3.66	2.73	3.53
	w/ Ψ_{noisy}	4.72	**4.11**	3.22	2.89	3.74
	w/ $\Psi_{refined}$	**4.99**	3.93	**5.00**	**3.18**	**4.28**

Table 2. Source separation performance of Demucs v3 [38] and CrossNet-Open-Unmix [39] trained on different training datasets. Sub-items below *noisy* dataset indicate data refined with the respective instrument classifiers, denoted as $\Psi_{\bullet}$.

with Ψ_{noisy} (*refined*). The evaluation metrics used are accuracy, F1 score, precision, and recall for each instrument class and the overall averaged result.

For single-labeled data, the classifier achieves the highest average performance on the *clean* dataset, with an accuracy of 95.1% and an F1 score of 0.906. As *clean* dataset does not contain any noisy labels, the obtained results can be considered an upper bound for the performances of the classifiers. The Ψ trained on *refined* dataset results in slightly lower performance, with an accuracy of 92.8% and F1 score of 0.866, while the *noisy* dataset shows an accuracy of 92.5% and F1 score of 0.860. Although the accuracy, F1 score, and precision are higher for the *noisy* dataset in the *bass, drums*, and *other* stems, the performance metrics for *vocals* and recall values across all stems exhibit superior results when trained with the *refined* dataset.

For instrument recognition on multi-labeled data, Ψ trained on *clean* dataset yields an average accuracy of 90.2% and F1 score of 0.907. The *noisy* dataset results in an accuracy of 87.6% and an F1 score of 0.887. The *refined* dataset achieves superior performance, with an accuracy of 89.2% and an F1 score of 0.901, which is comparable to the results obtained from the *clean* dataset. Contrary to the evaluation with single-labeled data, the *refined* dataset generally demonstrates superior performance across all metrics in comparison to the *noisy* dataset. Notably, the recall values are observed to be even higher than those of the *clean* dataset. An in-depth analysis of the multi-instrument classifier results, alongside the performance outcomes of the MSS models, is discussed in Section 5.2.

5.2 Source Separation

The results of MSS models trained on different training datasets are presented in Table 2. In our evaluation, we used Signal-to-Distortion Ratio (SDR) [45], which is calculated using the `museval` toolkit [46]. For all MSS experiments, we report the SDR median of frames and the median of tracks. The Demucs and X-UMX models are trained on *clean*, *noisy*, and data processed with multi-instrument classifiers, denoted by $\Psi_{\bullet}$. In this context, $\Psi_{\bullet}$ represents the classifier trained on each respective dataset, as described in Section 5.1.

Method	SDR [dB]				
	vocals	*bass*	*drums*	*other*	*avg*
proposed	4.99	3.93	5.00	3.18	4.28
threshold = 0.5	5.06	4.13	4.77	3.06	4.25
adaptive thresholds	4.70	3.72	3.70	2.62	3.68
train only w/ single-labeled	4.90	3.73	4.54	3.18	4.09
+ finetune w/ multi-labeled	4.33	4.33	4.19	3.14	4.00
self-refining $\times 5$	4.65	3.87	5.07	2.89	4.12

Table 3. Ablation studies on MSS performances with CrossNet-Open-Unmix.

The baseline for this experiment is established using MSS models trained on the *noisy* dataset. It is noteworthy that all the results presented in the table exceed the baseline performance. For the dataset processed with the multi-instrument classifier $\Psi_{refined}$, average SDR improvements of 2.45 and 2.31 are observed for Demucs and X-UMX models, respectively, in comparison to the *noisy* dataset. Specifically, in $\Psi_{refined}$ case, both Demucs and X-UMX models demonstrate substantial improvements in SDR values across all stems compared to those of Ψ_{noisy}, with the exception of *bass* in the X-UMX model.

5.2.1 Analysis in relation to instrument recognition

In Table 2, it is noteworthy that the performance of $\Psi_{refined}$ exceeds the performance of Ψ_{clean}, even though Ψ_{clean} is trained with a noise-free labeled dataset. This implies the classification performance of Ψ_{clean} is inferior to the classification performance of $\Psi_{refined}$. This discrepancy could be attributed to differences in the data distribution between the MUSDB18 and MDX2023 datasets. Moreover, the number of training samples varies, with 100 samples in the MUSDB18 dataset and 203 samples in the MDX2023 dataset. When refining a partially noisy dataset, employing the same partially noisy dataset can yield advantageous outcomes than using the smaller clean dataset. This observation might be aligned with the findings in [12], which report an improvement in performance when a larger quantity of unlabeled data is present.

An additional factor to consider is the distinctive nature of the MSS model training framework in our approach. MSS models utilize the output of the classifier as input. The performance of the MSS model can be affected differently depending on the type of error in the classifier's output. For example, assume that the MSS model receives a sample misclassified as a vocal stem when no vocals are actually present (i.e. a false-positive sample for vocals). In this case, the MSS model simply needs to predict silence for the vocals stem and produce it as output, resulting in no significant confusion. Conversely, consider a scenario in which the MSS model receives a sample misclassified as a non-vocal stem (e.g. drums + bass), despite the presence of vocals, resulting in a false-negative sample for vocals. In such a case, the model will attempt to allocate the vocals

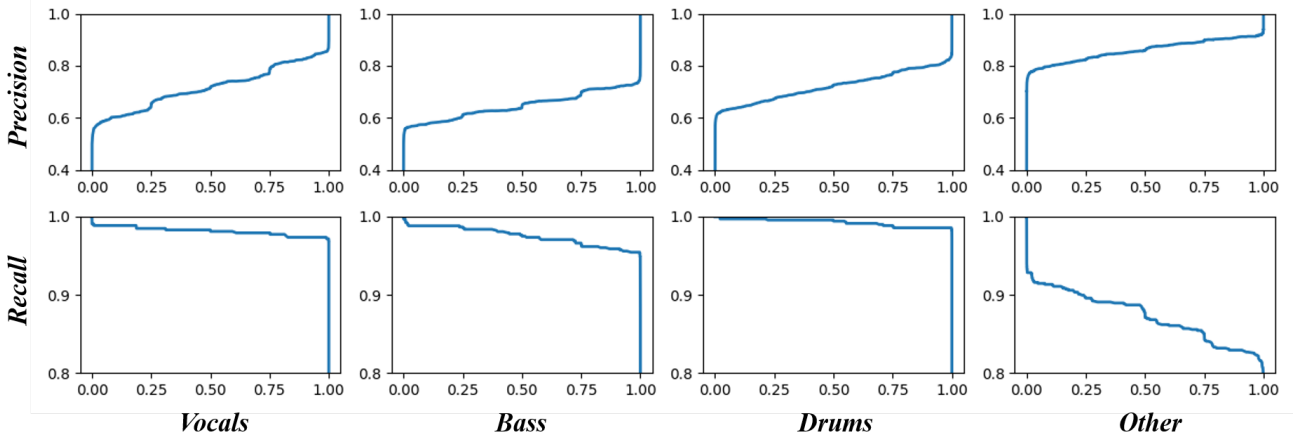


Figure 4. Precision and recall curves of the proposed classifier across different thresholds (x-axis) on each instrument. The curves are generated using the MUSDB18 test set (*clean*).

present in the input data to the drum and bass stems. Furthermore, our model differs from traditional MSS training methods as it also accepts multi-stem data as input. In this context, the vocals are present as the correct answer for multiple mislabeled non-vocal stems, which confuses the model. This not only negatively affects the performance of the mislabeled stems but also the vocal stem itself.

As a consequence of the unique characteristics of our training process, false-negative samples have a more significant impact on MSS compared to false-positive samples, highlighting the increased significance of the recall metric. Considering this perspective, the results presented in Table 1 imply the possibility of the sub-optimal performance of MSS trained on outputs of Ψ_{clean}, where the recall values are lower for all stems compared to $\Psi_{refined}$.

5.2.2 *Ablation studies*

As shown in Table 3, we evaluate the performance of X-UMX under various conditions to better understand the significance of distinct aspects of our proposed method.

Threshold. We conduct experiments to examine the impact of threshold determination for the classifier during the training of MSS models using a classified dataset. The evaluation is performed on the MUSDB18 test set. We observe that reducing the threshold to 0.5 only exhibits an SDR of 0.03 degradation compared to the original threshold value of 0.9. This outcome can be attributed to the fact that only 8% of $\Psi_{refined}$ outputs fall within the range of [0.1, 0.9] upon inference on the MUSDB18 test set. In Figure 4, we present the precision and recall curves for each threshold on individual instruments. It is evident from the curves that the variations within that range for both precision and recall are not substantial. Consequently, the choice between thresholds of 0.9 or 0.5 does not yield any noticeable disparity. Furthermore, we conduct an experiment involving adaptive thresholds for each instrument, where the threshold for each instrument was set to maximize the F1 score of the classification performance. However, we observe a significant degradation in performance across all instruments when employing adaptive thresholds. Maximizing the F1 score necessitates a trade-off between recall and precision, often leading to a decline in recall to enhance precision. Consequently, the performance of the MSS model experience degradation, aligning with the discussion presented in Section 5.2.1.

Training with multi-labeled data. When training solely with the data estimated as single-labeled, the performance is not as good as that of the proposed method. Incorporating both single- and multi-labeled data for fine-tuning after the initial training on single-labeled data leads to a slightly diminished performance, despite utilizing both types of labeled data during the training process.

Iterative self-refining. Finally, we examine the influence of the iterative self-refining technique on MSS performance. The results indicate that an MSS model trained with a noisy-labeled dataset refined five times through our method does not yield superior performance compared to the proposed model, trained on a dataset refined twice, and the performance difference is insignificant. This observation suggests that excessive refinement iterations do not necessarily lead to improved performance and that refining the dataset twice may be sufficient for optimal results.

6. CONCLUSION

In conclusion, this paper presented a self-refining approach to address the challenges of noisy-labeled data in training music source separation (MSS) models. Our proposed method refines mislabeled instrument tracks in partially noisy-labeled datasets, resulting in only a 1% accuracy degradation for multi-label instrument recognition compared to a classifier trained on a clean-labeled dataset. This study highlights the importance of refining noisy-labeled data for training MSS models effectively and demonstrates that utilizing the refined dataset for MSS yields results comparable to those obtained using a clean-labeled dataset. Considering the real-world scenario of accessibility only to a noisy dataset, MSS models trained on self-refined datasets outperformed those trained on datasets refined with a classifier trained on clean labels. The self-refining approach we introduced offers a promising direction for future research in the field of music information retrieval and has the potential to be extended to other applications requiring robust training on noisy-labeled datasets.

7. ACKNOWLEDGEMENTS

This work was partially supported by Culture, Sports and Tourism R&D Program through the Korea Creative Content Agency grant funded by the Ministry of Culture, Sports and Tourism in 2022 [No.R2022020066, 90%], and Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) [NO.2021-0-01343, Artificial Intelligence Graduate School Program (Seoul National University), 10%].

8. REFERENCES

[1] J. F. Woodruff, B. Pardo, and R. B. Dannenberg, "Remixing stereo music with score-informed source separation." in *ISMIR*, 2006, pp. 314–319.

[2] J. Pons, J. Janer, T. Rode, and W. Nogueira, "Remixing music using source separation algorithms to improve the musical experience of cochlear implant users," *The Journal of the Acoustical Society of America*, vol. 140, no. 6, pp. 4338–4349, 2016.

[3] 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," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[4] M. D. Plumbley, S. A. Abdallah, J. P. Bello, M. E. Davies, G. Monti, and M. B. Sandler, "Automatic music transcription and audio source separation," *Cybernetics &Systems*, vol. 33, no. 6, pp. 603–627, 2002.

[5] L. Lin, Q. Kong, J. Jiang, and G. Xia, "A unified model for zero-shot music source separation, transcription and synthesis," in *The 22nd International Society for Music Information Retrieval Conference (ISMIR)*, 2021.

[6] K. W. Cheuk, K. Choi, Q. Kong, B. Li, M. Won, J.-C. Wang, and Y.-N. H. D. Herremans, "Jointist: Simultaneous improvement of multi-instrument transcription and music source separation via joint training," *arXiv preprint arXiv:2302.00286*, 2023.

[7] C. Dittmar, E. Cano, J. Abeßer, and S. Grollmisch, "Music information retrieval meets music education," in *Dagstuhl Follow-Ups*, vol. 3. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2012.

[8] E. Cano, G. Schuller, and C. Dittmar, "Pitch-informed solo and accompaniment separation towards its use in music education applications," *EURASIP Journal on Advances in Signal Processing*, vol. 2014, pp. 1–19, 2014.

[9] C. T. Hoopen, "Issues in timbre and perception," *Contemporary Music Review*, vol. 10, no. 2, pp. 61–71, 1994.

[10] H. Scudder, "Probability of error of some adaptive pattern-recognition machines," *IEEE Transactions on Information Theory*, vol. 11, no. 3, pp. 363–371, 1965.

[11] D. Yarowsky, "Unsupervised word sense disambiguation rivaling supervised methods," in *33rd annual meeting of the association for computational linguistics*, 1995, pp. 189–196.

[12] Q. Xie, M.-T. Luong, E. Hovy, and Q. V. Le, "Self-training with noisy student improves imagenet classification," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 10 687–10 698.

[13] J. Schlüter, "Learning to pinpoint singing voice from weakly labeled examples." in *ISMIR*, 2016, pp. 44–50.

[14] 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 *The 19th International Society for Music Information Retrieval Conference (ISMIR)*, 2018.

[15] S. Keum, J.-H. Lin, L. Su, and J. Nam, "Semi-supervised learning using teacher-student models for vocal melody extraction," in *The 21th International Society for Music Information Retrieval Conference (ISMIR)*, 2020.

[16] E. Benetos, M. Kotti, and C. Kotropoulos, "Musical instrument classification using non-negative matrix factorization algorithms and subset feature selection," in *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, vol. 5. IEEE, 2006, pp. V–V.

[17] A. Eronen and A. Klapuri, "Musical instrument recognition using cepstral coefficients and temporal features," in *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 00CH37100)*, vol. 2. IEEE, 2000, pp. II753–II756.

[18] V. Lostanlen and C.-E. Cella, "Deep convolutional networks on the pitch spiral for musical instrument recognition," in *The 17th International Society for Music Information Retrieval Conference (ISMIR)*, 2016.

[19] S. Essid, G. Richard, and B. David, "Hierarchical classification of musical instruments on solo recordings," in *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, vol. 5. IEEE, 2006, pp. V–V.

[20] Y. Han, J. Kim, and K. Lee, "Deep convolutional neural networks for predominant instrument recognition in polyphonic music," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 1, pp. 208–221, 2016.

[21] Y.-N. Hung and Y.-H. Yang, "Frame-level instrument recognition by timbre and pitch," in *The 19th International Society for Music Information Retrieval Conference (ISMIR)*, 2018.

[22] S. Gururani, C. Summers, and A. Lerch, "Instrument activity detection in polyphonic music using deep neural networks." in *The 19th International Society for Music Information Retrieval Conference (ISMIR)*, 2018, pp. 569–576.

[23] S. Gururani, M. Sharma, and A. Lerch, "An attention mechanism for musical instrument recognition," in *The 20th International Society for Music Information Retrieval Conference (ISMIR)*, 2019.

[24] Y.-N. Hung, Y.-A. Chen, and Y.-H. Yang, "Multi-task learning for frame-level instrument recognition," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 381–385.

[25] H. Flores Garcia, A. Aguilar, E. Manilow, and B. Pardo, "Leveraging hierarchical structures for few-shot musical instrument recognition," in *The 22nd International Society for Music Information Retrieval Conference (ISMIR)*, 2021.

[26] L. C. Reghunath and R. Rajan, "Transformer-based ensemble method for multiple predominant instruments recognition in polyphonic music," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2022, no. 1, p. 11, 2022.

[27] K. Kim, M. Park, H. Joung, Y. Chae, Y. Hong, S. Go, and K. Lee, "Show me the instruments: Musical instrument retrieval from mixture audio," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[28] Z. Zhong, M. Hirano, K. Shimada, K. Tateishi, S. Takahashi, and Y. Mitsufuji, "An attention-based approach to hierarchical multi-label music instrument classification," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[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 *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2017, pp. 261–265.

[30] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *International Conference on Learning Representations*, 2018.

[31] 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*, 2022, pp. 11 976–11 986.

[32] 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.

[33] M. Kim, W. Choi, J. Chung, D. Lee, and S. Jung, "Kuielab-mdx-net: A two-stream neural network for music demixing," *arXiv preprint arXiv:2111.12203*, 2021.

[34] S. Rouard, F. Massa, and A. Défossez, "Hybrid transformers for music source separation," *arXiv preprint arXiv:2211.08553*, 2022.

[35] Y. Luo and J. Yu, "Music source separation with band-split rnn," *arXiv preprint arXiv:2209.15174*, 2022.

[36] 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. 18, 2022.

[37] Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimilakis, and R. Bittner, "Musdb18-hq - an uncompressed version of musdb18," Aug. 2019. [Online]. Available: https://doi.org/10.5281/zenodo.3338373

[38] A. Défossez, "Hybrid spectrogram and waveform source separation," *arXiv preprint arXiv:2111.03600*, 2021.

[39] R. Sawata, S. Uhlich, S. Takahashi, and Y. Mitsufuji, "All for one and one for all: Improving music separation by bridging networks," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 51–55.

[40] C.-B. Jeon and K. Lee, "Towards robust music source separation on loud commercial music," in *Proc. of the 23rd Int. Society for Music Information Retrieval Conference*, 2022.

[41] C. J. Steinmetz and J. D. Reiss, "pyloudnorm: A simple yet flexible loudness meter in python," in *150th AES Convention*, 2021.

[42] R. ITU-R, "Itu-r bs. 1770-2, algorithms to measure audio programme loudness and true-peak audio level," *International Telecommunications Union, Geneva*, 2011.

[43] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[44] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations*, 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] F. Stöter, A. Liutkus, and N. Ito, "The 2018 signal separation evaluation campaign," in *International Conference on Latent Variable Analysis and Signal Separation*. Springer, 2018, pp. 293–305.

QUANTIFYING THE EASE OF PLAYING SONG CHORDS ON THE GUITAR

Marcel A. Vélez Vásquez[1] **Mariëlle Baelemans**[1] **Jonathan Driedger**[2]
Willem Zuidema[1] **John Ashley Burgoyne**[1]
[1] ILLC, University of Amsterdam, the Netherlands [2] Chordify, Groningen, the Netherlands
m.a.velezvasquez@uva.nl

ABSTRACT

Quantifying the difficulty of playing songs has recently gained traction in the MIR community. While previous work has mostly focused on piano, this paper concentrates on rhythm guitar, which is especially popular with amateur musicians and has a broad skill spectrum. This paper proposes a rubric-based 'playability' metric to formalise this spectrum. The rubric comprises seven criteria that contribute to a single playability score, representing the overall difficulty of a song. The rubric was created through interviewing and incorporating feedback from guitar teachers and experts. Additionally, we introduce the playability prediction task by adding annotations to a subset of 200 songs from the McGill Billboard dataset, labelled by a guitar expert using the proposed rubric. We use this dataset to weight each rubric criterion for maximal reliability. Finally, we create a rule-based baseline to score each rubric criterion automatically from chord annotations and timings, and compare this baseline against simple deep learning models trained on chord symbols and textual representations of guitar tablature. The rubric, dataset, and baselines lay a foundation for understanding what makes songs easy or difficult for guitar players and how we can use MIR tools to match amateurs with songs closer to their skill level.

1. INTRODUCTION

Guitars have seen a 1.25-million-instrument sales rebound since the coronavirus pandemic, and the public's fascination with fretted instruments has never been higher [1]. While traditional methods of transferring musical playability knowledge via music schools or private teachers still exist, online resources have made learning to play the guitar more accessible [2]. Indeed, online tools have led to a significant increase in the accessibility of learning *any* musical instrument, with a growing number of children and adults learning to play [3]. In addition, research suggests that informal self-practice can enhance motivation compared to formal teaching [4]. Ultimate Guitar and Chordify are examples of web-based music services that facilitate the automatic extraction of chord progressions from audio recordings of songs or community-proposed chord transcriptions and present them in a simple and accessible format for the growing group of amateur guitar players to use for practice and pleasure. Currently, Ultimate Guitar and Chordify have 39.7 million and 8 million users, respectively [5, 6].

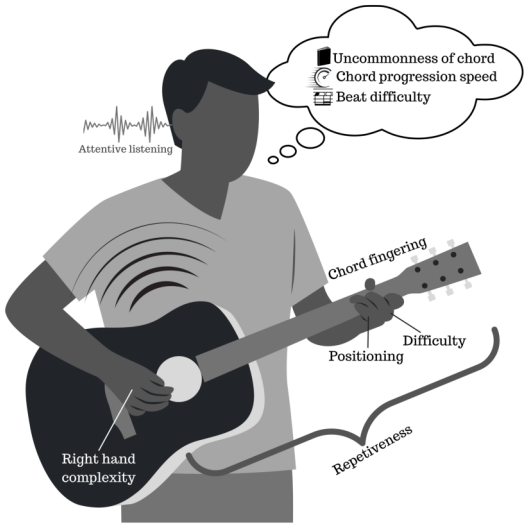


Figure 1. Physical and cognitive criteria for evaluating the playability of songs on the guitar position during guitar performance. Note that repetitiveness reflects both cognitive and physical factors, and that attentive listening to auditory feedback, while not a criterion itself, is necessary for developing and refining performative gestures.

Navigating the abundance of online chord data on platforms such as Ultimate Guitar or Chordify can be overwhelming, however, particularly for amateur learners seeking suitable pieces to enhance their expertise. While Chordify offers only a chord simplification option, Ultimate Guitar offers four categories of playability: absolute beginner, beginner, intermediate, and expert. Still, these categories may be too broad to suit all individuals. There is a need to establish a method that can predict a song's difficulty level in a more fine-grained, automated, and preferably interpretable manner to assist learners in selecting appropriate pieces based on their skill level and personal taste.

© M.A. Vélez Vásquez, M.C.E. Baelemans, J. Driedger, W.H. Zuidema, and J.A. Burgoyne. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** M.A. Vélez Vásquez, M.C.E. Baelemans, J. Driedger, W.H. 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.,* Milan, Italy, 2023.

The Ultimate Guitar community has proposed a difficulty measurement system, which relied until recently on the input of multiple users, but like any system relying on human annotation, it is difficult to scale and can suffer from low reliability unless annotators are well-qualified and familiar with the annotation scheme.

This paper argues that a robust, reliable, and publicly documented difficulty prediction system could significantly benefit music learners in selecting challenging and rewarding pieces. Our main contributions are: (1) an interpretable guitar playability metric; (2) an extension of the Billboard dataset of 200 playability annotated songs, tested for reliability; and (3) a rule-based baseline for our playability metric. Furthermore, we investigated how well a previously well-performing model of piano playability compares to our rule-based baseline when trained on our dataset. The rule-based baseline and source code for all experiments are available to download. [1] We also include dataset statistics and other information to aid future research on playability.

2. RELATED WORK

We define *playability* as the level of musical proficiency required to perform a musical song on a specific instrument. While it is a crucial aspect of musical analysis and performance, it is a complex and challenging concept to measure or quantify. The playability of musical songs can be influenced by various factors, such as the complexity of the musical structure [7], the instrument of choice [8], and the musical context in which it is played [9]. In addition, individual musical competence for a particular song requires developing physical and cognitive skills and is influenced by personality [10]. Physical skills for the guitar include refining gestural mechanics, both left (fret fingering) and right (strumming) hand positioning [11]; cognitive skills include a comprehensive understanding of music theory, the ability to read musical scores, and attentive listening to the auditory feedback of the instrument for monitoring and planning of the performative gestures [12, 13].

Several studies have attempted to develop methods for automating the estimation of the difficulty level of piano sheet music. In 2012, researchers proposed a method that used MusicXML and seven high-level, instrument-agnostic criteria to determine the difficulty level of a song [14]. They evaluated the accuracy of their criteria by testing them on 50 piano pieces and validated their performance using principal component analysis and human judgement. Although their criteria were not instrument-specific, some of their categories aligned with or were similar to those used in other studies. Another study focused on predicting the difficulty level of piano sheet music using regression [15]. The authors proposed using RReliefF, a method for selecting relevant symbolic music features, to improve their performance, yielding R^2 values of up to .40.

In a recent study, researchers developed a piano score difficulty classification task and a novel difficulty score dataset [16]. They used a gated recurrent unit (GRU) neural network with an attention mechanism and gradient-boosted trees to train their model on segments of musical scores with various piano-fingering representations. They derived the skill levels for each song from a musical practice-book series, where the editions were ordered based on difficulty. Books 1 and 2 were easier, classified as beginner by the authors; Books 3 and 4 as intermediate; and Books 5 and 6 as professional. They showed that novel piano fingering features were indicative of difficulty. Both machine-learning models performed better than their simple baseline, with the GRU with attention mechanisms performing best.

There has been limited research devoted to the investigation of guitar playability. Some studies have incorporated algorithmic proxies as a means of evaluating guitar playability [17]. Meanwhile, others have primarily focused on left-hand fingering aspects [18]. However, a conspicuous gap in the existing literature is the lack of manual annotation of difficulty by human experts. Like the practice-book dataset, any automatic system for assessing playability requires good human-generated ground truth. To address this challenge and move the scope from piano to guitar playability, we introduce a rubric-based metric to formalise the broad spectrum of playability levels.

3. A RUBRIC FOR GUITAR PLAYABILITY

In order to develop a rubric for guitar-playing difficulty, we interviewed local guitar experts, including guitar teachers, to investigate what they believed makes a song challenging to play, and what they consider when developing teaching material for a student (e.g., why it would or would not be suitable for their students, and how they simplify the chord progressions to make songs more accessible). Based on these interviews, we created a list of categories appropriate for evaluating playability and formulated four difficulty levels within each criterion, with a textual description for each level. We revised this initial draft by considering whether categories had too much overlap, and rephrased the names and level descriptions for each criterion accordingly. We requested and incorporated feedback on the updated rubric from two musical experts, and finally had a guitar expert annotate five songs with the rubric and give feedback as to whether it allowed annotating the data efficiently.

The final version of the rubric is in Table 1. It includes seven criteria: (1) 'uncommonness of chord', capturing the possibility of the player having played the chords in the specific song before, where unknown chords increase difficulty; (2) 'chord finger positioning', capturing how comfortably spaced the fingers on the guitar fretboard are positioned, wherein chords are more difficult to play if they contain very stretched out or cramped finger positions than when the fingers are close together and in a relaxed position; (3) 'chord fingering difficulty', capturing how many fingers a chord requires and the ratio of barre chords played in a song, based on guitar teaching books' build-up of number of fingers used, and later on to barre chords; (4) 'repetitiveness', capturing that a song is easier to play if it has more repetition since it requires less task switching than a less repetitive song; (5) 'right hand complexity',

[1] https://github.com/Marcel-Velez/playability-billboard

Criterion	Weight	Very difficult (3 points)	Difficult (2 points)	Easy (1 point)	Very Easy (0 points)
Uncommonness of chord	3	A lot of uncommon chords	Some uncommon chords	Few uncommon chords	No uncommon chords
Chord finger positioning	3	Very cramped or very wide fingerspread	Uncomfortable or spread out fingers	Slightly uncomfortable or spread out fingers	Comfortable hand and finger position
Chord fingering difficulty	2	Mostly chords that require four fingers or barre chords	Some chords require four fingers to be played or are barre chords (not A or E)	Most chords require three fingers or are A or E barre chords	Most chords can be played with two or three fingers
Repetitiveness	2	No repeated chord progressions	A few repeated chord progressions	Quite a bit of repetition of chord progressions	A lot of repetition of chord progressions
Right-hand complexity	2	For some chords multiple inner strings are not strummed	For some chords one inner string is not strummed	For some of the chords one or more outer strings are not strummed	For the chords all strings are strummed
Chord progression time	1	Very quick chord transitions	Quick chord transitions	Slow chord transitions	Very slow chord transitions
Beat difficulty (syncopes/ghostnotes)	0	A lot of syncopes or ghostnotes	Some syncopes or ghostnotes	A few syncopes or ghostnotes	No syncopes or ghostnotes

Table 1. Proposed rubric for human annotators evaluating the difficulty of playing the chords of a song on the guitar. Although the rubric functions acceptably using the raw scores from the table header, it has even better predictive power when weighting the criteria according to the factor in the weight column. Note that the beat difficulty criterion provides so little extra information that we recommend omitting it (i.e., setting its weight to zero).

capturing how difficult the strumming is, where dampening or skipping inner strings is thought to be more difficult for strumming than skipping outer strings or just strumming all strings; (6) 'chord progression tempo', covering the tempo at which the individual has to switch between chords, wherein matching the correct finger positions is linked to the playability proficiency of the individual; and (7) 'beat difficulty', which models the regularity of the beat within a song, a more regular strum being easier to play than irregular strumming, and mixed regularity like that typical of the reggaeton genre being easier than fully irregular beat patterns. Figure 1 visualises these criteria in the context of actual guitar playing and organises them into physical and cognitive factors. The purpose of the rubric is to generate a single, overall playability score as the sum of scores for each rubric category. As will be discussed in more detail below, while a simple unweighted sum of points for each criterion already provides a reliable measure of playability, the reliability is improved even further by using a weighted sum, with uncommonness and finger positioning receiving the most weight and beat difficulty the least.

Our playability rubric focuses on *rhythm guitar* playability over solo guitar playability: in other words, we are not interested in melodies but rather in how difficult it is for guitar players to reproduce the chord progressions and rhythms of Western-style pop music. For MIR research surrounding chords and timing in Western-style pop music, one of the most frequently-used datasets is the McGill Billboard dataset [19]. The original Billboard dataset consists of 740 songs that were part of the Billboard Hot 100 chart between 1958 and 1991 and have been part of the MIREX challenges. Each song has time-aligned chord transcriptions and higher-level structural information, including meter and phrase. Since its release, other researchers have enriched the Billboard dataset with further information (e.g., the Billboard sub-corpus of the CoCoPops project [20] and the Chord Annotator Subjectivity Dataset [21]). We decided to do the same as a testing ground for our playability rubric, creating the Billboard Playability Dataset.

4. THE BILLBOARD PLAYABILITY DATASET

As a basis for our dataset, we started with the 50 songs that are included in the Chord Annotator Subjectivity Dataset. Of the remaining 690 songs that appear in the original dataset and CoCoPops, we chose a random sample of 150, bringing the total number of songs in Billboard Playability Dataset to 200. In total, these 200 songs comprise 31 205 chords, 27 190 bars, and 5852 phrases.

For each song, we acquired the audio and made an online annotation dashboard with an audio player on the top, the Billboard chord transcriptions (including timing and phrasing information) on the left, and the rubric to the right. To create the dataset, we enlisted the assistance of a guitar expert who has previously demonstrated exceptional guitar skills and experience with other music annotation tasks. We instructed the annotator to perform the songs as written (i.e., without using a capo or making other simplifications, and also not adding extensions beyond those notated in the Billboard dataset), but they were free to choose any appropriate fingering. After using our dashboard to listen and play along with the song, the annotator filled in the rubric. Six of the songs appeared twice, unbeknownst to the annotator, and were scored similarly each time (maximally 5 points different on the weighted scale, whereas the standard deviation across all scores in the dataset was 6.6 points).

Histograms of the overall distributions per rubric criterion are in Figure 2. For one criterion, repetitiveness, the most difficult category was never used, which is somewhat to be expected given that all of the songs in the dataset are mainstream Western pop music. Most pop music tends to have some form of repetition, and not to consist of the unique chords and phrases that are characteristic of more

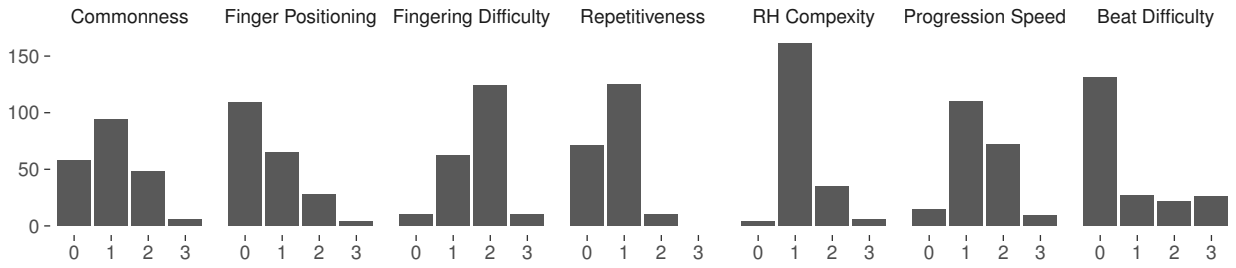


Figure 2. Histograms of playability scores per rubric criterion.

Bin	Chords	Bars	Phrases
All songs	156.03 (87.92)	135.95 (55.34)	29.26 (12.41)
Easy 25%	139.21 (85.66)	133.17 (44.64)	27.87 (10.87)
Moderate 25%	152.25 (73.79)	132.69 (42.13)	28.63 (9.48)
Hard 25%	158.29 (96.40)	135.59 (65.01)	28.27 (14.10)
Expert 25%	175.20 (89.84)	142.47 (64.69)	31.35 (14.19)

Table 2. Mean, and standard deviation (in brackets) of the number of chords, bars, and phrases for the entire dataset and per playability bins. The playability bins are based on quartiles of the weighted total score of the songs, the easiest having a score lower than 8, moderate lower than 12.5, hard lower than 18, and expert higher than 18.

Song	Artist	Score
Stand By Me	David and Jimmy Ruffin	1
Miss You	The Rolling Stones	2
No Charge	Melba Montgomery	2
Jungle Boogie	Kool and the Gang	2
Sunshine of Your Love	Cream	2
I Don't Need You	Kenny Rogers	28
Man In The Mirror	Michael Jackson	28
One Less Bell To Answer	The 5th Dimension	30
That Girl	Stevie Wonder	31
Do I Do	Stevie Wonder	34

Table 3. Easiest and most difficult songs in the dataset with their weighted playability scores.

experimental genres [22].

5. CAN PLAYABILITY BE MEASURED?

Given the inherent subjectivity in the concept of playability, one could be forgiven for wondering whether predicting playability is a well-posed question at all. Is there any common underlying measure of playability for the guitar, or is it merely a more-or-less arbitrary combination of criteria such as those we collected from guitar teachers for our rubric? To address this concern, we checked our annotator's scores for *reliability*: if one tries to predict our annotator's rubric scores from a single parameter per song, what proportion of variance in that parameter is 'true' variance as opposed to measurement error? Reliability can also be seen as the extent to which the rubric criteria co-vary, with high reliability indicating high covariance (and thus that all criteria are measuring a common underlying phenomenon), or alternatively, as the proportion of variance explained by the first principal component. Values of 0.7 or higher are desirable for this type of assessment [23].

Formally, we used a family of models known as *partial credit models* to assess reliability [24, 25]:

$$\mathrm{P}\left[x_{ni}\right] = \frac{\mathrm{e}^{\sum_{k=1}^{x_{ni}} \alpha_{ik}(\theta_n - \delta_{ik})}}{\sum_{k'=0}^{K} \mathrm{e}^{\sum_{k=1}^{x_{ni}} \alpha_{ik'}(\theta_n - \delta_{ik'})}} \quad , \qquad (1)$$

where x_{ni} denotes the rubric score given to song n for criterion i, $x_{ni} \in \{0, 1, \ldots, K\}$, θ_n represents the underlying difficulty of song n, the δ_{ik} are threshold parameters for each level of rubric criterion i, and the $\alpha_{ik} > 0$ represent the increase in difficulty score when moving from level $k-1$ to level k on rubric criterion i. We considered three variants of the model: (1) the simple partial credit model, for which all α_{ik} are fixed to one (corresponding to a simple tally of rubric scores); (2) the generalised partial credit model, for which α_{ik} is allowed to vary in i but not in k (corresponding to the weighted rubric scores in Table 1); and (3) the extended partial credit model, for which the α_{ik} vary freely.

We fit all three models to the Billboard Playability Dataset using a hierarchical Bayesian implementation in Stan. The model included two hyperparameters μ and σ with priors $\mu \sim \mathrm{N}(0, 1)$ and $\sigma \sim \mathrm{Exp}(1)$. Given these hyperparameters, the remaining priors were $\alpha_{ik} \sim \mathrm{Exp}(1)$, $\theta_n \sim \mathrm{N}(0, 1)$, $\delta_{ik} \sim \mathrm{N}(\mu, \sigma)$. We computed reliability according to the customary partial-credit formula [26]: the variance of the estimated song difficulties θ_n divided by the true difficulty variance. Because the true variance in our model is fixed to unity, we could estimate reliability directly as the variance of the set of posterior means $\hat{\theta}_n$. We compared the three models using approximate leave-one-out cross-validation [27]. The extended partial credit model performed best, but the generalised partial credit model was statistically indistinguishable from it (expected log probability difference = 8.7, *SE* = 5.2). The simple partial credit model was somewhat worse (elpd = 105.5, *SE* = 14.2). All models, however, showed good reliability: 0.74 for the simple partial credit model, 0.84 for the generalised, and 0.86 for the extended.

Given these results, we recommend the generalised partial credit model, which is statistically indistinguishable from the extended model and more parsimonious. The simple 3–3–2–2–2–1–0 weighting scheme accompanying the rubric in Table 1 falls within 90% credible intervals for all α_{ik} values from this model fit. Table 2 provides

some descriptive statistics for the dataset and quartile-based 'playability bins' under this weighting, and Table 3 lists the easiest and most difficult songs in the dataset. We can see an apparent increase in the mean number of chords, bars and phrases, which as described later in this paper, inspired us to try classifying difficulty based on length alone.

6. CAN PLAYABILITY BE PREDICTED?

In short, the rubric we developed can be used by expert guitarists to measure playability reliably, especially when weighting the criterion scores according to the generalised partial credit model. Expert annotation is expensive, however, and MIR can add value by automating this process.

6.1 Rule-Based Model

First, we developed a heuristic model as a baseline for comparison against more sophisticated learning methods. For those rubric criteria involving potentially different per-chord difficulties (e.g., fingering difficulty), we used a TF-IDF weighted average of the difficulty heuristic over all chords in the song:

$$\sum_{\text{c}} \text{TF}(c) \times \text{IDF}(c) \times \text{difficulty}(c) \tag{2}$$

where difficulty(c) represents the difficulty score associated with a specific chord, considering factors such as chord finger positioning (CFP), chord fingering difficulty (CFD), or Right-hand complexity (RHC). In our case, TF is how often a chord appears in a song divided by the number of chords in the said song, and IDF is the log of the total number of songs divided by the number of songs that contain that chord. For the criteria that depend on fingering, we assumed one possible fingering per chord based on an extensive list of set finger positions on the Chordify website. We also had to simplify certain chords for which standard fingerings proved difficult to find, for example, chord with extensions like ♯11; we added a simplification penalty to compensate.

Uncommonness of chord (UC) uses a difficulty of one for all chords (i.e., it is the average TF-IDF weight).

Chord finger positioning (CFP) requires the guitar diagram and is based on a naïve approach of counting the distance between the lowest and highest played fret, not considering which strings they are played.

$$\text{CFP} = (1 + \text{simplified} \times f_{\text{simple}}) \times \text{finger distance}$$

Chord fingering difficulty (CFD) is based on how many fingers are used, and if a finger is used for more than one string, it is counted as a barre chord. For this criterion, we had three learnable parameters, one for the importance of how many fingers were used, one the importance of barre chords, and one for simplification.

$$\text{CFD} = (1 + \text{simplified} \times f_{\text{simple}}) \times (\text{fingers} * f_{\text{finger}} + \text{bar} * f_{\text{bar}})$$

Repetitiveness (R) is the number of *unique* phrases in a song according to the Billboard annotations.

Right-hand complexity (RHC) is based on apply the rubric level descriptions to fingering diagrams.

$$\text{RHC} = \begin{cases} 0 & \text{if no un-strummed strings} \\ 1 & \text{if outer strings not strummed} \\ 2 & \text{if one inner string not strummed} \\ 3 & \text{if multiple inner strings not strummed} \end{cases}$$

Chord progression time (CPT) is the average chord duration (in s) according to the Billboard annotations.

Beat difficulty (BD) is the ratio of chords that were longer or shorter than the most common chord duration in the Billboard annotations.

Given these preliminary scores per criterion, averaged according to TF-IDF weights as necessary, we iterated over all annotations in Billboard Playability Dataset and grid-searched for the three optimal thresholds, one between each pair of adjacent difficulty levels. For categories with learnable parameters, we extended the grid search accordingly.

6.2 Classification Experiments

In addition to the rule-based model we also trained neural networks on the playability prediction task using two architectures: LSTMs and DeepGRU with attention, which have been applied recently to piano playability [16, 28]. We replicated the same parameter settings as used in these papers. Inspired by our findings on length and difficulty above, we also included models using *only* representation length, with thresholds trained in the same way as the rule-based model.

For each architecture, we tested three distinct types of input: (1) processing each song character by character, which does not explicitly imply chord information (e.g. A:maj → 'A', ':', 'm', 'a', 'j'); (2) splitting each chord into root and quality and treating those as unique input symbols, similar to music-theoretical understanding (e.g. A:maj → 'A', 'maj'); and (3) converting each chord into the corresponding guitar tablature, guitar-neck-like encodings displaying each of the six guitar strings with an 'x' label if it is skipped, 'o' if it is open, or which finger goes on which fret otherwise (e.g., A:maj → ['x', 'o', '2:1', '2:2', '2:3', 'o'], where '2:1' represents the 2nd fret being played by the first finger).

Given the characteristics of our rubric, we defined a custom loss function OL, which enforces an ordinal-like structure in the class prediction:

$$\text{OL} = \sum_{i=0}^{3} \rho_i \times (\text{target} - i) \quad , \tag{3}$$

where ρ_i is the predicted probability of level i for the criterion in question. We trained the models in two settings: first to predict the total weighted playability score, and then to predict each individual criterion in turn. For all training configurations, we subdivided our dataset into 10 sections for our experiments and conducted 10-fold cross-validation.

Model	Input	CFP ↓	CFD ↓	UC ↓	RHC ↓	CPT ↓	BD ↓	R ↓	Aggregate ↓
Rule-based	-	1.04 (0.05)	0.85 (0.04)	0.95 (0.05)	0.78 (0.05)	0.90 (0.05)	1.20 (0.06)	0.93 (0.06)	12.38 (0.52)
Length-based	char	1.09 (0.03)	0.88 (0.03)	1.01 (0.03)	0.80 (0.01)	0.87 (0.03)	1.20 (0.04)	0.94 (0.06)	12.46 (0.36)
Length-based	split	1.09 (0.02)	0.88 (0.03)	1.02 (0.03)	0.80 (0.01)	**0.86 (0.03)**	1.19 (0.04)	0.95 (0.05)	12.46 (0.35)
Length-based	diagram	1.09 (0.02)	0.88 (0.02)	1.02 (0.04)	0.80 (0.01)	**0.86 (0.03)**	1.19 (0.04)	0.95 (0.05)	12.46 (0.36)
LSTM	char	0.75 (0.18)	**0.50 (0.08)**	0.68 (0.11)	0.34 (0.13)	1.25 (0.14)	**0.74 (0.24)**	**0.70 (0.15)**	**5.27 (0.77)**
LSTM	split	0.77 (0.14)	0.52 (0.11)	**0.65 (0.08)**	0.33 (0.13)	1.25 (0.12)	0.77 (0.24)	0.72 (0.14)	5.96 (1.40)
LSTM	diagram	0.78 (0.14)	0.51 (0.08)	**0.65 (0.10)**	0.35 (0.13)	1.27 (0.15)	0.79 (0.23)	0.72 (0.13)	6.20 (1.02)
DeepGRU	char	**0.67 (0.18)**	0.60 (0.24)	0.77 (0.21)	0.47 (0.39)	0.92 (0.42)	1.10 (0.54)	**0.70 (0.15)**	5.61 (1.17)
DeepGRU	split	0.69 (0.15)	**0.50 (0.10)**	0.66 (0.22)	**0.30 (0.14)**	1.22 (0.30)	1.00 (0.28)	0.80 (0.29)	5.88 (0.93)
DeepGRU	diagram	0.68 (0.17)	0.55 (0.18)	0.80 (0.27)	0.80 (0.53)	0.90 (0.48)	0.84 (0.20)	0.96 (0.58)	5.99 (1.12)

Table 4. Playability prediction performances after training on the Billboard Playability Dataset. The columns are the performance when trained on and predicting each of the seven categories independently, followed by the error between all individual categories added together for the baselines and the error when trained to directly predict the aggregated score for the LSTM and DeepGRU models. Performances are reported in mean ordinal loss over 10 fold cross-validation with their standard deviation. The overall best performing model is the LSTM with chords split into root and quality, except for the two time-dependant categories: chord progression time (CPT) and beat difficulty (BD).

7. RESULTS

Our rule-based model performs better than the length-based difficulty predictions except for the chord progression time and beat difficulty category, as seen in Table 4. Since we use three different chords representations, each of which yield different lengths, we show length-based classification results for each representation, but in practice, these differences seem to play a negligible role in playability prediction based on length. All three length baselines-based achieve very similar losses for all categories.

When looking at the machine-learning models, we see that they are more variable, but on average substantially better, than all baseline models, both in classifying each criterion separately and predicting the weighted total difficulty. The only criterion where machine-learning models perform worse is the chord progression time. This criterion expresses the speed difficulty, which is characterised by chord duration. The lack in performance can be explained by the fact that the chord transcriptions which form the input to our model do not contain this duration information. Oddly, both machine learning models do outperform the baseline in predicting beat difficulty, which is also dependent on chord duration. When taking the histogram for this criterion into account, however, this performance can be explained by class imbalance: trying to set thresholds is worse than simply settling on the largest class. The same class imbalance is likely responsible for the partial-credit models assigning such a low weight.

Although there is no obvious best model when looking across performance on the individual criteria, the LSTM does show less variability than DeepGRU, and the LSTM trained on character input performs significantly better on predicting the weighted total score. We expected a bigger difference in input type, with the guitar chord diagram performing the best because this chord representation encodes the most guitar playing information, but this turned out to be the worst performing input type of the three. We hypothesise this is caused by the sequential models not picking up on the guitar or hand-related physics.

8. CONCLUSION

In this paper, we introduced a novel rubric that captures the playability of guitar songs. This rubric comprises seven criteria that can be combined into a single playability score. Next to this rubric, we also introduced the Billboard Playability Dataset, 200 playability annotations for songs from the Billboard dataset, which we used to validate the rubric's reliability and confirm that indeed, guitar playability can be measured. Following these results, we developed several models for playability prediction. As a baseline, we started with a rule-based model that follows the rubric as mechanically as possible. We then trained and evaluated an LSTM and DeepGRU on three different types of chord representations. The representation encoding the least guitar – only using textual characters – surprisingly performed best, and the representation encoding the most guitar chord information – guitar tablature – performed the worst. Nevertheless both LSTM and DeepGRU outperformed the rule-based model with the LSTM performing the best at predicting the overall playability. In future work, we aim to extend both the dataset and the models to capture more nuances of playability, and we hope this work will encourage and enable more MIR researchers to explore the field of playability and improve online instrument learning environments. Additionally, we envision the potential extension of our research to incorporate MusicXML or GuitarPro formats, enabling the integration of our playability scores and models into widely used music notation software.

9. ACKNOWLEDGEMENTS

We are sincerely grateful to Jeanine Sier, Barbara de Bruin, Robin Willems, and Tom Strandberg for their valuable input and feedback on the rubric, and to Tom again for diligently annotating the songs. This research was supported by the Dutch Research Council (NWO) as part of the project InDeep (NWA.1292.19.399). Additionally, we would like to express our appreciation to Chordify for their generous support in funding the annotations.

10. REFERENCES

[1] A. Williams, "Guitars are back, baby!" *The New York Times*, Sep. 2020. [Online]. Available: https://www.nytimes.com/2020/09/08/style/guitar-sales-fender-gibson.html

[2] R. C. Rodriguez and V. Marone, "Guitar learning, pedagogy, and technology: A historical outline," *Social Sciences and Education Research Review*, vol. 8, no. 2, pp. 9–27, Dec. 2021.

[3] The Associated Board of the Royal Schools of Music, "Making music: Teaching, learning & playing in the UK," 2014. [Online]. Available: https://gb.abrsm.org/media/12032/makingmusic2014.pdf

[4] R. Reynolds and M. M. Chiu, "Formal and informal context factors as contributors to student engagement in a guided discovery-based program of game design learning," *Learning, Media and Technology*, vol. 38, no. 4, pp. 429–462, 2013.

[5] "Chordify: Het Groningse paradepaardje van de IT- en muziekindustrie heeft al acht miljoen gebruikers," *Groninger ondernemers courant*, Jan. 2023. [Online]. Available: https://www.groningerondernemerscourant.nl/nieuws/chordify-het-groningse-paradepaard-van-de-it–en-muziekindustrie-heeft-al-acht-miljoen-gebruikers

[6] Ultimate Guitar. [Online]. Available: https://www.ultimate-guitar.com/forum/

[7] J.-P. Boon, A. Noullez, and C. Mommen, "Complex dynamics and musical structure," *Journal of New Music Research*, vol. 19, no. 1, pp. 3–14, 1990.

[8] T. Magnusson, "Of epistemic tools: Musical instruments as cognitive extensions," *Organised Sound*, vol. 14, no. 2, p. 168176, 2009.

[9] A. Chirico, S. Serino, P. Cipresso, A. Gaggioli, and G. Riva, "When music flows. state and trait in musical performance, composition and listening: A systematic review," *Frontiers in Psychology*, vol. 6, p. 906, 2015.

[10] S. Swaminathan and E. G. Schellenberg, "Musical competence is predicted by music training, cognitive abilities, and personality," *Scientific Reports*, vol. 8, no. 9223, 2018.

[11] J. De Souza, "Guitar thinking," *Soundboard Scholar*, vol. 7, no. 1, p. 1, 2022.

[12] R. M. Brown, R. J. Zatorre, and V. B. Penhune, "Expert music performance: Cognitive, neural, and developmental bases," *Progress in Brain Research*, vol. 217, pp. 57–86, 2015.

[13] C. Palmer, "Music performance," *Annual Review of Psychology*, vol. 48, no. 1, pp. 115–138, 1997.

[14] V. Sébastien, H. Ralambondrainy, O. Sébastien, and N. Conruyt, "Score analyzer: Automatically determining scores difficulty level for instrumental e-learning," in *Proceedings of the 13th International Society for Music Information Retrieval Conference*, Porto, Portugal, 2012, pp. 571–576.

[15] S.-C. Chiu and M.-S. Chen, "A study on difficulty level recognition of piano sheet music," *IEEE International Symposium on Multimedia*, pp. 17–23, 2012.

[16] P. Ramoneda, N. C. Tamer, V. Eremenko, X. Serra, and M. Miron, "Score difficulty analysis for piano performance education based on fingering," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2022, pp. 201–205.

[17] G. Hori and S. Sagayama, "Minimax Viterbi algorithm for hmm-based guitar fingering decision." in *Proceedings of the 17th International Society for Music Information Retrieval Conference*, New York, New York, 2016, pp. 448–453.

[18] S. Ariga, S. Fukayama, and M. Goto, "Song2guitar: A difficulty-aware arrangement system for generating guitar solo covers from polyphonic audio of popular music." in *Proceedings of the 18th International Society for Music Information Retrieval Conference*, Suzhou, China, 2017, pp. 568–574.

[19] 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*, vol. 11, Miami, Florida, 2011, pp. 633–638.

[20] N. Condit-Schulz and C. Arthur. (2023) Coordinated corpus of popular music. [Online]. Available: https://github.com/Computational-Cognitive-Musicology-Lab

[21] H. V. Koops, W. B. de Haas, J. A. Burgoyne, J. Bransen, A. Kent-Muller, and A. Volk, "Annotator subjectivity in harmony annotations of popular music," *Journal of New Music Research*, vol. 48, no. 3, p. 232252, 2019.

[22] J. Pauwels, K. O'Hanlon, E. Gómez, and M. B. Sandler, "20 years of automatic chord recognition from audio," in *Proceedings of the 20th International Society for Music Information Retrieval Conference*, Delft, the Netherlands, 2019, pp. 54–63.

[23] J. C. Nunnally, *Psychometric Theory*. New York: McGraw–Hill, 1978.

[24] G. N. Masters, "A Rasch model for partial credit scoring," *Psychometrika*, vol. 47, no. 2, pp. 149–174, 1982.

[25] E. Muraki, "A generalized partial credit model: Application of an EM algorithm," *Applied Psychological Measurement*, vol. 16, pp. 159–176, 1992.

[26] B. D. Wright and G. N. Masters, *Rating Scale Analysis*. Chicago: MESA Press, 1982.

[27] 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.

[28] M. Maghoumi and J. J. LaViola, "DeepGRU: Deep gesture recognition utility," in *Advances in Visual Computing: 14th International Symposium on Visual Computing*. Berlin: Springer, 2019, pp. 16–31.

FLEXDTW: DYNAMIC TIME WARPING WITH FLEXIBLE BOUNDARY CONDITIONS

Irmak Bükey
Pomona College
ibab2018@mymail.pomona.edu

Jason Zhang
University of Michigan
zhangjt@umich.edu

TJ Tsai
Harvey Mudd College
ttsai@hmc.edu

ABSTRACT

Alignment algorithms like DTW and subsequence DTW assume specific boundary conditions on where an alignment path can begin and end in the cost matrix. In practice, the boundary conditions may not be known a priori or may not satisfy such strict assumptions. This paper introduces an alignment algorithm called FlexDTW that is designed to handle a wide range of boundary conditions. FlexDTW allows alignment paths to start anywhere on the bottom or left edge of the cost matrix (adjacent to the origin) and to end anywhere on the top or right edge. In order to properly compare paths of very different lengths, we use a normalized path cost measure that normalizes the cumulative path cost by the path length. The key insight of FlexDTW is that the Manhattan length of a path can be computed by simply knowing the starting point of the path, which can be computed recursively during dynamic programming. We artificially generate a suite of 16 benchmarks based on the Chopin Mazurka dataset in order to characterize audio alignment performance under a variety of boundary conditions. We show that FlexDTW has consistently strong performance that is comparable or better than commonly used alignment algorithms, and it is the only system with strong performance in some boundary conditions.

1. INTRODUCTION

Dynamic Time Warping (DTW) is a dynamic programming algorithm for computing the optimal alignment between two sequences under certain assumptions. In the MIR literature, it is the most widely used method for aligning two audio recordings of the same piece of music. One of its assumptions is the boundary condition of the alignment path: it assumes that the alignment path begins at the origin of the pairwise cost matrix and ends in the opposite corner of the cost matrix. When working with real data like (say) Youtube recordings of a piece of classical music, the boundary conditions are usually unknown a priori and may not satisfy the restrictive assumptions of standard DTW. This may be due to silence or applause at the beginning or end of videos, or perhaps due to some videos containing only one movement of a piece. This paper seeks to develop a more flexible variant of DTW that can handle a wider range of boundary conditions.

Previous work. There is a very large body of work on variations or extensions of DTW. These works generally fall into one of two categories. The first category focuses on mitigating the quadratic computation and memory costs of DTW. Some works approach this by speeding up an exact DTW computation through the use of lower bounds [1, 2], early abandoning [3, 4], using multiple cores [5, 6], or specialized hardware [7, 8]. Tralie and Dempsey [9] introduce a method for computing exact DTW with linear memory by processing diagonals rather than rows/columns. Other works propose approximations to DTW that require less computation, runtime, or memory. Some approaches include approximate lower bounds [10, 11], approximations of DTW distance [12, 13], imposing bands in the cost matrix to limit extreme time warping [14, 15], computing alignments at multiple resolutions [16, 17], parallelizable approximations of DTW [18], or working with a fixed amount of memory [19]. The second category focuses on extending the behavior of DTW in some way. Some examples in the MIR literature include handling structural differences like repeats and jumps in music [20–22], performing alignment in an online setting [23–25], handling partial alignments [26, 27], using multiple performances to improve alignment accuracy [28], accounting for pitch drift in a capella music [29], and aligning sets of source recordings and mixtures [30].

Shortcomings. Our work aims to make DTW more flexible by focusing on an often overlooked aspect: boundary conditions. The vast majority of previous works on DTW or its variants focus on handling one specific type of boundary condition. For example, DTW (and any of its approximations or efficient implementations) assumes that an alignment path begins at the origin of the cost matrix and ends in the opposite corner. Similarly, subsequence DTW assumes that an alignment path begins somewhere on the longer edge of the cost matrix and ends on the opposite edge. As mentioned above, in many situations the boundary conditions are unknown a priori or may be incompatible with the assumptions of standard alignment methods.

Our approach. FlexDTW is designed to be flexible in handling a wide range of boundary conditions. Assuming that the origin of the cost matrix is in the lower left corner,

© I. Bükey, J. Zhang, and T. Tsai. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** I. Bükey, J. Zhang, and T. Tsai, "FlexDTW: Dynamic Time Warping With Flexible Boundary Conditions", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

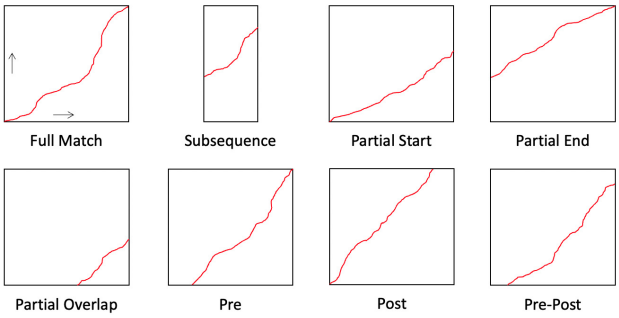


Figure 1. Different boundary conditions for the alignment path between two sequences. The full match and subsequence conditions are well handled by standard algorithms, but the other conditions are not.

FlexDTW allows an alignment path to begin anywhere on the left or bottom edge, and it allows the alignment path to end anywhere on the top or right edge. Figure 1 shows several examples of boundary conditions that FlexDTW can handle. To properly compare alignment paths of very different length, it is necessary to use a normalized path cost measure that normalizes the cumulative path cost by the path length. While it is possible to determine the optimal alignment path ending at any position by following the backpointers in the backtrace matrix, this would result in an impractically high computation overhead. The key insight with FlexDTW is that the Manhattan length of an alignment path can be computed by simply knowing the starting and ending location of the alignment path (without knowing the actual path itself). The starting location information can be computed in a recursive manner and stored during the dynamic programming stage, making it possible to compute normalized path costs in an efficient manner.

Contributions. This paper has three main contributions. First, we introduce an alignment algorithm called FlexDTW that handles a wide range of boundary conditions. FlexDTW allows an alignment path to start anywhere on the two edges of the cost matrix adjacent to the origin (e.g. bottom and left edge), and it allows alignment paths to end anywhere on the other two edges (top and right edge). Second, we design a suite of 16 benchmarks based on the Chopin Mazurka dataset [31] in order to characterize audio alignment performance under a variety of specific boundary conditions. Third, we present experimental results showing that FlexDTW has consistently strong performance across all 16 benchmarks that is comparable to or better than the best-performing system from a set of widely used audio alignment algorithms. We provide source code for our implementation of FlexDTW, along with code for running all experiments in this paper. [1]

2. SYSTEM DESIGN

In this section we describe the FlexDTW algorithm in detail. To make it clear how FlexDTW relates to previous work, we begin with a brief overview of DTW and subsequence DTW.

2.1 DTW and Subsequence DTW

Standard DTW estimates the alignment between two feature sequences $x_0, x_1, \ldots, x_{N-1}$ and $y_0, y_1, \ldots, y_{M-1}$. It accomplishes this by using dynamic programming to find the optimal path through a pairwise cost matrix $C \in \mathbb{R}^{N \times M}$ under a set of allowable transitions. DTW assumes that the alignment path begins at (0,0) and ends at ($N-1$, $M-1$) in the cost matrix. Subsequence DTW is a variant of DTW that finds the optimal alignment between a query sequence $x_0, x_1, \ldots, x_{N-1}$ and any subsequence within a (typically longer) reference sequence $y_0, y_1, \ldots, y_{M-1}$. Subsequence DTW assumes that the alignment path includes the entire query sequence but can begin and end anywhere in the reference sequence.

2.2 FlexDTW: Algorithm

FlexDTW is a variant of DTW that seeks to handle a much wider range of boundary conditions. It is designed to handle the boundary conditions of standard DTW, subsequence DTW, as well as many other conditions that are not handled by DTW or subsequence DTW. We first give an overview of the boundary conditions that FlexDTW is designed to handle, describe the main challenge in allowing flexible boundary conditions, introduce a key insight, and then explain the algorithm in detail.

Boundary conditions. Figure 1 shows an overview of the boundary conditions that FlexDTW is designed to handle. In the given figure, the alignment path may begin anywhere along the left edge or bottom edge of the cost matrix, and the alignment path can end anywhere along the top edge or right edge. [2] Note that the resulting set of allowable alignment paths is a superset that contains all allowable DTW paths and all allowable subsequence DTW paths, in addition to many other types of alignment paths.

Challenge. The main challenge in allowing such flexible boundary conditions is normalization. Because the set of allowable paths has such an enormous variation in path length, one must use a normalized path cost to fairly compare one alignment path with another. (Otherwise, the path with minimum cumulative path cost will simply be the alignment path with fewest elements.) This means that our metric for comparing different alignment paths must normalize the cumulative path cost by some measure of alignment path length. To determine the length of an alignment path ending at position (i,j), we could simply follow the backpointers in the backtrace matrix, but this introduces an impractically high computational overhead to the algorithm.

Key insight. The key insight with FlexDTW is that the Manhattan length of an alignment path does not require knowing what the actual alignment path is. Assuming that the alignment path is monotonically non-decreasing (as is the case with DTW), computing the Manhattan length of

[1] Code can be found at `https://github.com/anonymized/`.

[2] We exclude a buffer region near the top left and bottom right corners to avoid short, degenerate paths, as will be explained later.

an alignment path only requires knowing the starting point and ending point of the path. The starting location of any optimal alignment path can be computed recursively with minimal computational overhead and simply stored as an additional piece of information (similar to the backtrace information). Having the starting location of all optimal alignment paths allows us to efficiently calculate normalized path costs without having to perform any backtracking. We can then compare the goodness of alignment paths by comparing their path cost per Manhattan block.

Algorithm. We now describe the FlexDTW algorithm for aligning two feature sequences x_0, x_1,..., x_{N-1} and y_0, y_1, ..., y_{M-1}. Similar to DTW, one must specify a set of allowable transitions and corresponding transition weights. In addition, there is one hyperparameter `buffer` that specifies a minimum allowable path length, which helps to avoid short, degenerate alignment paths. The algorithm consists of five steps, which are described below.

The first step is to compute a pairwise cost matrix $C \in \mathbb{R}^{N \times M}$, where each element $C[i, j]$ indicates the distance between x_i and y_j under some distance metric.

The second step is to initialize three matrices: a cumulative cost matrix $D \in \mathbb{R}^{N \times M}$, a backtrace matrix $B \in \mathbb{Z}^{N \times M}$, and a starting point matrix $S \in \mathbb{Z}^{N \times M}$. In order to allow alignment paths to begin anywhere in either sequence without penalty, we initialize $D[0, j] = C[0, j]$, $j = 0, 1, \ldots, M-1$ and $D[i, 0] = C[i, 0]$, $i = 0, 1, \ldots, N-1$. We also initialize S for all valid starting points for alignment paths. Since the starting locations are all of the form $(0, j)$ or $(i, 0)$, we can efficiently encode the starting locations as a single integer, where positive integers indicate a starting location $(0, j)$ and negative integers indicate a starting location $(i, 0)$. This reduces the memory overhead of matrix S. Accordingly, we initialize $S[0, j] = j$, $j = 0, 1, \ldots, M-1$ and $S[i, 0] = -i$, $i = 0, 1, \ldots, N-1$.

The third step is to compute the elements in D, B, and S using dynamic programming. For a given set of allowable transitions $\{t_1, t_2, t_3\}$ (assumed to be $\{(1, 1), (1, 2), (2, 1)\}$ in the equation below) and corresponding multiplicative weights w_1, w_2, w_3, the optimal transition $B[i, j]$ can be computed with the following recursive formula:

$$B[i,j] = \underset{k=1,2,3}{\arg\min} \begin{cases} \frac{D[i-1,j-1]+w_1 \cdot C[i,j]}{i+j-|S[i-1,j-1]|} & \text{if } k=1 \\ \frac{D[i-1,j-2]+w_2 \cdot C[i,j]}{i+j-|S[i-1,j-2]|} & \text{if } k=2 \\ \frac{D[i-2,j-1]+w_3 \cdot C[i,j]}{i+j-|S[i-2,j-1]|} & \text{if } k=3 \end{cases} \quad (1)$$

The numerator elements in the equation above are cumulative path costs, and the denominator elements are the Manhattan lengths of each candidate path. Once the best transition has been determined, the value of $D[i, j]$ can be updated as:

$$D[i,j] = \begin{cases} D[i-1,j-1] + w_1 \cdot C[i,j] & \text{if } B[i,j]=1 \\ D[i-1,j-2] + w_2 \cdot C[i,j] & \text{if } B[i,j]=2 \\ D[i-2,j-1] + w_3 \cdot C[i,j] & \text{if } B[i,j]=3 \end{cases} \quad (2)$$

Similarly, the value of $S[i, j]$ can be updated as:

$$S[i,j] = \begin{cases} S[i-1,j-1] & \text{if } B[i,j]=1 \\ S[i-1,j-2] & \text{if } B[i,j]=2 \\ S[i-2,j-1] & \text{if } B[i,j]=3 \end{cases} \quad (3)$$

Note that the elements in D still indicate unnormalized path costs (as in DTW), but the decision of which transition is the best is made based on the normalized path cost (i.e. path cost per Manhattan block).

The fourth step is to identify the endpoint of the optimal alignment path. The candidate set of valid ending points is given by $E_{cand} = \{(N-1, j) \mid j = \texttt{buffer}, \ldots, M-1\} \cup \{(i, M-1) \mid i = \texttt{buffer}, \ldots, N-1\}$, which corresponds to any location in the top or right edge in Figure 1. We exclude a user-specified buffer region from the top left and bottom right corners, which ensures that the alignment path is of a certain minimum length. This buffer region helps to prevent the algorithm from selecting short, degenerate alignments paths with low normalized path cost. Given this set of candidate locations, we select the endpoint E_{best} to be

$$E_{best} = \underset{(i,j) \in E_{cand}}{\arg\min} \frac{D[i,j]}{i+j-|S[i,j]|} \quad (4)$$

where the objective function is the path cost per Manhattan block.

The fifth step is to backtrace from the selected endpoint using the backpointers in B until we reach an element $(0, j)$, $j = 0, 1, \ldots, M-1$ (on the bottom edge in Figure 1) or $(i, 0)$, $i = 0, 1, \ldots, N-1$ (on the left edge). The resulting alignment path is the final estimated alignment.

2.3 FlexDTW: Hyperparameters

In this subsection we discuss the hyperparameters in FlexDTW and our method for setting them. As mentioned previously, FlexDTW has three kinds of user-defined parameters: a set of allowable transitions, a corresponding set of transition weights, and a buffer hyperparameter that specifies a minimum path length for allowable alignment paths. Note that DTW also requires specifying a set of transitions and transition weights, so FlexDTW has one additional hyperparameter compared to DTW.

Transitions & weights. A typical set of transitions for audio alignment tasks is $\{(1, 1), (1, 2), (2, 1)\}$, which imposes a maximum warping factor of 2. This set is usually preferred to sets that include (0, 1) and (1, 0) transitions, since these transitions can lead to degenerate alignments. We will use this set of allowable transitions throughout this paper, unless otherwise noted. The associated transition weights can be set in many different ways. In FlexDTW, there is one particular setting of transition weights that is of theoretical interest: $w_1 = 2$, $w_2 = 3$, $w_3 = 3$. This setting weights each transition according to its Manhattan distance. Note that in standard DTW (where the alignment path is assumed to start at (0, 0) and end at $(N-1, M-1)$), every allowable alignment path has the same Manhattan

Piece	Files	mean	std	min	max
Opus 17, No 4	64	259.7	32.5	194.4	409.6
Opus 24, No 2	64	137.5	13.9	109.6	180.0
Opus 30, No 2	34	85.0	9.2	68.0	99.0
Opus 63, No 3	88	129.0	13.4	96.2	162.9
Opus 68, No 3	51	101.1	19.4	71.8	164.8

Table 1. Overview of the original Chopin Mazurka dataset. This is used as the source data to generate the benchmark suite. All durations are in seconds.

distance, so this setting effectively treats every path as being equally likely. It is analogous to a maximum likelihood formulation in which all possibilities are treated as equally likely a priori, and selection is made entirely based on the observations. For this reason, we recommend setting the transition weights in FlexDTW as $w_1 = W$, $w_2 = 3$, $w_3 = 3$, where W can be tuned on a validation dataset. $W = 2$ corresponds to a maximum likelihood formulation, and smaller values of W correspond to a bias towards diagonal alignment paths. In our experiments, we use $W = 1.25$, which provided optimal performance on the training set.

Buffer. The purpose of the buffer is to prevent the algorithm from selecting short, degenerate alignment paths that may have low normalized path cost. For example, silence at the end of one sequence may match silence at the beginning of the other sequence, resulting in a very short alignment path with low normalized path cost. The buffer should be interpreted as the minimum length along one sequence that an alignment path must match in order to be considered a valid path. This could simply be set manually based on knowledge of the task or data. In our case, however, our suite of benchmarks spans such a wide range of sequence lengths and alignment path lengths that a single setting is not ideal. Therefore, we determined the buffer hyperparameter in a data-dependent way for every individual query based on two considerations. First, when one sequence is much longer than the other sequence, the desired behavior is probably a subsequence alignment. In this case, we want the entire shorter (query) sequence to be matched. Second, when the two sequences are approximately the same length, much more flexibility can be afforded and an intuitive parameter is to define the minimum percentage of either sequence that must be matched. Putting these two considerations together, we recommend setting the buffer hyperparameter in the following way: for aligning two sequences of length L_1 and L_2, set $\texttt{buffer} = min(L_1, L_2) \cdot (1 - (1 - \beta) * \frac{min(L_1,L_2)}{max(L_1,L_2)})$. This sets the buffer to a fraction of the shorter sequence length, where the fraction is close to 1 when L_1 and L_2 are very different (i.e. the subsequence case) and close to β when L_1 and L_2 are approximately the same. β can thus be interpreted as the minimum fraction of either sequence that must be matched when both sequences are equal in length. We tuned β on the training set and found $\beta = 0.1$ to work well.

3. EXPERIMENTAL SETUP

In this section we describe the suite of 16 benchmarks that we use to characterize the performance of alignment algorithms under a variety of boundary conditions.

Original data. The raw source material for our benchmarks comes from the Chopin Mazurka dataset [31]. This dataset consists of numerous historic recordings of five different Chopin Mazurkas, along with beat-level annotations of each recording. All of the recordings for two of the Mazurkas (Opus 17 No. 4 and Opus 63 No. 3) were set apart for training and development, and the recordings from the other three Mazurkas were set apart for testing. Table 1 provides an overview of the dataset.

Evaluation. To evaluate alignment performance, we consider every pair of recordings of the same Mazurka. This results in $\binom{64}{2} + \binom{88}{2} = 5844$ training pairs and $\binom{64}{2} + \binom{34}{2} + \binom{51}{2} = 3852$ testing pairs. For each pair of recordings A and B, we compare the estimated alignment path against the ground truth beat timestamps in the following manner. At each ground truth beat timestamp in recording A, we compute the alignment error between the estimated corresponding timestamp in recording B (based on the predicted alignment path) and the ground truth corresponding timestamp in recording B (based on the beat annotations). We report aggregate alignment performance as an error rate indicating the percentage of alignments that have an alignment error greater than a fixed error tolerance.

Modifications: Overview. We generated synthetically modified versions of the Mazurka dataset in order to simulate a variety of boundary conditions. Each modified version of the Mazurka dataset contains the exact same number of recordings, but each recording has been modified to study a particular boundary condition. Thus, the number of training pairs and testing pairs is the same as in the original benchmark, but the audio data and corresponding annotations have been modified appropriately. Each benchmark is evaluated as described above. Below, we describe how we constructed each of the 16 benchmarks.

Full Match. The full match benchmark is the Mazurka dataset in its original unmodified form. This boundary condition assumes that both recordings start and end at the beginning and end of the piece. In Figure 1, this corresponds to an alignment path that starts in the lower left corner and ends in the upper right corner.

Subsequence. The subsequence benchmark assumes that one recording matches a subsequence in the other recording. For every pair of recordings A and B, a randomly sampled L-second interval within recording A is selected and aligned against the entirety of recording B. We construct three separate subsequence benchmarks with $L = 20, 30, 40$.

Partial Start. The partial start benchmark assumes that both recordings start together but that one recording ends early (e.g. only contains one movement). For every pair of recordings A and B, we randomly sample a number in the interval $[0.55, 0.75]$, select that percentage of recording A (starting from the beginning), and align the fragment of recording A against the entirety of recording B.

Partial End. The partial end benchmark assumes that both recordings end together but that one recording starts part way through the piece. For every pair of recordings A and B, we randomly sample a number in the interval $[0.55, 0.75]$, select that percentage of recording A at the end (i.e. starting in the middle of the recording and extending until the end), and then align the fragment of recording A against the entirety of recording B.

Partial Overlap. The partial overlap benchmark assumes that both recordings have some temporal overlap, but that one recording contains extra content before the region of overlap and the other recording contains extra content after the region of overlap. For every pair of recordings A and B, we (a) randomly sample a number in $[0.55, 0.75]$ and select that percentage of recording A starting from the beginning, (b) randomly sample a different number in $[0.55, 0.75]$ and select that percentage of recording B at the end, and then (c) align the fragment of A against the fragment of B.

Pre. The pre benchmark assumes that both recordings contain the entire piece but that one recording has a period of silence at the beginning. For every pair of recordings A and B, we prepend L seconds of silence to recording A and align it to the entirety of recording B. We construct three separate pre benchmarks with $L = 5, 10, 20$.

Post. The post benchmark assumes that both recordings contain the entire piece but that one recording has a period of silence after the piece ends. For every pair of recordings A and B, we append L seconds of silence to recording A and align it against the entirety of recording B. We construct three separate post benchmarks with $L = 5, 10, 20$.

Pre-Post. The pre-post benchmark assumes that both recordings contain the entire piece, but that one recording contains extra silence at the beginning and the other recording contains extra silence at the end. For every pair of recordings A and B, we prepend L seconds of silence to recording A, append L seconds to recording B, and then align the two recordings. We construct three separate pre-post benchmarks with $L = 5, 10, 20$.

4. RESULTS

We report experimental results with FlexDTW and several standard alignment algorithms:

- DTW1: Standard DTW with transitions of (1, 1), (1, 2), (2, 1) and corresponding weights 2, 3, 3.
- DTW2: Standard DTW with transitions of (1, 1), (1, 2), (2, 1) and corresponding weights 1, 1, 1.
- DTW3: Standard DTW with transitions of (1, 1), (1, 2), (2, 1) and corresponding weights 1, 2, 2.
- SubseqDTW1: Subsequence DTW with (query, reference) transitions of (1, 1), (1, 2), (2, 1) and corresponding weights 1, 1, 2.
- SubseqDTW2: Subsequence DTW with (query, reference) transitions of (1, 1), (1, 2), (2, 1) and corresponding weights 2, 3, 3.
- SubseqDTW3: Subsequence DTW with (query, reference) transitions of (1, 1), (1, 2), (2, 1) and corresponding weights 1, 2, 2.
- NWTW: A variant of DTW proposed in [21] that allows skip transitions (0, 1) and (1, 0), in addition to the usual (1, 1), (1, 2), (2, 1) transitions. The skip transitions incur a fixed penalty cost γ, which is a hyperparameter that we tuned on the training data.

We assessed the performance of a larger set of DTW versions (with different sets of allowable transitions and corresponding transition weights), but we only include the 3 versions with best performance to avoid overcluttering Figure 2. Of particular note, we did experiment with DTW versions that had (0, 1) and (1, 0) transitions, but always found those versions to perform much worse. Likewise, we considered other versions of subsequence DTW but only include the top 3 versions in Figure 2. The subsequence DTW systems are unique in that they are not symmetric. For these systems, we always assume that the alignment is trying to match the shorter recording against a subsequence in the longer recording. Note that all of the systems above can be used with any feature representation and distance metric. For simplicity, we use standard chroma features (as computed with default parameters in librosa) and a cosine distance metric for all systems.

Figure 2 compares the performance of FlexDTW and the above algorithms on our benchmark suite. For each system, we fixed the hyperparameter settings and evaluated its performance across all 16 benchmarks. Each panel in Figure 2 corresponds to one of the 16 benchmarks, and the different colored bars show the error rate at 200ms tolerance for different systems. On top of each colored bar, we have also overlaid two black horizontal bars indicating the error rate at 100ms tolerance (above) and at 500ms tolerance (below).

There are two things to notice about the results in Figure 2. First, the seven baseline systems only handle a subset of boundary conditions. In other words, each of the baseline systems performs well on certain benchmarks and very poorly on other benchmarks. For example, the DTW systems perform well on the fully matching benchmark (for which they are designed), but they perform terribly on the subsequence benchmarks and perform worse and worse as more silence is prepended or appended to either recording. Likewise, the subsequence DTW systems perform well on the subsequence benchmarks, but they fail on the partial overlap benchmark and have only moderate performance on the pre, post, and pre-post benchmarks. NWTW has strong performance across most benchmarks but fails completely on the subsequence and partial overlap benchmarks. All of the baseline systems completely fail on the partial overlap benchmark, since none are designed to handle that boundary condition. Second, FlexDTW has consistently strong performance across all 16 benchmarks. On all benchmarks, it has a performance that is comparable to or better than the best-performing baseline system. On the partial overlap benchmark, it is the only system that has strong performance, with an error rate that is comparable to its performance on the other benchmarks. These results demonstrate its flexibility in handling a wide range of boundary conditions.

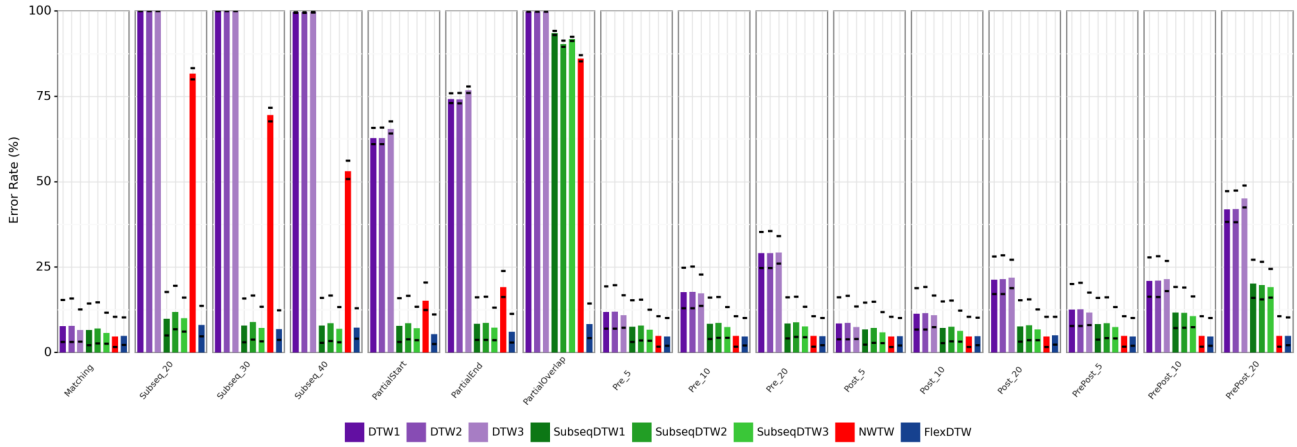


Figure 2. Performance of alignment algorithms on the 16 boundary conditions in our benchmark suite. Colored bars indicate error rate at 200ms error tolerance, and the horizontal bars indicate error rates at 100ms (above) and 500ms (below). Error rates greater than 50% are not shown.

System	1k	2k	5k	10k	20k	50k
DTW	.033	0.14	0.87	3.5	13.8	87.3
SubseqDTW	0.04	0.15	0.96	3.82	15.3	96.8
NWTW	.037	0.16	0.97	3.93	15.8	101.1
FlexDTW	.038	0.16	1.05	4.21	16.9	111.1

Table 2. Average runtime to process a cost matrix of size $N \times N$. Columns indicate different sizes N, and rows indicate different systems. Each reported number is an average over 10 trials, and times are expressed in seconds.

5. ANALYSIS

In this section we conduct several analyses to provide deeper insight into FlexDTW.

Table 2 compares the runtime of FlexDTW and the baseline alignment systems. We measured how long each alignment algorithm took to process a cost matrix of size $N \times N$, where N ranges from 1k to 50k. Each number in the table is an average over 10 trials. FlexDTW and NWTW were implemented in python with numba acceleration, and we used the librosa implementation for DTW and subsequence DTW (also using numba acceleration). All experiments were run on an Intel Xeon 2.40 GHz CPU. For longer sequence lengths, we can see that FlexDTW incurs a 20-25% runtime overhead compared DTW and a 10-15% runtime overhead compared to subsequence DTW. This overhead comes primarily from needing to perform a floating-point division to evaluate every candidate path.

Another drawback of FlexDTW is the additional memory overhead of storing S. We can estimate the memory overhead in the following manner. DTW requires allocating three matrices: the pairwise cost matrix $C \in \mathbb{R}^{N \times M}$, the cumulative cost matrix $D \in \mathbb{R}^{N \times M}$, and the backtrace matrix $B \in \mathbb{Z}^{N \times M}$. Assuming that C and D are matrices of 64-bit floating point numbers and B is a matrix of 8-bit unsigned integers, the total memory cost is $8NM + 8NM + 1NM = 17NM$ bytes. FlexDTW requires allocating an additional matrix S for storing the starting point locations. If the two sequence lengths are less than $2^{15} = 32768$, then S can be stored as a matrix of 16-bit integers, resulting in an extra memory overhead of $2NM$. If either sequence length is greater than 32768, then S must be stored as a matrix of 64-bit integers, resulting in an extra memory overhead of $4NM$. In summary, the memory overhead is $\frac{2NM}{17NM} \approx 12\%$ for sequence lengths less than 32768 and $\frac{4NM}{17NM} \approx 24\%$ for longer sequences.

We also investigated and identified two main failure modes of FlexDTW. The first failure mode occurs when there is extreme time warping between the two recordings. Because the (1, 1) transition is penalized proportionally less than the (2, 1) or (1, 2) transitions, the algorithm will sometimes take a “shortcut" of (1, 1) transitions to/from an edge of the cost matrix at the beginning or end of the alignment path. The second failure mode occurs when alternate matching paths are selected. For example, in the Mazurka Opus 17 No. 4, the first four measures and the last four measures match, which creates an additional matching alignment path under the flexible boundary conditions of FlexDTW.

6. CONCLUSION

We have introduced a time warping algorithm called FlexDTW that is designed to handle a wide range of boundary conditions. We artificially generate a suite of 16 benchmarks based on the Chopin Mazurka dataset, which characterizes alignment performance in a variety of boundary conditions. In all 16 boundary conditions, FlexDTW has strong performance that is as good or better than a set of widely used alignment algorithms. Compared to the librosa implementation of DTW and subsequence DTW, FlexDTW incurs a runtime overhead of 10-25% and a memory overhead of 12% for sequences less than length 2^{15} and 24% for longer sequences.

7. ACKNOWLEDGMENTS

We would like to thank Kate Perkins for her contributions in the early stages of this project. This material is based upon work supported by the National Science Foundation under Grant No. 2144050.

8. REFERENCES

[1] Y. Zhang and J. Glass, "An inner-product lower-bound estimate for dynamic time warping," in *Proc. of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2011, pp. 5660–5663.

[2] E. Keogh, L. Wei, X. Xi, M. Vlachos, S.-H. Lee, and P. Protopapas, "Supporting exact indexing of arbitrarily rotated shapes and periodic time series under euclidean and warping distance measures," *VLDB Journal*, vol. 18, no. 3, pp. 611–630, 2009.

[3] T. Rakthanmanon, B. Campana, A. Mueen, G. Batista, B. Westover, Q. Zhu, J. Zakaria, and E. Keogh, "Searching and mining trillions of time series subsequences under dynamic time warping," in *Proc. of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2012, pp. 262–270.

[4] J. Li and Y. Wang, "EA DTW: Early abandon to accelerate exactly warping matching of time series," in *International Conference on Intelligent Systems and Knowledge Engineering*, 2007.

[5] A. Shabib, A. Narang, C. P. Niddodi, M. Das, R. Pradeep, V. Shenoy, P. Auradkar, T. Vignesh, and D. Sitaram, "Parallelization of searching and mining time series data using dynamic time warping," in *IEEE International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2015, pp. 343–348.

[6] S. Srikanthan, A. Kumar, and R. Gupta, "Implementing the dynamic time warping algorithm in multithreaded environments for real time and unsupervised pattern discovery," in *International Conference on Computer and Communication Technology*, 2011, pp. 394–398.

[7] Z. Wang, S. Huang, L. Wang, H. Li, Y. Wang, and H. Yang, "Accelerating subsequence similarity search based on dynamic time warping distance with FPGA," in *Proceedings of the ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, 2013, pp. 53–62.

[8] D. Sart, A. Mueen, W. Najjar, E. Keogh, and V. Niennattrakul, "Accelerating dynamic time warping subsequence search with GPUs and FPGAs," in *IEEE International Conference on Data Mining*, 2010, pp. 1001–1006.

[9] C. J. Tralie and E. Dempsey, "Exact, parallelizable dynamic time warping alignment with linear memory," in *Proc. of the International Society for Music Information Retrieval Conference (ISMIR)*, 2020, pp. 462–469.

[10] R. Tavenard and L. Amsaleg, "Improving the efficiency of traditional DTW accelerators," *Knowledge and Information Systems*, vol. 42, no. 1, pp. 215–243, 2015.

[11] Y. Zhang and J. Glass, "A piecewise aggregate approximation lower-bound estimate for posteriorgram-based dynamic time warping," in *Proc. of the Annual Conference of the International Speech Communication Association*, 2011.

[12] A. Lods, S. Malinowski, R. Tavenard, and L. Amsaleg, "Learning DTW-preserving shapelets," in *International Symposium on Intelligent Data Analysis*, 2017, pp. 198–209.

[13] G. Nagendar and C. Jawahar, "Efficient word image retrieval using fast DTW distance," in *Proc. of the IEEE International Conference on Document Analysis and Recognition (ICDAR)*, 2015, pp. 876–880.

[14] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, no. 1, pp. 43–49, 1978.

[15] F. Itakura, "Minimum prediction residual principle applied to speech recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 23, no. 1, pp. 67–72, 1975.

[16] M. Müller, H. Mattes, and F. Kurth, "An efficient multiscale approach to audio synchronization," in *Proc. of the International Society for Music Information Retrieval Conference (ISMIR)*, 2006, pp. 192–197.

[17] S. Salvador and P. Chan, "FastDTW: Toward accurate dynamic time warping in linear time and space," in *Proc. of the KDD Workshop on Mining Temporal and Sequential Data*, 2004.

[18] T. Tsai, "Segmental DTW: A parallelizable alternative to dynamic time warping," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 106–110.

[19] T. Prätzlich, J. Driedger, and M. Müller, "Memory-restricted multiscale dynamic time warping," in *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2016, pp. 569–573.

[20] C. Fremerey, M. Müller, and M. Clausen, "Handling repeats and jumps in score-performance synchronization," in *Proc. of the International Society for Music Information Retrieval Conference (ISMIR)*, 2010, pp. 243–248.

[21] M. Grachten, M. Gasser, A. Arzt, and G. Widmer, "Automatic alignment of music performances with structural differences," in *Proc. of the International Society for Music Information Retrieval Conference (ISMIR)*, 2013.

[22] M. Shan and T. Tsai, "Improved handling of repeats and jumps in audio-sheet image synchronization," in *Proc. of the International Society for Music Information Retrieval Conference (ISMIR)*, 2020, pp. 62–69.

[23] S. Dixon, "Live tracking of musical performances using on-line time warping," in *Proc. of the International Conference on Digital Audio Effects*, 2005, pp. 92–97.

[24] S. Dixon and G. Widmer, "MATCH: A music alignment tool chest," in *Proc. of the International Society for Music Information Retrieval Conference (ISMIR)*, 2005, pp. 492–497.

[25] R. Macrae and S. Dixon, "Accurate real-time windowed time warping," in *Proc. of the International Society for Music Information Retrieval Conference (ISMIR)*, 2010, pp. 423–428.

[26] M. Müller and D. Appelt, "Path-constrained partial music synchronization," in *Proc. of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2008, pp. 65–68.

[27] M. Müller and S. Ewert, "Joint structure analysis with applications to music annotation and synchronization," in *Proc. of the International Society for Music Information Retrieval Conference (ISMIR)*, 2008, pp. 389–394.

[28] S. Wang, S. Ewert, and S. Dixon, "Robust and efficient joint alignment of multiple musical performances," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 11, pp. 2132–2145, 2016.

[29] S. Waloschek and A. Hadjakos, "Driftin' down the scale: Dynamic time warping in the presence of pitch drift and transpositions," in *Proc. of the International Society for Music Information Retrieval Conference (ISMIR)*, 2018, pp. 630–636.

[30] D. Yang, K. Ji, and T. Tsai, "Aligning unsynchronized part recordings to a full mix using iterative subtractive alignment," in *Proc. of the International Society for Music Information Retrieval Conference (ISMIR)*, 2021, pp. 810–817.

[31] C. Sapp, "Hybrid numeric/rank similarity metrics for musical performance analysis," in *Proc. of the International Conference for Music Information Retrieval (ISMIR)*, 2008, pp. 501–506.

MODELING BENDS IN POPULAR MUSIC GUITAR TABLATURES

Alexandre D'Hooge **Louis Bigo** **Ken Déguernel**
Univ. Lille, CNRS, Centrale Lille, UMR 9189 CRIStAL, F-59000 Lille, France
alexandre.dhooge@algomus.fr

ABSTRACT

Tablature notation is widely used in popular music to transcribe and share guitar musical content. As a complement to standard score notation, tablatures transcribe performance gesture information including finger positions and a variety of guitar-specific playing techniques such as *slides*, *hammer-on/pull-off* or *bends*. This paper focuses on bends, which enable to progressively shift the pitch of a note, therefore circumventing physical limitations of the discrete fretted fingerboard. In this paper, we propose a set of 25 high-level features, computed for each note of the tablature, to study how bend occurrences can be predicted from their past and future short-term context. Experiments are performed on a corpus of 932 *lead guitar* tablatures of popular music and show that a decision tree successfully predicts bend occurrences with an F_1 score of 0.71 and a limited amount of false positive predictions, demonstrating promising applications to assist the arrangement of non-guitar music into guitar tablatures.

1. INTRODUCTION

The guitar, whether acoustic or electric, is (in most cases) a fretted instrument which enforces the playing of discrete pitch values on a chromatic scale. This constraint can be beneficial, as it limits the risk of playing out-of-tune. However, it also prevents microtonal experiments or continuous pitch shifts, which can be a powerful means of musical expressiveness. To overcome this limitation, guitarists can alter the string tension with their fretting hand [1] to reach a completely new pitch, up to several semitones higher. This technique is called string bending, or just *bends,* and is an important part of guitar playing in blues, rock or pop music. Even though bending a string can only increase the pitch of a note, a variety of bend types are used. While guitar players mostly agree over the existing variations, the names used can differ. In this paper, we consider the five bend types described in Gomez's work [2]: *Basic upward*, *Held*, *Reverse*, *Up & Down*, and *Complex* bends for bends that do not belong to any of the previous categories.

Guitar tablatures, compared to standard staff notation, include fingering information on where to play a note with a given pitch on the fretboard. Tablatures are therefore an effective notation to display playing techniques, the position of the fretting hand being critical to know how to perform a bend. Examples of bends are shown in a tablature in Figure 1, different bend types are represented with differently shaped arrows. Knowing where and when to use bends is part of the idioms of guitar pop music and an important part of learning this style. However, the variety of bend types can make it difficult to choose how to use them. For instance, a guitarist playing a score that was composed for another instrument may want to add expressiveness using bends, and could need help on deciding which notes to bend and which bend to use. Moreover, a tool suggesting bends could also improve the quality of online tablatures that sometimes do not have guitar techniques annotations.

Could bends be inferred from musical context? Are they correlated to other elements of a score or a tablature such as pitch, rhythm, or hand position? In this paper, we propose to model bent notes and their context through temporal, pitch and tablature related information. Such a representation could be used to predict which notes are bent from a score or a tablature. Our contribution is three-fold: (1) we define a set of high-level features to model bent notes and their context, (2) we conduct a statistical study on bends based on those features, and (3) we propose a method for predicting bends from tablatures.

The rest of this paper is organized as follows: after discussing related works in Section 2, we introduce our modeling choices in Section 3. The dataset and its statistical study are presented in Section 4. We introduce our prediction algorithm in Section 5 and reflect on this work in Section 6.

2. RELATED WORKS

Guitar tablatures are often studied in the audio realm for guitar music transcription [3, 4]. In particular, automatic transcription of playing techniques from audio has also been studied, for instance with hexaphonic microphones [5] or deep learning techniques [6]. Of course, this task is not restricted to guitar and has been applied to other instruments such as the Chinese *guqin*, which also features several string bending techniques [7].

Another related task is the generation of tablatures, which has been studied increasingly in the last decade. Playing techniques can be included in generation frameworks to obtain results closer to actual human performance. The Transformer-XL model presented in [8, 9] can for instance generate tokens representing playing tech-

© A. D'Hooge, L. Bigo and K. Déguernel. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).
Attribution: A. D'Hooge, L. Bigo and K. Déguernel, "Modeling bends in popular music guitar tablatures", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

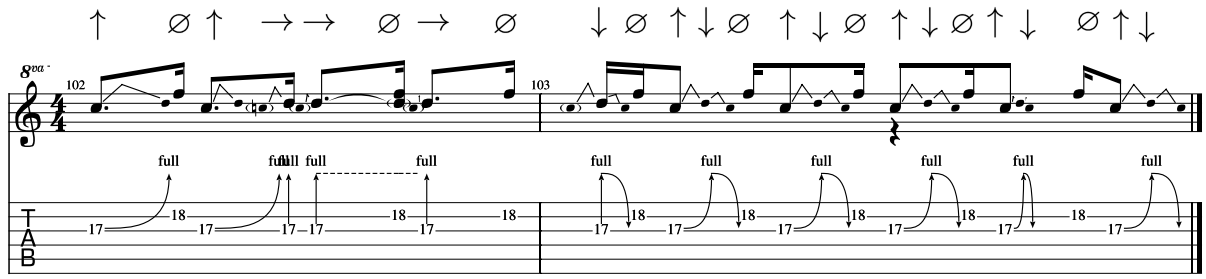


Figure 1: Excerpt from Lynyrd Skynyrd's *Free Bird* solo. In the first measure, the first two bends are *basic upward bends*, the remaining ones are *held bends*. In the second measure, the first bend is a *reverse bend*, and the other ones are *up and down bends*. While all bends of this example are whole tones (denoted by *full*), the amplitude of a bend can vary. String movements are labeled above, according to the representation presented in Section 3. An audio rendering of this excerpt is available on the **accompanying repository**.

niques. Chen et al. [10] also consider *Technique* events, but only for picking-hand techniques, since the generation focuses on fingerstyle guitar. McVicar et al. present in [11] a method to generate tablatures of guitar solo phrases, and fretting-hand techniques are added in a post-processing step. Beyond the case of guitar, playing techniques modeling includes symbolic piano music generation with sustain pedal information [12].

Another large part of guitar tablature MIR research focuses on fingering prediction, where a model is designed to predict the pitch/fret combination for each note of a musical score. This problem has been studied with path-finding algorithms [13] and minimax techniques [14]. More recently, Cheung et al. [15] used a deep learning model to generate fingering annotations, though not for guitar but violin. Those instruments nonetheless share some properties, and a study of fingering prediction on string instruments like the violin or the guitar, compared to other instruments, can be found in [16]. However, the task of using symbolic music to study occurrences of playing techniques is rarely studied. Xie and Li [17] propose to predict playing techniques as a tagging task on symbolic bamboo flute music but, to the best of our knowledge, no such work has been proposed for guitar music.

3. MODELING BENDS IN TABLATURE

To formulate bend prediction as a machine learning task, we adopt a representation for bent notes consisting of four different labels. We also need to pre-process our data to obtain bend-less scores, and musical features that could be used to train a machine learning model. Those considerations are presented hereafter.

3.1 Labeling

In the introduction, we presented the different types of bends that can be encountered in guitar tablatures. Based on this taxonomy, we define 4 labels that represent the motion and current state of the played string:

- $\varnothing$ — the string is not bent;
- $\uparrow$ — the string is bent, causing the pitch to go up;
- $\rightarrow$ — the string was bent previously and is plucked again in that state. The pitch is constant, but is not the one expected from the note's string/fret position;
- $\downarrow$ — the string was bent and is released, making the pitch go down accordingly.

We define those labels to circumvent the issue with *up & down* and *complex bends* that are not transcribed consistently. Labeling notes with the associated string movements therefore permits representing bends accurately, without loss of generality. Using these labels, all non-bent notes will be labeled $\varnothing$, *basic upward bends* are labeled $\uparrow$, *held bends* correspond to $\rightarrow$, and *reverse bends* are $\downarrow$. To fit with this representation, *up & down bends* are split into two notes of equal duration, the first one being labeled with $\uparrow$, and the second one with $\downarrow$. An example of such labeling on an actual guitar track is shown on top of Figure 1.

While bends can be of different amplitude, we do not include that information in our labeling, as we do not aim at predicting it in this work but only focus on predicting when bends occur. Similarly, we do not distinguish single notes from chords when predicting bends. This means that when the system predicts the presence of a bend in a chord, it does not specify on which string it occurs. The impact of this simplification is however limited, as bends rarely occur inside chords (12% of bend events are found in chords in our corpus).

3.2 Deriving a bend-less score simplification

Since our goal is to predict whether a note is played with a bend, we must start from a *simplified* tablature that does not have any bend information. Removing bend annotations from a tablature is however not a straightforward task since this technique affects the pitch of the performed note. We design the following procedure when translating bent notes from the fret/string space to the pitch space:

- If a note is labeled by $\varnothing$, its pitch is directly obtained from the string/fret combination;

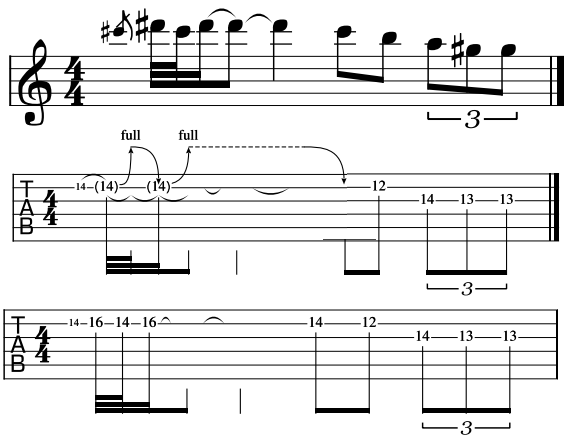


Figure 2: Excerpt of *Watermelon in Easter Hay*, Frank Zappa, as transcribed [18] by Steve Vai in standard notation (top). Below are two possible tablature representations of this excerpt with (middle), or without (bottom) bends.

- Otherwise, the pitch is the one of the bend arrival note. In particular:
  - if the label is ↑ or → , the arrival pitch is the pitch of the string/fret combination, to which the bend amplitude is added;
  - if the label is ↓ , the arrival pitch is the one corresponding to the string/fret position, because the string is released to its default state.

This approach is the one chosen by Steve Vai when transcribing Frank Zappa's melodies to standard notation, as we illustrate in Figure 2. The middle tablature shows how this excerpt is actually played (based on a live performance) and the bottom tablature illustrates how the same excerpt might be played without bends. While it is not an issue in this example, there is an uncertainty regarding where a bent note would be played on the fretboard, without a bend (keeping its destination pitch but losing the technique). A guitar player might indeed choose to play a note on a higher string, if remaining on the same string called for an uncomfortably large hand span. Because deciding arbitrarily of a hand position could introduce bias into our model, we choose not to include any string/fret information concerning the current note in the proposed features for our classification task, as explained hereafter.

3.3 Selected Features

To predict whether a note is bent, we propose an intermediate representation as a set of high-level features, presented in Table 1. Some descriptors focus on the event under scrutiny, while others provide short-term context information, both from the past and the future. Part of the features are derived from standard staff notation and convey *temporal* and pitch information, while others are related to *position* and the tablature space. If the studied event is a chord, the pitch, fret, and string values are averaged over all its constituting notes. While the average might seem an overly simple statistic, experiments with other functionals

Temporal	Duration
	Beat Strength
	Longer than previous
	Shorter than previous
	Same duration as previous
Pitch	Number of notes
	Pitch$^{(j)}$
	Pitch jump$^{(\mathrm{n}\pm k)}$
	Accidentals
	Pitch-class w.r.t scale root
Position	Fret$^{(\mathrm{n}\pm k)}$
	String$^{(\mathrm{n}\pm k)}$
	Fret jump $^{(\mathrm{n}\pm 2)}$
	String jump $^{(\mathrm{n}\pm 2)}$

Table 1: List of high-level features extracted from the note events. Let n denote the current note index, an exponent on a feature tells on which neighbors it is computed. $k \in \{1, 2\}$ because we discard any positional information related to the current note, and $j \in [\![\mathrm{n}-2, \mathrm{n}+2]\!]$. Other features are only computed on n.

such as *min, max* or *standard deviation* did not improve the results. We have discarded any open strings for the fretboard features so that the average fret and string represents the actual position of the fretting hand. Apart from absolute/relative duration values, temporal features include the *beat strength*, which is a value between 0 and 1 suggesting how *strong* the beat of a note is. We obtain this value from the default implementation of the `music21` library [19]. These beat strength values have been designed for Western classical music, and therefore may be debatable for pop and rock music. However, they are mostly used here to represent onset times independently of the time signature, while grouping notes that share rhythmic properties.

In addition to the features on the current note, we extract a context of two past and two future note events, as preliminary experiments did not show any benefits of longer contexts. Additional boolean features are provided to recall if a neighboring note event is missing, when a note is preceded or followed by a whole rest for instance. When a note is missing, all corresponding features are set to 0. From this context, we compute the pitch jump between neighboring notes as well as the string and fret jumps when they are defined, *i.e.,* not with respect to the current note – because we do not know where the guitarist would play the note if they were to bend it. We expect these features to help our algorithm derive the hand position on the fretboard, which would be useful since bends are more likely to occur on certain spots of the fretboard, as will be shown subsection 4.2. Furthermore, we add information about the key signature through the number of accidentals (positive for sharps, negative for flats). From those accidentals, we derive the root note of the corresponding pentatonic minor scale (that scale encompassing much of guitar popular music [20]) and store the position of each note on this scale. For example, one sharp would make the root E, and an A would be numbered 5 since it is 5 semitones above E.

∅	↑	→	↓	Total
123 231	9627	1270	3314	137 442

Table 2: Number of notes per label in our dataset.

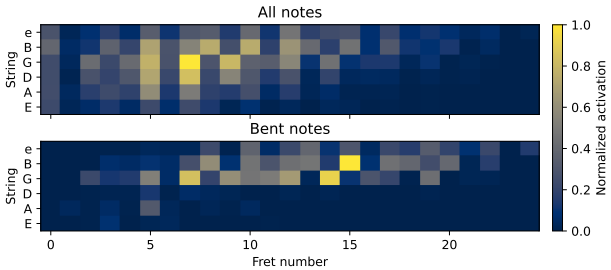


Figure 3: Normalized Heatmaps of all notes (Top) and bent notes (Bottom). The letters refer to the open string pitch in standard tuning with *e* being the high E string.

4. DATASET

4.1 Guitar Tablature Corpus

Our experiments are performed on the proprietary corpus *MySongBook* composed of 2247 guitar tablatures accurately transcribed by professional musicians in the `.gp` GuitarPro format. A subset of 932 tracks estimated as *lead guitar* – totaling more than 130 000 notes – was extracted by applying the classification technique from [21]. Our experiments focus on lead guitar parts, as they were felt to feature heavier use of playing techniques. In contrast with the whole corpus, which includes 2.5 % of bent notes, our lead guitar sub-corpus indeed contains 10% of bent notes, slightly mitigating the observed class imbalance.

Our work is implemented in *Python* and uses `music21` [19] and `scikit-learn` [22] libraries. To foster reproducibility, all our code is made publicly available (parsing of `.gp` files, extraction of features, training, and evaluation of bend classification models). We also release the complete set of features extracted on each note of our corpus, plus corresponding labels at:
`http://algomus.fr/code/`.

4.2 Statistical study

Table 2 reports the distributions of bend labels in our corpus. The distribution of bent notes on the fretboard, compared to all notes, is shown in Figure 3. We observe that most bends occur on the top 3 strings in the middle area of the fretboard. This observation differs from notes in general that are played on all strings, and especially on the two middle ones and around the 7th fret. While it is possible that the obtained heatmaps are biased by an over-representation of certain key signatures in the dataset – 43% of the tracks are in G major/E minor or C major/A minor – this bias should affect both heatmaps equally, so their mutual comparison is still possible. Because bent notes are found on both higher strings and higher frets than all notes, their pitch is similarly higher on average, as it can be observed in Figure 4a.

The distribution of *beat strength* values is shown in Figure 4b. Because most beats and sub-beats in a measure have a beat strength of 0.25 or below, no label is mostly played on strong beats.An interesting result is that ↑ and ↓ labels appear more often on stronger beats than ∅ and →. This apparent correlation of ↑ and ↓ labels with the meter might suggest a link between note expressiveness and accentuation in performance, which would need to be investigated further. In contrast, the → label is most often encountered on *weaker* beats. This observation can be linked to the fact that this technique is often used as a quick repetition of the previous note and will thus be played on the next offbeat, like in Figure 1.

The comparison of the duration of notes with or without bends (Figure 4c) confirms that ↑ and ↓ labels share some essential properties. Both labels have a proportionally higher tendency to be found on notes with longer duration, even though eighth note is the most common duration for all classes. This figure also confirms that ∅ and → classes share some context properties. Figure 4d shows a strong tendency of ↑ labels to appear on notes with longer duration than their predecessor. This further supports the hypothesis that bends could be used to emphasize significant notes in a lead guitar part. That result could also be related to the substantial physical effort required to bend a string on short duration notes. The accompanying code provides interactive computation of the distribution of the other features.

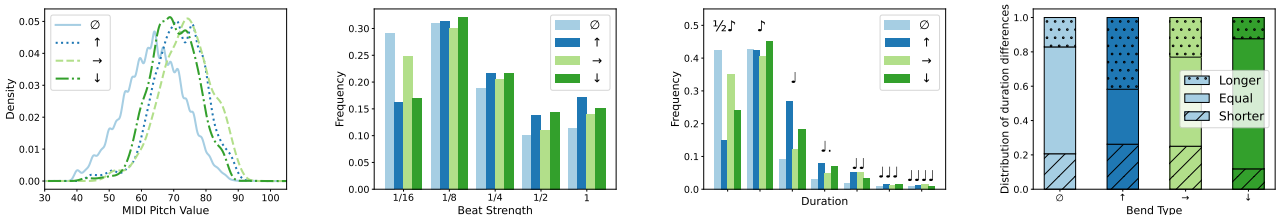
5. CLASSIFICATION RESULTS

A decision tree [23] was trained to predict the bend label of a note from its feature representation. We choose this high-level approach to facilitate the interpretation of the results as well as the analysis of the contribution of the features. In addition to the elaboration of a predictive model, conducting our experiments in an explainable AI framework allows us to improve our understanding of the use of bends in this repertoire. We hope that the use of light models enabled by highly expressive musical representations will also contribute to promoting low energy consumption approaches in machine learning for MIR.

5.1 Model performance

Our classifier is trained on 75% of the dataset, and evaluated on the remaining 25%. To avoid some leakage from the training set to the test set, we ensure the split does not separate notes from a same track. We also ensure that the class imbalance is similar in both sets. Duplicate feature vectors are removed track-wise to avoid overfitting due to repeated riffs/patterns. But, we acknowledge the fact that identical feature vectors can be found in different tracks and thus keep duplicates when found in different files.

The confusion matrix of Figure 5 shows the results of the multi-class classification on the joint prediction of all labels. Because the dataset is highly unbalanced, our model is naturally biased towards the ∅ label. However, it successfully identifies more than half of the ↑ and ↓ la-



(a) Pitch (b) Beat Strength (c) Duration (d) Duration diff. with previous

Figure 4: Distribution of four of the extracted features, normalized per bend class.

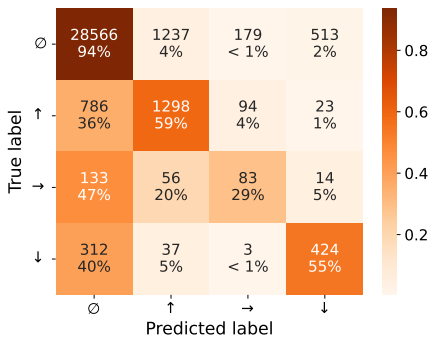


Figure 5: Confusion matrix obtained for classifying each note event to one of the bend class. This matrix was obtained on a split with average performance.

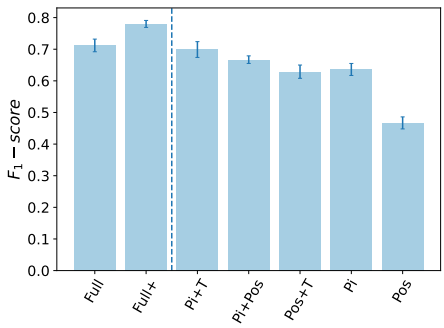


Figure 6: Average F_1-scores from 4 different train/test splits for the binary classification task. The leftmost part shows the performance of the decision tree trained on all features, with (Full+) or without (Full) SMOTE oversampling. The rightmost part corresponds to decision trees trained with a reduced set of features. *T* stands for temporal , *Pi* for pitch and *Pos* for position features.

bels. Samples labeled as → are often misidentified as ↑ but this result still shows, presumably, that the model captures the difference between ∅ and → labels. We tried applying SMOTE oversampling [24] to the training data and observed that it doubles the number of correctly identified → notes and increases the ratio of well-classified ↓ notes by approximately 10%. Nevertheless, ↑ notes *True Positives* (TP) ratio is about the same while the quantity of ↑ notes misidentified as → or ↓ increases. Similarly, TP ratio of ∅ notes drops by 5 p.p., so 2000 more notes are wrongly predicted as bent. Because we observed that bent notes are sparse in guitar tracks, we consider that *precision* is more important than *recall* and do not use any oversampling for the rest of our analysis.

5.2 Feature importance

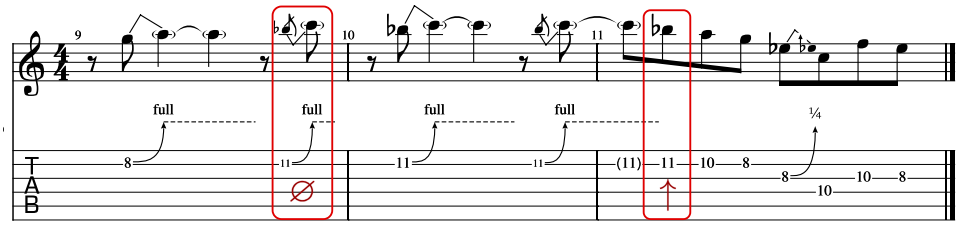
To assess the contribution of each feature, we conduct an *all bend* binary classification experiment where ↑, →, ↓ are merged into a single class, versus the ∅ class. Table 3 shows the importance of the eight most contributing features, computed using the random feature permutation technique introduced in [25] and monitoring its impact on the F_1 score of our model. Temporal and pitch features appear to have a higher impact on classification than position-related features, an observation confirmed by training the binary classifier on selected subsets of features. The results in Figure 6 confirm the dominant influence of pitch features. However, adding gesture and temporal information noticeably improve the results. This result suggests that, while fret context contributes to induce bent notes, a large part of the prediction can be done from the strict musical content as notated in musical scores.

6. PREDICTION ANALYSIS

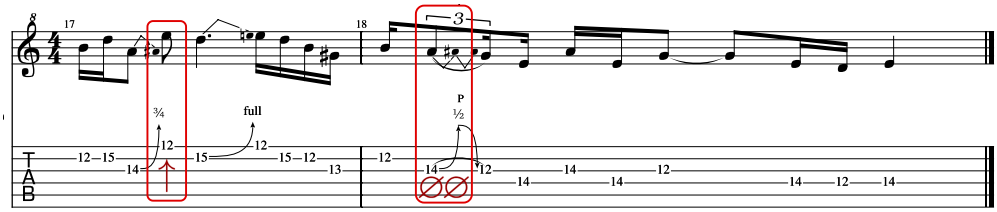
In addition to the quantitative results presented in the last section, we present a qualitative analysis of selected predictions. Figure 7a shows one bent note wrongly identified as ∅ and, conversely, one non-bent note identified as ↑ . Following the decision path provided by the decision tree, we can gain some insight on what feature differences have

Feature	Importance
Pitch	0.20
Pitch jump$^{(n+1)}$	0.17
Pitch jump$^{(n-1)}$	0.16
Duration	0.14
Same dur. as previous	0.07
Fret jump$^{(n-2)}$	0.07
String$^{(n+1)}$	0.05
Pitch$^{(n+1)}$	0.05

Table 3: Feature importance of the 8 most significant features for the decision tree. Standard deviation of any feature importance is never above 0.005.



(a) Excerpt from *Highway Star*, Deep Purple.



(b) Excerpt from *Jailbreak*, AC/DC.

Figure 7: Examples of predictions obtained with our Full Tree model on two different excerpts. Labels shown represent the predicted label for the current note. Only wrong predictions are shown for clarity. All other notes are labeled correctly.

caused those wrong predictions. Both notes actually have more than half their decision path in common and split on their Pitch jump$^{(n+2)}$ value, suggesting that the first discrepancy was due to future context. In particular, the second false prediction did not use any features related to past context. This might also explain this error because the pitch could be obtained by bending on the 10th fret by one semitone – an information that could be derived from future context – but continuity with the previous notes called for playing the note without bend on the 11th fret – an information that should have been derived from past context. Another observation is that, in spite of similar context, the second bent note was misidentified whereas the fourth bent note was not. While those two notes look very similar at first glance, the latter has a longer duration because it is tied to the following eighth note, which illustrates the importance of the *duration* featureAn analysis of the decision paths indeed shows a divergence from the second decision rule, based on that feature. This highlights the presumably strong influence of rhythm in the classification of the first four bent notes, which bypasses pitch features.

Figure 7b also shows a regular note wrongly tagged with a ↑ label. The decision path for this prediction does not consider any feature related to the next note. It does however use many features concerning the second next note, which was correctly classified as ∅ , most likely because of its lower duration. The lack of information about the current note's position was probably critical in that case. The second error on that tablature is an *up & down bend* that was not identified, probably because of the low duration of the involved notes. Nevertheless, this example suggests that our method to obtain a bend-less transcription from an up & down bend might be detrimental to the algorithm performance. Indeed, our procedure has an impact on *duration* and *pitch jump*$^{(\mathrm{n}\pm1)}$ which are among the most useful features to our algorithm. We observe however that our algorithm predicted correctly six bend labels in the selected examples with a limited amount of false positives. These encouraging results suggest that our method could be used as a suggestion tool for the idiomatic use of bends.

7. CONCLUSION

In this paper, we proposed a model of guitar bends and discussed how these expressive playing techniques relate to both tablature and score content. Introducing a set of high-level features, we showed that a decision tree can successfully predict bend occurrences with satisfactory precision, in spite of the difficulty of the task due to the low proportion of bent notes in guitar music. In particular, the low performance on predicting → labels suggests that our modeling choices could be improved and that held bends might not be considered as an expressiveness technique but rather another way of playing regular notes. An advantage of our approach is the use of a lightweight and explainable algorithm, facilitating its use in an assisted-composition context. In future work, this approach could be extended to other guitar playing techniques, and might benefit from adding more context information like the chord being played over a bar, using rhythm guitar parts aligned with lead guitar. Because bends are arguably more easily performed with the ring finger and little finger than other fingers of the fretting-hand, combining our work with finger prediction technique [16] might also improve prediction performance. Finally, our modeling strategy could also be used to study the playing style of specific guitarists, and evaluate the potential of bends for automatic guitarist identification. Indeed, our approach supposes that bends can be explained by general musical features regardless of the artist. This is debatable since famous players can be identified by their solos (without considering audio nor any playing technique) [26]. It would be interesting to see if bends are artist-dependent and, if so, to develop a model that predicts bends in the style of a specific guitarist.

Acknowledgements. This work is made with the support of the French National Research Agency, in the framework of the project TABASCO (ANR-22-CE38-0001). The authors would like to thank Arobas Music for providing the dataset and colleagues from the Algomus team for their thorough proofreading and insightful comments.

8. REFERENCES

[1] D. R. Grimes, "String Theory - The Physics of String-Bending and Other Electric Guitar Techniques," *Public Library of Science One*, vol. 9, no. 7, Jul. 2014.

[2] P. J. Gomez, "Modern Guitar Techniques; a view of History, Convergence of Musical Traditions and Contemporary Works (A guide for composers and guitarists)," Ph.D. dissertation, UC San Diego, 2016.

[3] Q. Xi, R. M. Bittner, J. Pauwels, X. Ye, and J. P. Bello, "Guitarset: A Dataset for Guitar Transcription," in *Proc. of the 19th International Society for Music Information Retrieval Conference*, Paris, France, 2018.

[4] A. Wiggins and Y. Kim, "Guitar Tablature Estimation with a Convolutional Neural Network," in *Proc. of the 20th International Society for Music Information Retrieval Conference*, Delft, The Netherlands, 2019.

[5] L. Reboursière, S. Dupont, O. Lähdeoja, C. Picard-Limpens, T. Drugman, and N. Riche, "Left and right-hand guitar playing techniques detection," *NIME*, 2012.

[6] S.-H. Chen, Y.-S. Lee, M.-C. Hsieh, and J.-C. Wang, "Playing Technique Classification Based on Deep Collaborative Learning of Variational Auto-Encoder and Gaussian Process," in *2018 IEEE International Conference on Multimedia and Expo (ICME)*, Jul. 2018, pp. 1–6.

[7] Yu-Fen Huang, Jeng-I Liang, I-Chieh Wei, and Li Su, "Joint analysis of mode and playing technique in Guqin performance with machine learning," in *Proc. of the 21st International Society for Music Information Retrieval Conference*, Montréal, Canada, 2020.

[8] P. Sarmento, A. Kumar, C. J. 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. Society for Music Information Retrieval Conf.*, Online, 2021.

[9] P. Sarmento, 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 *Artificial Intelligence in Music, Sound, Art and Design*, ser. Lecture Notes in Computer Science, C. Johnson, N. Rodríguez-Fernández, and S. M. Rebelo, Eds. Cham: Springer Nature Switzerland, 2023, pp. 260–275.

[10] Y.-H. Chen, Y.-H. Huang, W.-Y. Hsiao, and Y.-H. Yang, "Automatic Composition of Guitar Tabs by Transformers and Groove Modeling," in *Proc. of the 21st International Society for Music Information Retrieval Conference*, Montréal, Canada, 2020.

[11] M. McVicar, S. Fukayama, and M. Goto, "AutoLeadGuitar: Automatic generation of guitar solo phrases in the tablature space," in *2014 12th International Conference on Signal Processing (ICSP)*. Hangzhou, Zhejiang, China: IEEE, Oct. 2014, pp. 599–604.

[12] J. Ching and Y.-H. Yang, "Learning To Generate Piano Music With Sustain Pedals," in *Extended Abstracts for the Late-Breaking Demo Session of the 22nd Int. Society for Music Information Retrieval Conf.*, 2021.

[13] S. I. Sayegh, "Fingering for String Instruments with the Optimum Path Paradigm," *Computer Music Journal*, vol. 13, no. 3, pp. 76–84, 1989.

[14] G. Hori and S. Sagayama, "Minimax Viterbi algorithm for HMM-based Guitar fingering decision," in *Proc. of the 17th International Society for Music Information Retrieval Conference*, 2016.

[15] V. K. M. Cheung, H.-K. Kao, and L. Su, "Semi-supervised violin fingering generation using variational autoencoders," in *Proc. of the 22nd International Society for Music Information Retrieval Conference*, Online, 2021.

[16] G. Hori, "Three-Level Model for Fingering Decision of String Instruments," in *Proc. of the 15th International Symposium on CMMR*, Online, 2021.

[17] Y. Xie and R. Li, "Symbolic Music Playing Techniques Generation as a Tagging Problem," Oct. 2020, preprint.

[18] F. Zappa and S. Vai, *The Frank Zappa Guitar Book*. Hal Leonard Corporation, 2017.

[19] M. S. Cuthbert and C. Ariza, "music21: A Toolkit for Computer-Aided Musicology and Symbolic Music Data," in *Proc. of the 11th International Society for Music Information Retrieval Conference*, 2010.

[20] D. Temperley, *The Musical Language of Rock*. Oxford University Press, Jan. 2018.

[21] D. Régnier, N. Martin, and L. Bigo, "Identification of rhythm guitar sections in symbolic tablatures," in *Proc. of the 22nd International Society for Music Information Retrieval Conference*, Online, 2021.

[22] 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 Duchesnay, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, no. 85, pp. 2825–2830, 2011.

[23] L. Breiman, *Classification and regression trees*. Routledge, 2017.

[24] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Oversampling Technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, Jun. 2002.

[25] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001.

[26] O. Das, B. Kaneshiro, and T. Collins, "Analyzing and classifying guitarists from rock guitar solo tablature," in *Proceedings of the Sound and Music Computing Conference*, Limassol, Chypre, 2018.

SELF-SIMILARITY-BASED AND NOVELTY-BASED LOSS FOR MUSIC STRUCTURE ANALYSIS

Geoffroy Peeters
LTCI, Télécom-Paris, Institut Polytechnique de Paris, France

ABSTRACT

Music Structure Analysis (MSA) is the task aiming at identifying musical segments that compose a music track and possibly label them based on their similarity. In this paper we propose a supervised approach for the task of music boundary detection. In our approach we simultaneously learn features and convolution kernels. For this we jointly optimize - a loss based on the Self-Similarity-Matrix (SSM) obtained with the learned features, denoted by SSM-loss, and - a loss based on the novelty score obtained applying the learned kernels to the estimated SSM, denoted by novelty-loss. We also demonstrate that relative feature learning, through self-attention, is beneficial for the task of MSA. Finally, we compare the performances of our approach to previously proposed approaches on the standard RWC-Pop, and various subsets of SALAMI.

1 Introduction

Music Structure Analysis (MSA) is the task aiming at identifying musical segments that compose a music track (a.k.a. segment boundary estimation) and possibly label them based on their similarity (a.k.a. segment labeling). We deal here with MSA from audio. MSA is one of the oldest task in Music Information Retrieval [1] but still one of the most challenging. This is due to the difficulty to exactly define what music structure is and hence be able to create annotated datasets to measure progress or train systems. People agree that the structure can be considered from multiple viewpoints [2] [2] [3], is hierarchical [4] and is partly subjective [5]. Probably because of this complexity, the number of contributions in MSA has remained low despite its large number of applications: audio summarization [6], interactive browsing [7–9], musical analysis [10], tools for researcher (to help chord recognition [11], source separation [12] or downbeat estimation [13]).

To solve the two MSA tasks (boundary detection and segment labeling), three assumptions [14] are commonly used: (1) *novelty* (we assume that segments are defined by large —novel— changes of the musical content over time), (2) *homogeneity* (the musical content is homogeneous within a given segment) and (3) *repetition* (the musical content —homogeneous or not— can be repeated over time). This has been extended by [15] to a fourth *regularity* assumption (the segment's durations are regular over time). Combining those allows to construct MSA systems.

1.1 Related works

Over time, a large palette of approaches has been proposed for MSA. We only review the ones related to our work and refer the reader to Nieto et al. [16] for a good overview. We consider three periods according to the nature of the audio features –hand-crafted (HC) or learned by deep learning (DL) –, and the nature of the detection system which uses the audio features – HC or trained by DL –.

First period: HC detection system applied to HC audio features. In these systems HC audio features (such as MFCC or Chroma) were given as input to HC detection system (such as the checkerboard kernel, novelty-score [17]), unsupervised training (such as HMM [6], NMF [18]), supervised (such as OLDA [19]) or pattern matching algorithms (such as DTW [20] or variants [21]) .

Second period: DL detection system applied to HC audio features. Over time, more and larger annotated datasets for MSA have been developed; which concomitantly with the development of DL has allowed to reformulate the MSA task in terms of supervised learning. The detection system developed here mainly target the task of boundary detection. For example, [22] [23] [24] propose to train in a supervised way a Convolutional Networks (ConvNet) $\hat{y} = f^{\theta}(\mathbf{X})$ to estimate if the center of a patch of HC audio features $\mathbf{X}$ is a boundary (y=1). Various HC audio features (or combinations of) are used here: Log-Mel-Spectrogram, Pich-Class-Profile through SSM expressed in (time,time) or (time,lag).

Third period: HC detection system applied to DL audio features. To deal with the endless debate about the choice of HC audio features, McCallum et al. [25] propose to learn them. For this, they train an encoder f^{θ} by minimizing a Triplet Loss (TL) [26] between patches of beat-synchronous Constant-Q-Transform (CQT). For the TL, they propose a Self-Supervised-Learning (SSL) paradigm [3] to define the anchor A patch, positive P patch and negative N patch. Using the homogeneity assumption, neighboring times are supposed to be more similar to each

[1] Foote's paper [1] on SSM was published in 1999.
[2] musical role, acoustic similarity, instrument role, perceptual tests
[3] which does not require any annotated segments and labels

© G. Peeters. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** G. Peeters, "Self-Similarity-Based and Novelty-based loss for music structure analysis", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

other (therefore used to define A and P) than to distant ones (used to define N). For training they use a very large unlabeled dataset of 28345 songs. This method however does not consider the repetition assumption [4].

Wang et al. [27] revised McCallum approach in a supervised setting. In this, the patches P (resp. N) are now explicitly chosen so as to have the same (resp. different) annotated segment label than the patches A. This supervised method now consider both the homogeneity and repetition assumption. In another work [28], they propose a spectral-temporal Transformer-based model (SpecTNT) trained with a connectionist temporal localization (CTL) loss to jointly estimate music segments ad their labels.

McCallum approach has also been extended by Buisson et al. [29] to take benefit from the hierarchy of structure in music. They show that the obtained deep embeddings can improve segmentation at various levels of granularity.

Rather than learning features for MSA, Salamon et al. [30] proposed to re-use pretrained ones. Those are obtained using encoders previously trained on different tasks (Few-Shot Learning sound event and music auto-tagging). Those are then used as input to a Laplacian Structural Decompositon algorithm for MSA.

1.2 Proposal and paper organization

Following the previous taxonomy, our proposal would belong to the category "DL detection system applied to DL audio features" . Unlike previous feature learning approaches (that rely on a Triplet Loss paradigm), we utilize a more straightforward paradigm (illustrated in Figure 1) which is a succession of two steps, each with its own objective. The two objectives are jointly optimized.

In the **first step**, we learn the parameters θ of an encoder f^θ such that when applied to the sequence of inputs $\{\mathbf{X}_i\}_{i\in\{1...T\}}$ that represent a given track (where T is the length of temporal sequence), the encoded features allows the estimation of a SSM, $\hat{\mathbf{S}}^\theta_{ij}$, which attempts to reproduce a ground-truth SSM, $\mathbf{S}_{ij}$. For training f^θ we use an approach similar to the SSM-Net approach proposed in [31], i.e. defining a loss which directly compare the obtained SSM $\hat{\mathbf{S}}^\theta_{ij}$ to a ground-truth SSM $\mathbf{S}_{ij}$.

In the **second step**, we learn a set of kernels $\mathbf{K}^\theta$ such that when convolved over the main diagonal of the estimated SSM $\hat{\mathbf{S}}^\theta_{ij}$ it allows the estimation of a novelty score $\hat{\mathbf{n}}^\theta_i$, which attempts to reproduce a ground-truth novelty score, $\mathbf{n}_i$. This novelty score is usually obtained using a fixed checkerboard kernel [32]. The resulting function is named novelty score since high values in it indicate times where the content change (it is homogeneous before and after). It has been shown that better kernels can be used (for example using multi-scale kernels [33]) and that it is possible to train such kernels $\mathbf{K}^\theta$ considered as the kernels of a ConvNet (for example [22] and [23] in the case of a (time,lag) SSM or [24] in the case of a (time,time) SSM, which is our case). This is the approach we follow here.

[4] N could potentially be in a segment which is a repetition of the segment containing A

Another proposal we make in this paper, is to consider the learning of relative features, i.e. features which are relative to the given track.

Paper organization. We provide an overview of our system in part 2, describe the inputs to our system (part 2.1), detail the two losses (parts 2.2 and 2.3), motivate relative feature learning (part 2.4), detail the architecture of our encoder f^θ (part 2.5) and the training process (part 2.6). In part 3, we provide a large-scale evaluation of our proposal. It should be noted that although we only evaluate our method for the task of segment boundary detection, it can also be used for segment labeling given the clearness of the obtained SSM.

2 Proposal

2.1 Input data $\{\mathbf{X}_i\}$

The inputs $\{\mathbf{X}_i\}$ to our system are simple patches [5] of Log-Mel-Spectrogram. We didn't consider beat-synchronous features as in [25] given the non-reliability of beat estimation outside popular music. Using `librosa` [34], we first computed Mel-spectrogram with 80 mel-bands, using a 92ms window length and 23ms hop size. Those are then converted to log-amplitude using $\log(1+100\cdot mel)$. We then aggregate them (mean operator) over time to lead to a 0.1s hop size. The final $\{\mathbf{X}_i\}$ are then patches of 40 successive frames (corresponding to 4s.) with a hop size of 5 frames (corresponding to 0.5s.).

2.2 SSM-loss

Given a sequence of inputs $\{\mathbf{X}_i\}_{i\in\{1...T\}}$, we apply the same encoder f^θ individually to each $\mathbf{X}_i$ to obtain the corresponding sequence of embeddings $\{\mathbf{e}^\theta_i\}_{i\in\{1...T\}}$. Those are then L2-normalized. We can then easily construct an estimated SSM, $\hat{\mathbf{S}}^\theta_{ij}$, using a distance/similarity/divergence g between all pairs of projections:

$$\hat{\mathbf{S}}^\theta_{ij} = g(\mathbf{e}^\theta_i = f^\theta(\mathbf{X}_i), \mathbf{e}^\theta_j = f^\theta(\mathbf{X}_j)), \quad \forall i,j \tag{1}$$

We use here a "scaled" cosine-similarity for g which, because the embeddings are L2-normalized, reduces to

$$\hat{\mathbf{S}}^\theta_{ij} = 1 - \frac{1}{4}\|\mathbf{e}^\theta_i - \mathbf{e}^\theta_j\|^2_2 \quad \in [0,1] \tag{2}$$

It is then possible to compare $\hat{\mathbf{S}}^\theta_{ij}$ to a ground-truth binary SSM, $\mathbf{S}_{ij}$, derived from annotations. We consider this as a multi-class (a set of T^2 binary classifications) problem and hence minimize the sum of Binary-Cross-Entropy (BCE) losses. However, given the unbalancing between the two classes in $\mathbf{S}_{ij}$ (which contains much more 0 than 1), we used a weighting factor λ computed as the percentage of positive values in $\mathbf{S}_{ij}$. The lower λ is, the more we put emphasis on positive ($\mathbf{S}_{ij}$=1) examples:

$$\mathcal{L}^\theta_{SSM} = -\frac{1}{T^2}\sum_{i,j=1}^{T}(1-\lambda)\left[\mathbf{S}_{ij}\log(\hat{\mathbf{S}}^\theta_{ij})\right] + \lambda\left[(1-\mathbf{S}_{ij})\log(1-\hat{\mathbf{S}}^\theta_{ij})\right] \tag{3}$$

Since the computation of the SSM $\hat{\mathbf{S}}^\theta_{ij}$ is differentiable w.r.t. to the embeddings $\{\mathbf{e}^\theta_i\}$, we can compute $\frac{\partial \mathcal{L}^\theta_{SSM}}{\partial \theta}$:

[5] We utilized patches as input (rather than frames) because we believe that homogeneity exists at the pattern level rather than the frame level.

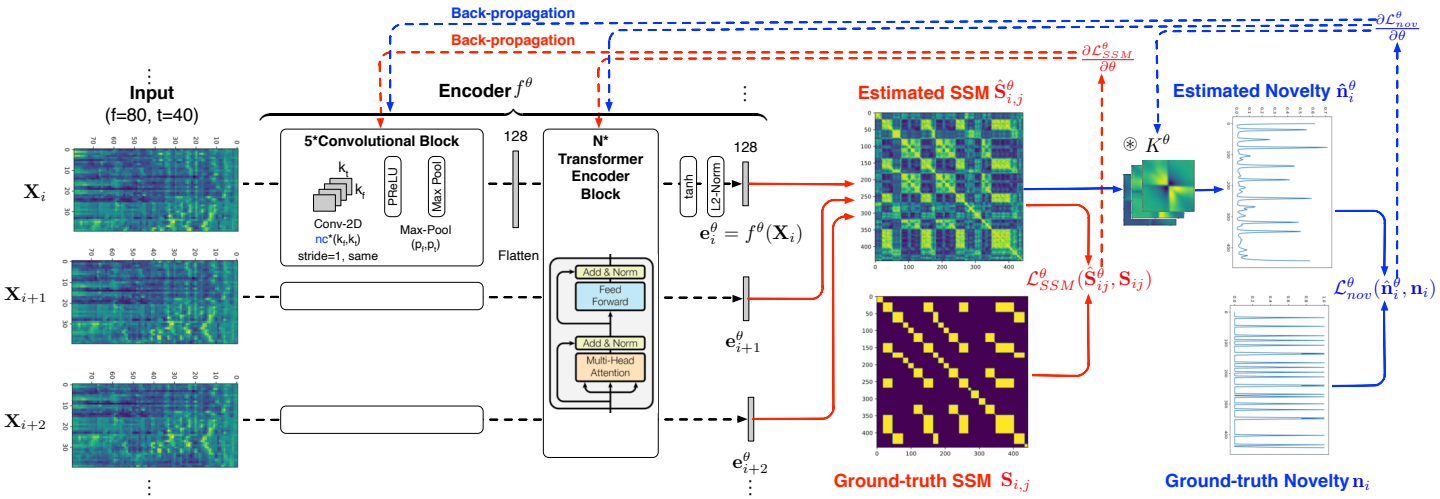


Figure 1. Proposed architecture and training paradigm minimizing a SSM loss $\mathcal{L}_{SSM}^\theta$ and a novety loss $\mathcal{L}_{nov}^\theta$.

$$\frac{\mathcal{L}_{SSM}^\theta}{\theta} = \sum_{i\ j=1}^{T} \frac{\mathcal{L}_{SSM}^\theta}{\mathbf{S}_{ij}^\theta} \left(\frac{\mathbf{S}_{ij}^\theta}{\mathbf{e}_i^\theta} \frac{\mathbf{e}_i^\theta}{\theta} + \frac{\mathbf{S}_{ij}^\theta}{\mathbf{e}_j^\theta} \frac{\mathbf{e}_j^{\theta\prime}}{\theta} \right. \tag{4}$$

We can then use standard gradient-descent algorithms to optimize θ which will jointly optimize f^θ for all the $\mathbf{X}_i$.

Optimizing directly $\mathbf{S}_{ij}^\theta$ has relationship with Metric Learning / Contrastive Learning in which the A,P,N are chosen based on their similarity (such as in Wang et al. [27]). In comparison, we consider here simultaneously all possible pairs of time as A,P,N. This is actually in line with the fact that we aim at learning features relative to a track (see part 2.4) and we therefore need to consider simultaneously the interaction between all projections $\mathbf{e}_i^\theta$.

2.3 Novelty-loss

We propose to learn the kernels $\mathbf{K}^\theta$ such that when convolved with the estimated SSM $\mathbf{S}_{ij}^\theta$ (see eq.(2)) along its main diagonal the resulting estimated novelty score $\mathbf{n}_i^\theta$ approximate a ground-truth novelty score $\mathbf{n}_i$. This kernel convolution can be simply implemented as an extra convolution layer (without bias) on top of the estimated SSM $\mathbf{S}_{ij}^\theta$ with a sigmoid output activation. We then define the novelty-loss as

$$\mathcal{L}_{nov}^\theta = \frac{1}{T} \sum_{i=1}^{T} BCE(\mathbf{n}_i^\theta\ \ \mathbf{n}_i) \tag{5}$$

2.4 Relative feature learning

In previous works dealing with feature learning for MSA it is assumed that, once trained, the network f^θ always projects a given segment $\mathbf{X}_i$ in the same way whatever its surrounding context.

We advocate here that for the task of MSA the projection of $\mathbf{X}_i$ should depend on its context. The motivation for doing so is that the features that highlight the temporal structure of a music track usually depend on the track itself. For example, if the instrumentation or the timbre remains constant over the track, the structure may arise from variation of the harmonic content; in other cases, it will be the opposite. Therefore, feature learning for MSA should be made relative-to-a-track.

To let each feature $\mathbf{X}_i$ "know" about surrounding times features $\mathbf{X}_1$ $\mathbf{X}_{i-1}$ $\mathbf{X}_{i+1}$ $\mathbf{X}_T$ we introduce layers of Self-Attention (SA) [35] in our encoder [6] .

2.5 Network architecture f^θ

The architecture of the encoder f^θ is given in Figure 1. It is made of a succession of 5 consecutive convolution blocks followed by N blocks of Transformer-Encoder.

Each convolution block is made of a 2D convolution followed by a PReLU [36] activation and a 2D max-pooling. The kernel size $(k_f\ \ k_t)$, the number of channels n_c and pooling size $(p_f\ \ p_t)$) of each layer are the following: layer-1: $(k_f\ \ k_t)$=(5,5) n_c=32 $(p_f\ \ p_t)$=(2,2), layer-2: (5,5) 32 (2,2), layer-3: (5,5) 64 (2,2), layer-4: (5,5) 64 (2,2), layer-5: (5,2) 128 (5,2). The output of the last convolutional blocks has dimension (1,1) with n_c=128 channels and is flattened to a 128-dim vector.

Each input $\mathbf{X}_i$ is independently projected using the convolutional blocks. These outputs are then considered as a temporal sequence which is fed to N blocks of Transformer Encoder (each made up of a SA layer with 8 heads, skip-connection, a normalization layer and two fully-connected layers with an internal dimension of 128). The outputs are then passed to a tanh and L2-normalized. They form a sequence of embeddings $\mathbf{e}_i^\theta$ i 1 T with $\mathbf{e}_i^\theta$ R^{128} which are used to compute $\mathbf{S}_{ij}^\theta$.

The size of the kernels $\mathbf{K}^\theta$ is fixed to (41,41) which roughly corresponds to 20s. The kernels $\mathbf{K}^\theta$ are either initialized randomly or initialized with checkerboard kernels similar to the ones of [32]. In this case, checkerboard kernels have the same size (41,41) but are damped with Gaussian function with different σ (randomly chosen in the range $[3s\ \ 5s]$). We used 3 different kernels $\mathbf{K}^\theta$ which are

[6] Note that the use of the SSM-loss alone does not allows f^θ to encode relative features; this is the task of the SA.

then combined using (1x1) convolution. The diagonal of the resulting feature-map then goes to a sigmoid activation and is considered as the estimated novelty $\mathbf{n}_i^\theta$.

Our architecture remains lightweight with a number of parameters ranging from 268K to 567K depending on the number of Transformer Encoder blocks (from N=0 to 3).

2.6 Training.

We train our network by minimizing jointly the two losses defined by eq. (3) and eq. (5):

$$\mathcal{L}^\theta = \alpha \mathcal{L}^\theta_{SSM} + (1-\alpha)\mathcal{L}^\theta_{nov} \quad (6)$$

We used the ADAM optimizer with a learning rate of 0.001, used early-stopping monitoring $\mathcal{L}^\theta$ on the validation data with a patience of 50 and a maximum of 500 epochs.

Considering that we need the whole sequence of embeddings $\mathbf{e}_i^\theta$ of a track to compute $\mathbf{S}_{ij}^\theta$ and get the gradients $\frac{\mathcal{L}^\theta}{\theta}$, the mini-batch-size m is here defined as the number of tracks. We used a value of m=10 tracks.

2.6.1 *Generating ground-truth for training*

Ground-truth SSM $\mathbf{S}_{ij}$. The ground-truth SSM, $\mathbf{S}_{ij}$, is constructed using annotated segments (start and end time) and their associated labels. We rely on the homogeneity assumption, i.e. we suppose that all times t_i that fall within a segment are identical since they share the same label. If we denote by seg(t_i) the segment t_i belongs to and by label(seg(t_i)) its label, we assign the value $\mathbf{S}_{ij} = 1$ if label(seg(t_i)) = label(seg(t_j)) and 0 otherwise. Note that we could relax this identity constraint to allow representing similarity between labels (for example using an edit distance between labels). This is for example important for RWC-POP dataset, where labels denotes some proximities (`verse A` and `verse B`) but are here considered as different. Also, it could be important to consider the case of non-homogeneity of the repetitions and create a ground-truth $\mathbf{S}_{ij}$ made of "sub-diagonals" rather than "blocks".

Ground-truth novelty score $\mathbf{n}_i$. The ground-truth novelty score, $\mathbf{n}_i$, is also constructed using the annotated segments (start and end time). We set $\mathbf{n}_i$ to 1 when segment changes at time i, 0 otherwise. As proposed by [37] we smooth over time the boundary annotations by applying a low-pass filter with a triangular-shape $0\ 25\ 0\ 5\ 1\ 0\ 5\ 0\ 25$.

3 Evaluation

We assess here the performance of our proposal using various test sets, compare it to previously published results, conduct an ablation study, and illustrate its results.

3.1 Datasets

For training we used a subset of 693 tracks from the **Harmonix** dataset [38] [7] and the 298 tracks of the **Isophonics** dataset [39]. For testing we used

[7] Given the non-accessibility of Harmonix audio, those have been downloaded from `YouTube` and re-annotation has been necessary because of non-synchronicity of the original annotations.

Datasets	T	S	L	S	L
Harmonix	693	13	17.15		
Isophonics	298	11	15.98		
RWC-Pop-AIST	100	17	14.31		
		Upper		Lower	
SA-Pop (An1)	276	12	15.49	30	5.73
SA-Pop (An2)	175	12	14.64	31	5.67
SA-IA (An1)	444	14	18.32	50	4.43
SA-IA (An2)	244	12.5	18.67	37	7.00
SA-Two (An1)	882	11	18.25	30	6.89
SA-Two (An2)	882	11	17.76	31	6.39

Table 1. Description of the datasets used in our evaluation: number of tracks T, median value of the number of segments per track S , median value of segment duration L in seconds (note that [29] indicate L in number of beats).

- **RWC-Pop-AIST** the 100 tracks of the RWC-Pop [40] with AIST annotations [41] and the following three subsets of the SALAMI [3] dataset:
- **SA-Pop** is the subset of SALAMI tracks with CLASS equal to Popular,
- **SA-IA** is the subset of SALAMI tracks with SOURCE equal to IA (Internet Archive),
- **SA-Two** is the subset of SALAMI tracks with at least two annotations.

For each SALAMI subset we considered the two annotations (An1, An2) and the two levels of flat annotations (Upper, Lower); those correspond to the files `textfile{1,2}_{upper,lowercase}.txt`.

In Table 1 we describe these datasets. According to the values of L our training-sets better match the Upper annotations than the Lower ones of SALAMI. Also, the size of our kernels $\mathbf{K}^\theta$ (roughly 20s., see part 2.5) assumes homogeneous segments of roughly 10s. and are therefore closest to the L of Upper annotations.

3.2 Segment detection from novelty score

To get the estimated segment boundaries from the estimated novelty score $\mathbf{n}_i^\theta$ we used a simple peak-to-mean ratio algorithm similar to [25]. Using the same notations as [25] eq. (5), we compute the mean with a window of duration $2T$=20s, and then detect local peaks with a threshold τ=1.35 and a minimum inter-distance of 7s.

3.3 Performance metrics

We evaluate the performance of segment boundary detection using the common Hit-Rate metrics using precision-windows of 3s and 0.5s. We only display here the Hit-Rate F-measures denoted by `HR3F` and `HR0.5F`. We used `mir_eval` [43] with `mir_eval.segment.detection` ignoring track start and end annotations (`Trim=True`). We point out that without "trimming" (the start and end time) we would gain +3% on average (from .713 to .743 for RWC-Pop).

	RWC-Pop-AIST		SA-Pop		SA-IA		SA-Two		
	HR.5F	HR3F	HR.5F	HR3F	HR.5F	HR3F	HR.5F	HR3F	Annotation
Grill [23, 42] GS1	.506	**.715**	-	-	-	-	.541	.623	Up./An-*
McCallum [25] Unsynch.	-	-	-	-	-	.497	-	-	
Beat-synch.	-	-	-	-	-	**.535**	-	-	
Salamon [30] $\text{DEF}_{0.5,0.5}/*_{\mu^H\ \gamma^H}$	-	-	-	-	-	-	.337	.563	Up./An-*
Wang [27] scluster/D/eu/mul	.438	.653	.447	.623	-	-	.356	.553	Up./An-*
Buisson [29] HE_0/HE_1	-	.681	-	-	-	-	-	**.597 / .595**	Up./An-1/2
			-	-	-	-	-	**.611 / .600**	Low./An-1/2
Ours (best conf.)	.399	**.713**	.298 / .295	**.631/ .624**	.250 / .261	**.520 / .511**	.231 / .237	.521 / .530	Up./An-1/2
			.296 / .318	.570 / .610	.302 / .336	.547 / .612	.287 / .287	.589 / .589	Low./An-1/2
Ablation study N									
N=3/α=0.5/K:train-Init:chck		**.713**		.532		.472		.448	Up./An-1
N=2/α=0.5/K:train-Init:chck		.701		.535		.474		.449	Up./An-1
N=1/α=0.5/K:train-Init:chck		.677		**.631**		**.520**		**.521**	Up./An1
N=0/α=0.5/K:train-Init:chck		.696		.535		.459		.443	Up./An-1
Ablation study α									
N=3/α=1/K:train-Init:chck		.154		.121		.102		.111	Up./An-1
N=3/α=0/K:train-Init:chck		.007		.120		.026		.095	Up./An-1
Ablation study $\mathbf{K}^\theta$									
N=3/α=0.5/K:train-Init:randn		**.713**		.543		.470		.457	Up./An-1
N=1/α=0.5/K:train-Init:randn		.709		.547		.470		.457	Up./An-1
N=3/α=0.5/K:fix-Init:chck		.330		.250		.199		.196	Up./An-1

Table 2. Results of segment boundary detection using various test-sets and configurations

3.4 Comparison with previous works

In the following we will compare our results with the ones previously published by Grill and Schlüter in [23, 42], McCallum et al. in [25], Salamon et al. in [30], Wang et al. in [27] and Buisson et al. in [29]. We first check if their test-sets match ours.

For SA-Pop, Wang [27] used *"a subset with 445 annotated songs (from 274 unique songs) in the "popular" genre"*. This roughly matches our SA-Pop (An1)+(An2) which provides 276+175=451 annotations. They used the Upper-case annotations (personal communication).

For SA-IA, McCallum [25] used *"the internet archive portion of the SALAMI dataset (SALAMI-IA) consisting of 375 hand annotated recordings"*. This is much lower than our SA-IA (An1)+(An2) which provides 444+244=688 annotations. Moreover, it is not clear whether they used the Upper, Lower or Functional annotations.

Finally, for SA-Two, Salamon [30] Table 3 used the Upper-case annotations of tracks with at least 2 annotations (884 tracks); Wang et al. [27] *"we treat each annotation of a song separately, yielding 2243 annotated songs in total"* and Buisson et al. [29] used the Upper and Lower-case annotations of tracks with at least 2 annotations (884 tracks). This roughly corresponds to our SA-Pop (An1)+(An2) which has 882 tracks.

3.5 Results and discussions

Results are given in Table 2. The upper part shows previously published results, although not all systems were evaluated on all test sets. The middle part shows the results achieved with the best configuration of our system.

For **RWC-Pop-AIST**, we obtained a HR3F= .713 [8] which is comparable to those of Grill and Schlüter (.715). However, for HR.5F our results are below (.399 < .506). This can be explained by the fact that the hop-size of our features $\mathbf{X}_i$ was chosen large (0.5s) and does not allow to have a precise boundary estimation. We have chosen a large hop size to reduce the size of $\mathbf{S}^\theta_{ij}$ (hence the computation time and memory footprints); it also allows to keep the size of the $\mathbf{K}^\theta$ manageable. Because of this, all our results with HR.5F are actually low. Therefore, we only discuss the results for HR3F in the following.

For **SA-Pop**, we obtained a HR3F of .631/.624 [9] for the two Upper annotations (Up./An-1/2) which is slightly above those of Wang et al. (.623). For the two lower annotations (Low./An-1/2) we get a HR3F of .570/.610 [10]. Wang et al. does not provide these results.

For **SA-IA**, we obtained a HR3F of .520/.511 [11] for the two Upper annotations and .547/.612 [12] for the two Lower annotations. This has to be compared to the .497 (unsynchronized) and .535 (beat-synchronized) obtained by McCallum et al., but as explained, it is not clear whether they used Upper, Lower or Functional annotations.

For **SA-Two**, we obtained a HR3F of .521/.530 [13] for the two Upper annotations. This is slightly lower than the results of Wang et al. (.553), Salamon et al. (.563), Buisson et al. (.597) and largely below the ones of Grill and Schlüter (.623). For the Lower annotations, we obtained a HR3F of .589/.589 [14] which is slightly below the ones of Buisson et al. (.611). It should be noted however that in our work we didn't used any data from SALAMI, neither for training or validation (such as early stopping).

For SA-IA and SA-two, our results are higher for the Lower annotations than the Upper ones. This is surprising since according to Table 1 the characteristics (L value) of our training sets are closer to the Upper case. Also (see footnotes 8–14), our algorithm tends to over-segment when

[8] The Precision and Recall at 3seconds are P3F=.735, R3F=0.715

[9] P3F=.581, R3F=0.760/ P3F=.566, R3F=0.771 → over-segmentation

[10] P3F=.860, R3F=0.468/ P3F=.877, R3F=0.497 → under-segment.

[11] P3F=.435, R3F=0.718/ P3F=.411, R3F=0.751 → over-segment.

[12] P3F=.811, R3F=0.451/ P3F=.756, R3F=0.546 → under-segment.

[13] P3F=.433, R3F=0.749/ P3F=.442, R3F=0.754→ over-segment.

[14] P3F=.768, R3F=0.523/ P3F=.768, R3F=0.523→ under-segment.

considering the Upper annotation and under-segment when considering the Lower ones. Our kernel size is actually between the L values of the Upper and Lower annotations.

3.6 Ablation study

In the lower part of Table 2 we perform an ablation study of our system. For the SA-{Pop,IA,Two} test-sets, we only perform the study using the Upper/An1 annotations

We first check the optimal number N 0 1 2 3 of layers of Transformer Encoder. We see that for all test-sets the use of Transformer Encoder ($N > 0$) is beneficial. For RWC-Pop-AIST, the optimal number is N=3 while for all three SA-{Pop,IA,Two} test-sets it is always N=1.

We then check whether jointly optimizing the two losses L^{θ}_{SSM} and $\mathcal{L}^{\theta}_{nov}$ of eq. (6) is necessary. We considered three cases: α=1 (only optimizing L^{θ}_{SSM}), α=0.5 (optimizing both), α=0 (only optimizing $\mathcal{L}^{\theta}_{nov}$). For all test-sets, we see that optimizing jointly the two losses is highly beneficial: for example, for RWC-Pop-AIST, HR3F=.713 with α=0.5, .154 with α=1 and .007 for α=0.

Finally, we check various configurations of the kernels $\mathbf{K}^{\theta}$. $\mathbf{K}^{\theta}$ is either [K:train-Init:chck]: trained starting from checkerboard kernels initialisation, [K:train-Init:randn]: trained starting from random initialisations, [K:fix-Init:chck]: fixed (not trained) to checkerboard kernels (we still train the 1x1 convolution). We see that for all test-sets it is beneficial to train $\mathbf{K}^{\theta}$ (the worst results are obtained with [K:fix-Init:chck]). For RWC-Pop-AIST, the results are the same whether kernels are initialized randomly or with checkerboard kernels. For SA-{Pop,IA,Two} the checkerboard kernels initialization is beneficial.

3.7 Examples

Figure 2 illustrates the three kernels $\mathbf{K}^{\theta}$ learned using the [N=3/α=0.5/K:train-Init:chck] configuration. As one can see, while the middle one looks close to the classical checkerboard kernel of Foote [32] (but with an emphasis on the diagonal), the first seems to highlight the transition from a non-homogeneous to an homogeneous part; while the third seems a re-scaled version of the second (homogeneity at a larger scale). Figure 3 illustrates the $\mathbf{S}^{\theta}_{ij}$ and $\mathbf{n}^{\theta}_{i}$ obtained by our system on track-01 from RWC-Pop-AIST (chosen as the first item of our test-set). We compare the results when trained in the [N=3 / α=0.5 / K:train-Init:chck] configuration and with the untrained system using [K:fix-Init:chck] for the kernels. For comparison we indicate the ground-truth $\mathbf{S}_{ij}$ and $\mathbf{n}_{i}$. In this figure, the benefits of training both L^{θ}_{SSM} and$\mathcal{L}^{\theta}_{nov}$ appears clearly.

Reproducibility. The code and the datasets used in this work are available at: https://github.com/geoffroypeeters/ssmnet_ISMIR2023

4 Conclusion

In this work, we proposed a simple approach for deep learning-based Music Structure Analysis algorithm: we learn an encoder f^{θ} such that the resulting learned features allow to best approximate a ground-truth SSM; we jointly learn segmentation kernels that when applied to the estimated SSM we best approximate a ground-truth novelty score. We also propose to learn relative features, i.e. features relative to a track, by introducing Self-Attention layers in our encoder. According to HR3F, our results are either better than previous state-of-the-art (SA-Pop, SA-IA unsynchronous), similar (RWC-Pop-AIST) or worst (SA-Two). Our approach has the advantage to be lightweight (around 500K parameters) and based on criteria which are semantically linked to the task of MSA. Future works will concentrate on making our approach applicable to finer temporal resolutions, therefore allowing improving our performances at HR.5F.

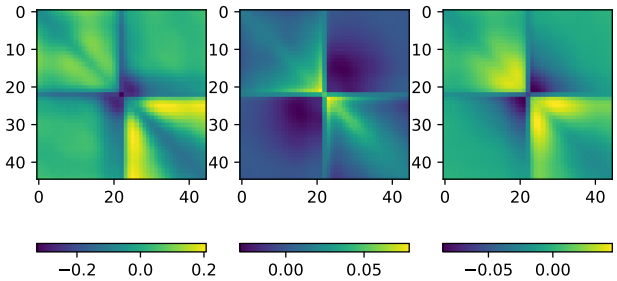


Figure 2. The three kernels $\mathbf{K}^{\theta}$ learned using the [N=3 / α=0.5 / K:train-Init:chck] configuration.

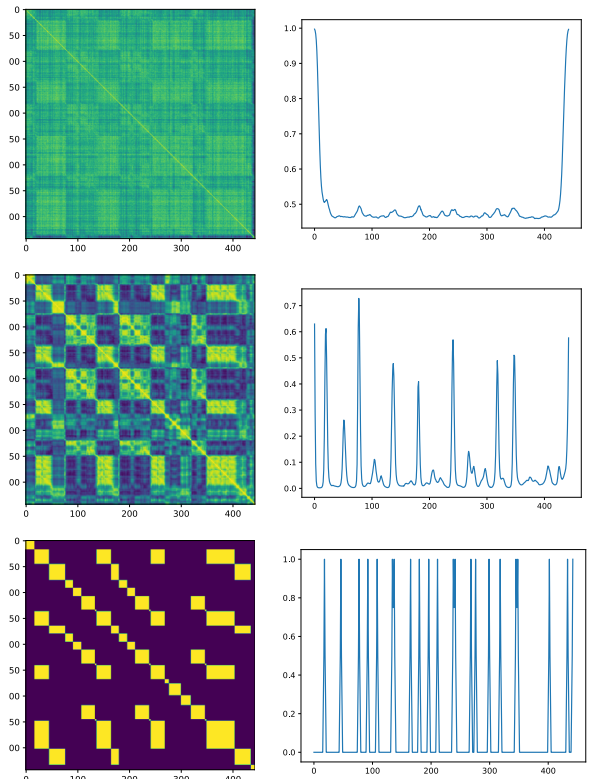


Figure 3. [Top] $\mathbf{S}^{\theta}_{ij}$ and $\mathbf{n}^{\theta}_{i}$ obtained with untrained system using [K:fix-Init:chck] for the kernels, [Middle] same with [N=3 / α=0.5 / K:train-Init:chck], [Bottom] ground-truth $\mathbf{S}_{ij}$ and $\mathbf{n}_{i}$.

5 References

[1] J. Foote, "Visualizing music and audio using self-similarity," in *Proc. of ACM Multimedia*, Orlando, Florida, USA, 1999, pp. 77–80.

[2] G. Peeters and E. Deruty, "Is music structure annotation multi-dimensional ? a proposal for robust local music annotation," in *Proc. of LSAS (International Workshop on Learning the Semantics of Audio Signals)*, Graz, Austria, 2009.

[3] J. B. L. Smith, J. Burgoyne, I. Fujinaga, D. Roure, and J. S. Downie, "Design and creation of a large-scale database of structural annotations," in *Proc. of ISMIR (International Society for Music Information Retrieval)*, Miami, Florida, USA, 2011.

[4] B. McFee, O. Nieto, M. M. Farbood, and J. P. Bello, "Evaluating hierarchical structure in music annotations," *Frontiers in Psychology*, vol. 8, p. 1337, 2017.

[5] M. Bruderer, M. McKinney, and A. Kohlrausch, "Structural boundary perception in popular music," in *Proc. of ISMIR (International Society for Music Information Retrieval)*, Victoria, Canada, 2006.

[6] G. Peeters, A. Laburthe, and X. Rodet, "Toward automatic music audio summary generation from signal analysis," in *Proc. of ISMIR (International Society for Music Information Retrieval)*, Paris, France, 2002, pp. 94–100.

[7] G. Peeters, D. Fenech, and X. Rodet, "MCIpa: A music content information player and annotator for discovering music," in *Proc. of ISMIR (International Society for Music Information Retrieval)*, Philadelphia, PA, USA, 2008.

[8] G. Peeters, F. Cornu, D. Tardieu, C. Charbuillet, J. J. Burred, M. Ramona, M. Vian, V. Botherel, J.-B. Rault, and J.-P. Cabanal, "A multimedia search and navigation prototype, including music and video-clips," in *Proc. of ISMIR (International Society for Music Information Retrieval)*, Porto, Portugal, October 2012.

[9] M. Goto, K. Yoshii, H. Fujihara, M. Mauch, and T. Nakano, "Songle: A web service for active music listening improved by user contributions." in *Proc. of ISMIR (International Society for Music Information Retrieval)*, Miami, Florida, USA, 2011.

[10] M. Mueller and N. Jiang, "A scape plot representation for visualizing repetitive structures of music recordings," in *Proc. of ISMIR (International Society for Music Information Retrieval)*, Porto, Portugal, 2012.

[11] M. Mauch, K. Noland, and S. Dixon, "Using musical structure to enhance automatic chord transcription," in *Proc. of ISMIR (International Society for Music Information Retrieval)*, Kobe, Japan, 2009.

[12] Z. Rafii and B. Pardo, "Repeating pattern extraction technique (repet): A simple method for music/voice separation," *Audio, Speech and Language Processing, IEEE Transactions on*, vol. 21, no. 1, pp. 73–84, January 2013.

[13] M. Fuentes, B. McFee, H. C. Crayencour, S. Essid, and J. P. Bello, "A music structure informed downbeat tracking system using skip-chain conditional random fields and deep learning," in *Proc. of IEEE ICASSP (International Conference on Acoustics, Speech, and Signal Processing)*, Brighton, UK, 2019, pp. 481–485.

[14] J. Paulus, M. Müller, and A. Klapuri, "Audio-based music structure analysis," in *Proc. of ISMIR (International Society for Music Information Retrieval)*, Utrecht, The Netherlands, 2010.

[15] G. Sargent, F. Bimbot, and E. Vincent, "A regularity-constrained viterbi algorithm and its application to the structural segmentation of songs," in *Proc. of ISMIR (International Society for Music Information Retrieval)*, Miami, Florida, USA, 2011.

[16] 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*, vol. 3, no. 1, 2020.

[17] J. Foote, "Automatic audio segmentation using a measure of audio novelty," in *Proc. of IEEE ICME (International Conference on Multimedia and Expo)*, New York City, NY, USA, 2000.

[18] F. Kaiser and T. Sikora, "Music structure discovery in popular music using non-negative matrix," in *Proc. of ISMIR (International Society for Music Information Retrieval)*, Utrecht, The Netherlands, 2010.

[19] B. McFee and D. P. W. Ellis, "Learning to segment songs with ordinal linear discriminant analysis," in *Proc. of IEEE ICASSP (International Conference on Acoustics, Speech, and Signal Processing)*, Florence, Italy, 2014.

[20] M. Müller, N. Jiang, and P. Grosche, "A robust fitness measure for capturing repetitions in music recordings with applications to audio thumbnailing," *Audio, Speech and Language Processing, IEEE Transactions on*, vol. 21, no. 3, pp. 531–543, 2013.

[21] 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.

[22] K. Ullrich, J. Schlüter, and T. Grill, "Boundary Detection in Music Structure Analysis using Convolutional

Neural Networks," in *Proc. of ISMIR (International Society for Music Information Retrieval)*, Taipei, Taiwan, 2014.

[23] T. Grill and J. Schlüter, "Music Boundary Detection Using Neural Networks on Combined Features and Two-Level Annotations," in *Proc. of ISMIR (International Society for Music Information Retrieval)*, Malaga, Spain, 2015.

[24] A. Cohen-Hadria and G. Peeters, "Music Structure Boundaries Estimation Using Multiple Self-Similarity Matrices as Input Depth of Convolutional Neural Networks," in *AES International Conference on Semantic Audio*, Erlangen, Germany, June, 22–24, 2017.

[25] M. C. McCallum, "Unsupervised Learning of Deep Features for Music Segmentation," in *Proc. of IEEE ICASSP (International Conference on Acoustics, Speech, and Signal Processing)*, Brighton, UK, May 2019.

[26] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2015, pp. 815–823, iSSN: 1063-6919.

[27] J.-C. Wang, J. B. L. Smith, W.-T. Lu, and X. Song, "Supervised metric learning for music structure features," in *Proc. of ISMIR (International Society for Music Information Retrieval)*, Online, November, 8–12 2021.

[28] 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 *Proc. of IEEE ICASSP (International Conference on Acoustics, Speech, and Signal Processing)*, Virtual and Singapore, May 2022.

[29] M. Buisson, B. McFee, S. Essid, and H.-C. Crayencour, "Learning multi-level representations for hierarchical music structure analysis," in *Proc. of ISMIR (International Society for Music Information Retrieval)*, 2022.

[30] J. Salamon, O. Nieto, and N. J. Bryan, "Deep embeddings and section fusion improve music segmentation," in *Proc. of ISMIR (International Society for Music Information Retrieval)*, Online, November, 8–12 2021.

[31] G. Peeters and F. Angulo, "Ssm-net: Feature learning for music structure analysis using a self-similarity-matrix based loss," in *Late-Breaking/Demo Session of ISMIR (International Society for Music Information Retrieval)*, 2022.

[32] J. Foote, "Automatic audio segmentation using a measure of audio novelty," in *Proc. of IEEE ICME (International Conference on Multimedia and Expo)*, New York City, NY, USA, 2000, pp. 452–455.

[33] F. Kaiser and G. Peeters, "Multiple hypotheses at multiple scales for audio novelty computation within music," in *Proc. of IEEE ICASSP (International Conference on Acoustics, Speech, and Signal Processing)*, Vancouver, British Columbia, Canada, May 2013.

[34] 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.

[35] 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*, vol. 30, 2017.

[36] G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter, "Self-normalizing neural networks," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, pp. 972–981.

[37] J. Schlüter and S. Böck, "Improved musical onset detection with Convolutional Neural Networks," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, pp. 6979–6983, iSSN: 2379-190X.

[38] O. Nieto, M. McCallum, M. E. P. Davies, A. Robertson, A. Stark, and E. Egozy, "The Harmonix Set: Beats, Downbeats, and Functional Segment Annotations of Western Popular Music," in *Proc. of ISMIR (International Society for Music Information Retrieval)*, Delft, The Netherlands, 2019.

[39] M. Mauch, C. Cannam, M. Davies, S. Dixon, C. Harte, S. Klozali, D. Tidhar, and M. Sandler, "Omras2 metadata project 2009," in *Late-Breaking/Demo Session of ISMIR (International Society for Music Information Retrieval)*, Kobe, Japan, 2009.

[40] M. Goto, "Development of the RWC Music Database," *Proc. of ICA (18th International Congress on Acoustics)*, 2004.

[41] ——, "Aist annotation for the rwc music database," in *Proc. of ISMIR (International Society for Music Information Retrieval)*, Victoria, BC, Canada, 2006, pp. pp.359–360.

[42] T. Grill and J. Schlüter, "Structural segmentation with convolutional neural networks MIREX submission," 2015.

[43] 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. of ISMIR (International Society for Music Information Retrieval)*, Taipei, Taiwan, 2014.

MODELING HARMONIC SIMILARITY FOR JAZZ USING CO-OCCURRENCE VECTORS AND THE MEMBRANE AREA

Carey Bunks[1,2] **Tillman Weyde**[2] **Simon Dixon**[1] **Bruno Di Giorgi**[3]
[1] Queen Mary University of London, UK
[2] City, University of London, UK
[3] Apple, UK

ABSTRACT

In jazz, measuring harmonic similarity is complicated by the common practice of reharmonization – the altering or substitution of chords without fundamentally changing the piece's harmonic identity. This is analogous to natural language processing tasks where synonymous terms can be used interchangeably without significantly modifying the meaning of a text. Our approach to modeling harmonic similarity borrows from NLP techniques, such as distributional semantics, by embedding chords into a vector space using a co-occurrence matrix. We show that the method can robustly detect harmonic similarity between songs, even when reharmonized. The co-occurrence matrix is computed from a corpus of symbolic jazz-chord progressions, and the result is a map from chords into vectors. A song's harmony can then be represented as a piecewise-linear path constructed from the cumulative sum of its chord vectors. For any two songs, their harmonic similarity can be measured as the minimal surface membrane area between their vector paths. Using a dataset of jazz contrafacts, we show that our approach reduces the median rank of matches from 318 to 18 compared to a baseline approach using pitch class vectors.

1. INTRODUCTION

Measuring similarity between songs is important for many music information retrieval tasks, for example, recommendation systems, copyright infringement detection, and genre classification systems. Many different types of features can be used to compare songs, but the specific focus of this paper is on jazz harmony as represented by the symbolic chord progressions found on leadsheets.

The analysis of harmonic similarity has been studied using N-grams [1], parse trees [2, 3], and NLP methods such as TF-IDF, Latent Semantic Analysis (LSA), and Doc2Vec [4]. The approach taken in this paper is based on embedding chord symbols into a vector space through the computation of a co-occurrence matrix [5]. As will be seen when we describe the data in Section 2, many chord symbols occur only rarely. To reduce computational problems due to sparsity, the dimensionality of chord space should be reduced [6]. A typical machine learning approach for this might use an algorithm such as truncated singular value decomposition after vectorization [7]. In this work, however, we use music theory to reduce the number of effective chord symbols prior to vectorization, which in turn reduces the chord space dimensionality. In the ensuing sections we describe the data, explain our approach to dimensionality reduction, and give computational details of how we compute the co-occurrence matrix. We then explain how the chord vectors generated from the co-occurrence matrix are used to represent chord progressions, and we present a novel harmonic-similarity metric, the *membrane area*.

The experimental part of our paper is based on analyzing contrafacts. In jazz, a contrafact is a song whose harmony is similar to that of another song, but which has a different melody [8]. The tune *I Got Rhythm*, by George Gershwin (1930), is a well-known source of many contrafacts, [1] and there are numerous other examples [9–11]. In addition to the difference in melody, contrafact chord progressions often feature reharmonization, a common practice in jazz that makes chord substitutions in a song while maintaining its harmonic identity [12]. Reharmonization is a core characteristic of jazz – so much so that there are typically reharmonizations from chorus to chorus even in a single performance of a jazz song.

2. THE DATA

The data used in this paper is a corpus of symbolic chord progressions similar to those found in jazz fake books, such as the Real Book [13]. The progressions are mainly from jazz standards, but also include some blues, jazz-blues, modal jazz, and jazz versions of pop tunes. The corpus is derived from a collection distributed with *Impro-Visor*, an open-source music notation program. [2] Our modifications remove control information used by the Impro-Visor application, retaining the musical content and song-specific metadata. We have performed numerous quality checks on the data, have made corrections where required, and have enriched some of the metadata. The re-

© C. Bunks, T. Weyde, S. Dixon, and B. Di Giorgi. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** C. Bunks, T. Weyde, S. Dixon, and B. Di Giorgi, "Modeling Harmonic Similarity for Jazz Using Co-occurrence Vectors and the Membrane Area", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

[1] https://en.wikipedia.org/wiki/Rhythm_changes
[2] https://www.cs.hmc.edu/~keller/jazz/improvisor/

sulting corpus and the code we used to generate our examples is available on GitHub.[3] The Impro-Visor corpus provides chord progressions for 2,612 songs, and is the largest digital collection of jazz chord progressions we know of. For comparison, the applications iRealPro[4] and Band-in-a-Box[5] contain chord progressions for roughly 1400 and 226 jazz standards, respectively. The Weimar Jazz Database contains chords for 456 jazz songs.[6]

Of the 134,182 chord symbol instances in the corpus, there are 1,542 unique symbols, of which many are rare, with 20% occurring just once, and 50% fewer than six times. As the corpus consists mainly of jazz standards, there is a preponderance of 7th chords, comprising at least the root, 3rd, 5th, and 7th notes. These types of chords often have additional extensions (9th, 11th, 13th) and chromatic alterations (♭9, ♯9, ♭5, ♯5). A common variation of jazz chords replaces the 7th with a 6th for major7 and minor7 chords. As 7th chords are the basic harmonic unit in jazz [14], and make up 77% of our corpus, they are the focus of our approach to dimensionality reduction described in the next section. Of the remaining chords, 16% are three-note chords (triads), and 7% are drawn from a variety of special types, as shown in Table 1, which provides a list of all the types and their frequencies.

Type	Percentage
7th chords (and extensions)	76.939%
major triads	11.484%
slash chords	4.781%
minor triads	4.320%
sus chords	1.364%
no chord	0.458%
augmented triads	0.392%
major triads add9	0.127%
diminished triads	0.095%
power chords	0.031%
polychords	0.009%

Table 1. Corpus chord types and their frequencies

3. DIMENSIONALITY REDUCTION

Our approach to reducing dimensionality is based on mapping chords to a reduced vocabulary of functionally equivalent symbols (similar to [15]). This is important because 20% of the chords in the corpus occur only a single time (known as *hapax legomena*), and without additional processing, these types of terms would provide no predictive value [16]. Many techniques are used in NLP to better leverage hapax legomena. For example, stemming, lemmatization, and thesauri are all useful. This paper takes a similar approach for harmony, making use of music theory to reduce the dimensionality of chord space. Our method is akin to lemmatization, applying concepts from functional harmony to group similar chords into classes (for example, see [17]). Based on standard practices in jazz [12, 18, 19], we reduce the set of 1,542 chord symbols to 61 chord classes, as detailed in the following sections.

3.1 7th Chord Types

Our choice of base chord types is built on the four-note 7th chords diatonically generated from the major scale, and making up 77% of our corpus. These are the major7 (M), minor7 (m), dominant7 (7), and minor7♭5 (h), where the symbols shown in parentheses are abbreviations we use in this paper. To these we add a fifth base chord type, the diminished7 (o). Combining the five types with the root notes from the 12 pitch classes yields 60 chord classes. Instances of these classes can occur with extensions or alterations, and we map these to the base class without extension/alteration. For example, we map the symbols Cm9 and Cm11 to the Cm7 class; C7♭9, C7♯5, and C13 to the C7 class; and CM7♯11 to the CM7 class. In addition, in accordance with reharmonization practices, we assign chords such as CmM7 to the Cm7 class and C6 to the CM7 class. We also include the symbol *NC* (no chord) to account for the absence of harmony (0.5% of the corpus).

3.2 Other Chord Type Mappings

In the following discussion, we describe a rationale for mapping the remaining 22.5% of the symbols into classes of the five base types defined above. The mapping choices described in the following discussion are imperfect, but they are simple to implement, and we show they are adequate for our application.

3.2.1 Triads

Triads represent 16% of the corpus. As they do not contain a 7th note, mapping them to the base chord types can be ambiguous. For example, a C major triad shares all of its notes with both the CM7 and C7 chords. We attempt to resolve triad ambiguities using principles from tonal harmony and the local harmonic context. Based on the chord following a triad, we decide whether it has a subdominant, dominant, or tonic function [19]. For example, for a major triad, if the root of the following chord is a fifth down and a member of the major7 or minor7 classes we assign the triad to the dominant7 class with the same root. Otherwise, we assign it to its corresponding major7 class. Major triads with an added 9th are handled in the same way. Augmented triads share their notes with dominant7♯5 chords, an alteration of the dominant, and so we map these to the dominant7 class with the same root. Finally, we map all the minor and diminished triads to their corresponding minor7 and diminished7 classes, respectively.

3.2.2 Sus Chords and Slash Chords

Sus chords also have a harmonic function that depends on context [18]. When followed by a dominant7 chord with

[3] https://github.com/carey-bunks/Jazz-Chord-Progressions-Corpus
[4] https://www.irealb.com/forums/showthread.php?12753-Jazz-1350-Standards
[5] https://members.learnjazzstandards.com/sp/biab-jazzstandards/
[6] https://jazzomat.hfm-weimar.de/dbformat/dbcontent.html

the same root, they act like a subdominant and we opt to map them to a minor7 class with a root a fifth above. For example, a G7sus4 would map to a Dm7. Otherwise, they act like a dominant and we map them to the dominant7 class with the same root. Slash chords are chords played over a specific bass note, for example C/G or Dm7/G, where the symbol above (to the left of) the slash is the chord and below is the bass note. If the bass note belongs to the chord above the slash (for example, C/G), it is an inversion. For such cases, we map it to the class of the chord above the slash. Slash chords are also commonly used to represent sus chords. For example, Dm7/G is harmonically equivalent to G9sus4. We map these according to the process for sus chords. For all other slash chords, we map the chord as if the bass note were an extension or alteration of the chord above the slash.

3.2.3 Power Chords and Polychords

Power chords consist of just two notes, a root and a fifth. As they have no 3rd or 7th, they are harmonically ambiguous. With only 42 instances in our corpus, we have opted to map these chords to the no-chord class. With only 12 instances, polychords are also rare. These chords, used mainly by pianists, consist of a lower triad and an upper triad or 7th chord. We map polychords according to their lower structure, interpreting the upper structure as a collection of extensions or alterations.

4. KEY SIGNATURE BASED REPRESENTATION

To make distributional semantics more effective, we transpose all songs to a common key, and represent them in Roman numeral notation. However, transposition requires knowing the correct key of each song, and from extensive manual checking, we know that our database contains a fair number of songs for which the stated key signature is in error. For this reason, we introduce a key signature estimation algorithm, as described in the following section.

4.1 Key Signature Estimation Algorithm

Several authors have proposed key estimation algorithms for music information retrieval tasks [20–24]. However, our objective is not to estimate the key that is cognitively perceived by a listener, but rather a simpler problem, the key signature that minimizes the number of accidentals needed when writing out the song's chords. Some prior work exists for this [25], however, it is based on machine learning models applied to MIDI data for classical music. Our algorithm selects the key signature most consistent with the chord progression. For each chord in a progression, we map it to one of the described 61 classes, and identify all the major scales it could belong to (excluding diminished7 and no chord classes). The major scale that accumulates the most beats is the resulting estimate of the key signature for that song.

Figure 1 provides a concrete illustration of how the key estimation algorithm works for the case of a short chord progression: A7-Dm7-G7-CM7-CM7. Each column of the table represents one measure, and in this example, there is one chord per measure. The column labels correspond to the chords, and each row label is a key signature whose major scale diatonically contains one or more of the chords in the progression. As shown, the A7 chord belongs to D major; the Dm7 chord belongs to B♭, C, and F major; G7 belongs to C major; and CM7 belongs to both C and G major. Presuming four beats per measure, C accumulates the most beats (16), and is the resulting key signature estimate.

Example: 6-2-5-1 Chord Progression

Major Scales	A7	Dm7	G7	CΔ7	CΔ7	Totals
Bb		4				4
C		4	4	4	4	16
D	4					4
F		4				4
G				4	4	8

Figure 1. Illustration of key signature estimation

4.2 Algorithm Evaluation

As already mentioned, there are quite a few songs in our corpus where the key signature is incorrect or in doubt. Nevertheless, it is worthwhile comparing the outputs of our key estimation algorithm with the keys recorded in the corpus. Of the 2,612 songs, the algorithm concurs with the database for 1,763 (67.5%) of them. For the 849 songs with database key signatures that do not agree with our estimates, we use the Circle of Fifths as a distance metric to evaluate the magnitude of differences between the two. Adjacent key signatures on the circle of fifths correspond to major scales that differ in a single pitch class. Table 2 shows the distribution of circle-of-fifths distances between estimated and database key signatures for all of the songs in the corpus. The first row is the distance in number of sharps or flats from the estimated to the database key, where 0 corresponds to agreement. The last column of Table 2 is labelled "Amb." for ambiguous. There are 123 songs in the database for which the key estimation algorithm returns a non-unique result, finding two or more equally good major scales. This occurs for 4.7% of the songs in the corpus, and when it does our estimation algorithm defaults to the database key.

Dx	6♭	5♭	4♭	3♭	2♭	1♭	0	1♯	2♯	3♯	4♯	5♯	Amb.
Frq	10	22	33	55	99	304	1763	183	22	25	12	1	123

Table 2. Key signature estimation statistics with the circle of fifths distance **Dx** by the frequency of occurrence **Frq**

4.3 Mapping to Roman Numeral Notation

Once a song's key has been estimated, all the chords in its progression can be mapped to Roman numeral notation. Table 3 shows the Roman numerals corresponding to chord roots for C major. As an example, the sequence of chords A7-Dm-G7-CM maps to vi7-iim-v7-iM. In our system, we

represent minor keys by their relative major, so the relative minor cadence, Bm7b5-E7-Am7, maps to viih-iii7-vim.

Root	C	D♭	D	E♭	E	F	G♭	G	A♭	A	B♭	B
RN	i	♭ii	ii	♭iii	iii	iv	♭v	v	♭vi	vi	♭vii	vii

Table 3. Roman numeral notation: chord roots in C major

5. VECTOR REPRESENTATION

Sections 3 and 4 described our approach for reducing the dimensionality of chord space, distilling the 1,542 chord symbols in our corpus to 61 classes. In this section we describe our method for embedding the chord classes into a vector space. Our design objective is that common reharmonizations be close to each other in cosine similarity, and it is known that the co-occurrence matrix can capture this type of characteristic [5, 26–28].

Given a corpus of D chord progressions, with progression $d \in \{1, 2, \ldots, D\}$ containing N_d chords with indices $1, 2 \ldots, N_d$, we can represent the corresponding sequence of chord symbols as $s_{d,1}, s_{d,2}, \ldots, s_{d,N_d}$. We define the symmetric, sliding context window, $W_{k,d}$, of nominal width N_w with the indices $W_{k,d} = [w_l, \ldots, (k-1), (k+1), \ldots, w_r]$, where the left and right endpoints are $w_l = \max(k - N_w, 1)$ and $w_r = \min(k + N_w, N_d)$, respectively. With these definitions, the $(i, j)^{\text{th}}$ element of the co-occurrence matrix, $C_{i,j}$ is computed by

$$\mathbf{C}_{i,j} = \sum_{d=1}^{D} \sum_{k=1}^{N_d} \sum_{w \in W_{k,d}} \begin{cases} 1, & \text{if } s_{d,k} = c_i \text{ and } s_{d,w} = c_j \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

This produces a square, symmetric matrix whose row $\mathbf{C}_i$ (or alternatively, column) is a vector representations of the ith chord class c_i. As it will be useful in the following, we normalize each row to have unit length. Because co-occurrence matrices capture contextual information, the vectors of chord classes that have similar harmonic function are expected to be close to each other with respect to the cosine similarity measure, and this seems to be borne out by an inspection of certain chord vectors. For example, of 60 chord classes, the closest vector to the v7 is its tritone substitute, the ♭ii7, and the closest to the iim is the iih, a common substitute from the parallel minor scale (see modal interchange in [19]).

6. MEMBRANE-AREA DISTANCE METRIC

We use the co-occurrence vectors to represent chord progressions in a way that represents each chord type, duration, and metric position, while being robust to reharmonizations. The normalized chord vectors derived from the co-occurrence matrix can be used to plot the path of a song's progression through 61-dimensional space. Starting from the origin, the sequence of chord vectors can be concatenated from head to tail, beginning with the first, and terminating with the last vector (see Figure 2). Each unit vector is scaled by the number of beats of the chord it represents, and the result is a piecewise linear function through $\mathbf{R}^{61}$. The comparison of two songs in this space can be formulated as a trajectory comparison problem, for which there are many existing techniques [29]. The most popular ones, however, are not well adapted to our problem. The Fréchet distance, dynamic time warping, longest common subsequence, and the edit distance are all based on matching and comparing points, and would not directly factor in information about reharmonized chords embodied in the co-occurrence vectors. For this reason, we introduce a new metric that accounts for reharmonizations by computing the membrane area between the paths of two songs.

Expressed formally, we represent song vector paths by piecewise linear functions of the form $\boldsymbol{f}(t) \in \mathbf{R}^{61}$, where $t \in [0, 1]$ is a parametric variable representing the number of normalized beats traversed in the song. We can move along the entire length of $\boldsymbol{f}$ in discrete, equal increments, dt, where the starting point of the function, $\boldsymbol{f}(0)$ at $t = 0$ is the origin, and the end point of the function is at $t = 1$. Given two songs and their corresponding piecewise linear functions, $\boldsymbol{f}(t)$ and $\boldsymbol{g}(t)$, and letting $K = 1/dt$, we can define a distance metric between them as the area of a 2D membrane, M, stretched between the two paths. M is calculated as the integral obtained in the limit of

$$M(\boldsymbol{f}, \boldsymbol{g}) = \lim_{dt \to 0} \sum_{k=0}^{K} \|\boldsymbol{f}(kdt) - \boldsymbol{g}(kdt)\| dt, \tag{2}$$

where $\| \cdot \|$ is the Euclidean norm. The piecewise linear functions for two identical chord progressions would, naturally, overlay each other, yielding a membrane area of zero. Two harmonically similar songs should trace out similar paths keeping the membrane area small. For example, two chord progressions that differ in just a tritone substitution will only slightly perturb the path and the membrane area between songs. Figure 2 is a notional illustration of how the measure in Equation 2 is evaluated. The red and blue paths represent two different songs, each having three chords. Each song begins at the origin, and the chord vectors are added head-to-tail to trace out a piecewise linear path. The membrane area metric is approximated by summing the lengths of the N equally spaced black line segments drawn between the two songs. Note that this way of representing the harmony of a song accounts for positions and durations of each chord in the progression, as well as capturing harmonic similarities of chord transitions.

7. EXPERIMENTS

We have designed some experiments based on a set of jazz contrafacts listed in a Wikipedia article. [7] The list has 252 jazz songs whose harmonies are known to be based on other songs (see also [30]). A subset of 91 contrafacts are available in our corpus, but for 11 of them, only a section of the harmony is borrowed, and we remove these from the list. The basic structure of all of our experiments is the same: for each contrafact, we compute the membrane area

[7] https://en.wikipedia.org/wiki/List_of_jazz_contrafacts

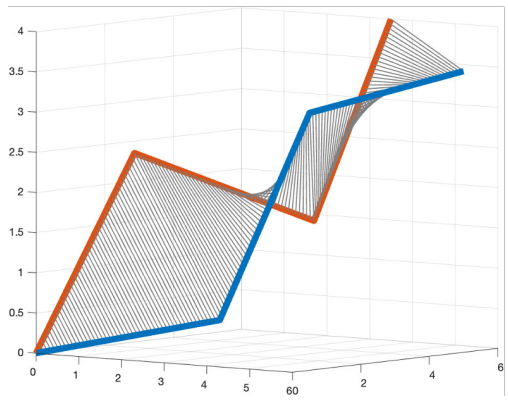


Figure 2. Conceptual illustration of the membrane-area distance metric for two, 3-chord sequences

distance between it and each of the other 2,611 songs in our corpus. We then sort the songs from smallest membrane area to largest, and note the original song's rank in that list. Because of reharmonizations, we don't expect the membrane area to be zero for all contrafact-original pairs, but matches should rank high in the list. Original songs often inspire multiple contrafacts, and some may be closer to each other than to the original. For these reasons, we use the histogram of original song rankings to present the overall performance of our method, and we use the median rank as a method of comparison between approaches.

7.1 Using Co-Occurrence Vectors

We evaluated six variants of our approach using co-occurrence chord vectors. The first three were based on the context window widths $N_w = [1, 2, 3]$. The second three variants used the same context window values, but applied to a filtered version of the chord progressions. For each chord progression, the filter collapses adjacent identical chords to a single instance. For $N_w = 1$, this has the effect of eliminating the co-occurrence of chords with themselves, making the diagonal of the co-occurrence matrix zero. Of the six versions, the best result was obtained for the filtered chord progressions with the context window width $N_w = 1$. Figure 3 shows the histogram of original song rankings for this case. The median rank is 18, meaning that half of the original songs rank in the top 0.7% in harmonic similarity to their contrafacts. As there is some histogram mass out to rank 1,382, the histogram makes use of a log-scale on the x-axis. It is likely that some of the songs ranking better than the original are also contrafacts, as the Wikipedia list is far from exhaustive, but it would require substantial effort and expertise to evaluate this.

As noted, some original songs have inspired many contrafacts. As an example of this in our corpus, there are four known contrafacts of the song *All the Things You Are*. The ranks and membrane areas of the original song for each contrafact are shown in Table 4. The original ranks highly for three of the four contrafacts in the table. As the chord progressions for *Prince Albert* and *All the Things You are* are identical, their membrane area is zero. The contrafacts

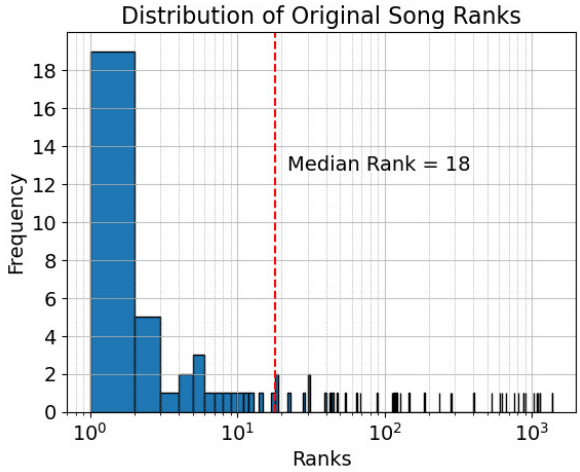


Figure 3. Histogram of original song ranks for 80 contrafacts (median rank = 18)

Ablution and *Boston Bernie* have some chord substitutions, and the original song ranks highly for both of them. The song *I Want More*, however, does quite poorly, with a rank of 758th out of the 2,611 songs in our corpus.

Contrafact	Rank	Membrane Area
Prince Albert	1	0.00
Ablution	1	6.72
Boston Bernie	2	7.72
I Want More	758	26.89

Table 4. Rank and membrane area for *All the Things You Are* against its four contrafacts

To investigate, we use the jazz harmony visualization tool described in [31] to display the chord progressions for these two songs. The visualization shows a tabular format with each rectangle representing a measure. Figures 4 and 5 show *All the Things You Are* and *I Want More*, respectively. The background colors indicate the key the chords belong to. Red is for the main key, which is A♭ for both songs. Other colors indicate modulations. Some chords are embedded in a geometric shape to indicate they are tonicizations: diamonds are secondary dominants, pentagons are borrowed chords. As the figures illustrate, the two songs have some similar chords, however, the sequences of modulations are completely different. Whereas *All the Things You Are* modulates through the tonal centers of C major, E♭ major, G major, and E major, *I Want More* modulates to D♭ major and C minor. After verifying the latter's chord progression, [8] we conclude that, harmonically, these two songs have very little in common, and we question the annotation of this song as a contrafact.

7.2 Using Pitch-Class Vectors

To evaluate the effect of using co-occurrence vectors, we compare with a baseline vector embedding scheme based

[8] Jamey Aebersold play-along book, volume 82, Dexter Gordon

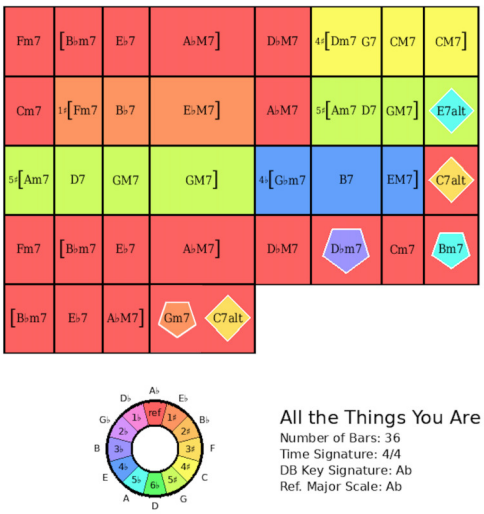


Figure 4. Chord Progression for *All the Things You Are*

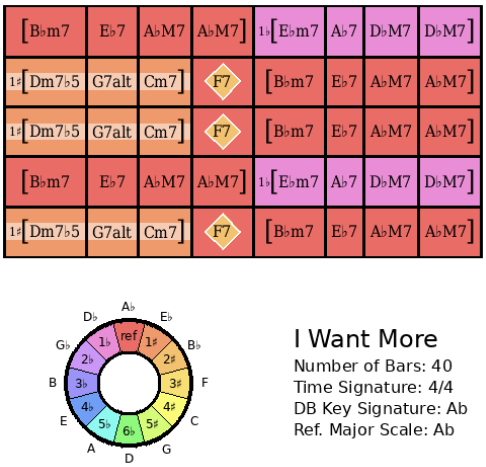


Figure 5. Chord progression for *I Want More*

on converting chord symbols to their pitch-class vectors. This is similar to the starting point of the approach used in [32]. We begin by applying the key estimation algorithm described in Section 4.1 to transpose all chords in our corpus to the key of C. Subsequently, each chord in the corpus is converted to a 12-dimensional binary pitch-class vector, with ones in positions corresponding to pitch classes belonging to the chord, and zeroes elsewhere. Thus, for a C7 chord with the notes C, E, G, and B♭, the corresponding pitch-class vector is $[1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0]$.

Following a similar schema as for the previous experiment, the pitch-class vectors can be used to construct piecewise linear paths, however, now they are constructed in a 12-dimensional space. We use the membrane area as previously to rank songs by harmonic similarity. Table 5 compares the performance of co-occurrence vectors for the best case (chord progression filtering with a window size of $N_w = 1$) versus pitch-class vectors using three metrics: median rank, mean rank, and mean reciprocal rank. Co-occurrence vectors outperform the pitch-class vectors by a large margin for each of these criteria.

Vector Type	Median	Mean	MRR
Co-occurrence	18	222	0.305
Pitch-class	318	457	0.200

Table 5. Comparison of median rank, mean rank, and mean reciprocal rank (MRR) for the filtered-progression, co-occurrence vectors ($N_w = 1$) and pitch-class vectors

8. DISCUSSION AND CONCLUSIONS

We showed how co-occurrence vectors can be used to model harmonic similarity, and introduced the membrane area as a evaluation metric that is well-adapted for handling reharmonizations. We use music theory to reduce the dimensionality of chord space, and provide a comprehensive map of all 1,542 chord symbols in our corpus to 61 classes. The results are used to compute a dense co-occurrence matrix without needing to resort to non-parametric approximations such as truncated SVD or gradient descent. Using the cosine similarity measure, we show that the rows of the co-occurrence matrix embody some characteristics of common reharmonizations. Using the normalized rows of the matrix as vector embeddings of chord classes, we modeled songs as piecewise linear paths in $\mathbf{R}^{61}$. A novel distance metric, the membrane area, was introduced and used as a measure of harmonic similarity between songs. We showed that the similarity metric can be used to retrieve contrafacts from a database of jazz standards, and that it performs significantly better than a baseline system using binary pitch-class vectors as chord embeddings.

Although our approach is successful for contrafact detection, there are several weaknesses that require future work. Our key detection algorithm is simple and static, despite the fact that jazz harmony exhibits many local key changes (e.g. see Figures 4 and 5). We also treat minor keys as equivalent to their relative major, which is not strictly correct. The chord mapping scheme is limited in its ability to distinguish common progressions such as triad progressions i-iv and v-i. A richer chord vocabulary or local key estimation could disambiguate such situations. Our song-level similarity assumes only minor structural differences between pieces. Modifying it to perform subsequence matching would overcome this limitation.

We believe that the methods discussed in this paper have many additional applications, such as those in evaluating harmonic complexity [33] and in musicology [34]. We intend to investigate whether our harmonic similarity measure can be used to cluster jazz songs by composer or decade of publication. Although our focus has been on jazz, chords have similar functions across much of Western tonal harmony. For this reason, we believe that this work can be adapted to other genres such as classical, rock, and pop. Furthermore, as our methods are based on capturing the distributional semantics of harmony, the approach may also be useful in discovering harmonic relationships in non-Western music genres.

9. ACKNOWLEDGEMENTS

The first author 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].

10. REFERENCES

[1] D. Ponsford, G. Wiggins, and C. Mellish, "Statistical learning of harmonic movement," *Journal of New Music Research*, vol. 28, no. 2, pp. 150–177, 1999.

[2] W. B. De Haas, "Music information retrieval based on tonal harmony," Ph.D. dissertation, Utrecht University, 2012.

[3] 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.

[4] D. Zahnd, "Similarity analysis of jazz tunes with vector space models," Ph.D. dissertation, Hochschule für Musik, Freiburg, 2022.

[5] K. Lund and C. Burgess, "Producing high-dimensional semantic spaces from lexical co-occurrence," *Behavior Research Methods, Instruments, & Computers*, vol. 28, no. 2, pp. 203–208, 1996.

[6] E. Bruni, N.-K. Tran, and M. Baroni, "Multimodal distributional semantics," *Journal of Artificial Intelligence Research*, vol. 49, pp. 1–47, 2014.

[7] C. O. S. Sorzano, J. Vargas, and A. P. Montano, "A survey of dimensionality reduction techniques," *arXiv preprint arXiv:1403.2877*, 2014.

[8] B. D. Kernfeld, *The New Grove Dictionary of Jazz*. Grove's Dictionaries Incorporated, 2002, vol. 2.

[9] H. Martin, *Charlie Parker, Composer*. Oxford University Press, USA, 2020.

[10] D. H. Rosenthal, *Hard Bop: Jazz and Black Music 1955-1965*. Oxford University Press, 1994.

[11] T. Owens, *Bebop: The Music and its Players*. Oxford University Press, 1996.

[12] D. Berkman, *The Jazz Harmony Book: A Course in Adding Chords to Melodies*. Sher Music Co., 2013.

[13] H. Leonard, *The Real Book*. Hal Leonard Publishing Corporation, 2016.

[14] R. Rawlins and N. E. Bahha, *Jazzology: The Encyclopedia of Jazz Theory for All Musicians*. Hal Leonard Corporation, 2005.

[15] X. Serra, "Audio-aligned jazz harmony dataset for automatic chord transcription and corpus-based research," in *in Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2018, pp. 483–490.

[16] J. Pierrehumbert and R. Granell, "On hapax legomena and morphological productivity," in *Proceedings of the Fifteenth Workshop on Computational Research in Phonetics, Phonology, and Morphology*, 2018, pp. 125–130.

[17] D. Harasim, C. Finkensiep, P. Ericson, T. J. O'Donnell, and M. Rohrmeier, "The jazz harmony treebank," in *in Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2020, pp. 207–215.

[18] M. Levine, *The Jazz Theory Book*. Sher Music Co., 1995.

[19] J. Mulholland and T. Hojnacki, *The Berklee Book of Jazz Harmony*. Berklee Press, 2013.

[20] M. Mauch and S. Dixon, "Simultaneous estimation of chords and musical context from audio," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 6, pp. 1280–1289, 2009.

[21] J. Pauwels and J.-P. Martens, "Combining musicological knowledge about chords and keys in a simultaneous chord and local key estimation system," *Journal of New Music Research*, vol. 43, no. 3, pp. 318–330, 2014.

[22] T. Rocher, M. Robine, P. Hanna, L. Oudre, Y. Grenier, and C. Févotte, "Concurrent estimation of chords and keys from audio," in *in Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2010, pp. 141–146.

[23] K. C. Noland and M. B. Sandler, "Key estimation using a hidden Markov model," in *in Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2006, pp. 121–126.

[24] E. Benetos, A. Jansson, and T. Weyde, "Improving automatic music transcription through key detection," in *Audio Engineering Society Conference*. Audio Engineering Society, 2014.

[25] F. Foscarin, N. Audebert, and R. Fournier-S'Niehotta, "PKSpell: Data-driven pitch spelling and key signature estimation," *arXiv preprint arXiv:2107.14009*, 2021.

[26] S. Bordag, "A comparison of co-occurrence and similarity measures as simulations of context," in *International Conference on Intelligent Text Processing and Computational Linguistics*. Springer, 2008, pp. 52–63.

[27] A. Globerson, G. Chechik, F. Pereira, and N. Tishby, "Euclidean embedding of co-occurrence data," *Advances in Neural Information Processing Systems*, vol. 17, 2004.

[28] L. Leydesdorff and L. Vaughan, "Co-occurrence matrices and their applications in information science: Extending ACA to the web environment," *Journal of the American Society for Information Science and Technology*, vol. 57, no. 12, pp. 1616–1628, 2006.

[29] K. Toohey and M. Duckham, "Trajectory similarity measures," *Sigspatial Special*, vol. 7, no. 1, pp. 43–50, 2015.

[30] F. Tirro, "The silent theme tradition in jazz," *The Musical Quarterly*, vol. 53, no. 3, pp. 313–334, 1967.

[31] C. Bunks, T. Weyde, A. Slingsby, and J. Wood, "Visualization of tonal harmony for jazz lead sheets," in *24th EG Conference on Visualization (EuroVis) Short Papers*, 2022, pp. 109–113.

[32] 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 (NeurIPS)*, 2016.

[33] B. Di Giorgi, S. Dixon, M. Zanoni, and A. Sarti, "A data-driven model of tonal chord sequence complexity," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 11, pp. 2237–2250, 2017.

[34] A. Moore, "Patterns of harmony," *Popular Music*, vol. 11, no. 1, pp. 73–106, 1992.

SINGSTYLE111: A MULTILINGUAL SINGING DATASET WITH STYLE TRANSFER

Shuqi Dai[1] **Yuxuan Wu**[1] **Siqi Chen**[2] **Roy Huang**[1] **Roger B. Dannenberg**[1]

[1] Computer Science Department, Carnegie Mellon University , USA
[2] University of Southern California, USA

shuqid@cs.cmu.edu, rbd@cs.cmu.edu

ABSTRACT

There has been a persistent lack of publicly accessible data in singing voice research, particularly concerning the diversity of languages and performance styles. In this paper, we introduce SingStyle111, a large studio-quality singing dataset with multiple languages and different singing styles, and present singing style transfer examples. The dataset features 111 songs performed by eight professional singers, spanning 12.8 hours and covering English, Chinese, and Italian. SingStyle111 incorporates different singing styles, such as bel canto opera, Chinese folk singing, pop, jazz, and children. Specifically, 80 songs include at least two distinct singing styles performed by the same singer. All recordings were conducted in professional studios, yielding clean, dry vocal tracks in mono format with a 44.1 kHz sample rate. We have segmented the singing voices into phrases, providing lyrics, performance MIDI, and scores with phoneme-level alignment. We also extracted acoustic features such as Mel-Spectrogram, F0 contour, and loudness curves. This dataset applies to various MIR tasks such as Singing Voice Synthesis, Singing Voice Conversion, Singing Transcription, Score Following, and Lyrics Detection. It is also designed for Singing Style Transfer, including both performance and voice timbre style. We make the dataset freely available for research purposes. Examples and download information can be found at `https://shuqid.net/singstyle111`.

1. INTRODUCTION

In recent years, deep learning technologies have significantly advanced the field of Artificial Intelligence Generative Content (AIGC) [1], leading to breakthroughs in Computer Vision for image synthesis and manipulation [2–5], Natural Language Processing (NLP) for text generation and summarization [6–8], and audio signal processing for Text-to-Speech (TTS) generation [9–11]. In particular, advanced generative models such as Variational Autoencoders (VAEs) [12–14], Generative Adversarial Networks (GANs) [15, 16], Transformer-based models [17, 18], and Diffusion Models [19, 20] resulted in a series of exceptional TTS models that achieve not only realistic results [9–11, 21] but also explore stylistic and emotional speech synthesis [22, 23] in a more controllable way. However, the development of singing tasks such as Singing Voice Synthesis (SVS) [24–28] and Singing Voice Conversion (SVC) [29] have yet to progress as fast as TTS. One primary reason is the lack of data on several key aspects:

- Lack of high-quality data. Tasks such as SVS and SVC require monophonic, clean, and dry sound singing data with studio quality. Unfortunately, due to the limitations of Source Separation and Denoising technologies [30–33], as well as copyright issues, most available cover songs online cannot meet these quality requirements. Datasets recorded with studio quality are predominantly composed of amateur performances, which often exhibit off-key and cracking issues that could mislead the generative models and diminish their quality.
- Lack of diversity. Most available singing datasets cover only one language, resulting in a severely imbalanced language distribution. For example, there is a fair amount of Chinese singing data, while clean English data is very scarce. In addition, most datasets only focus on one pop singing style, and the distributions of different singing styles and vocal ranges are too narrow.
- Lack of annotations. Many datasets lack proper phrase-level segmentation, lyrics, and scores, and are not aligned at the phoneme level, making it impossible to conduct score-based SVS and more detailed performance control.
- Lack of large-scale data. The current data volume of high-quality singing is still insufficient for deep generative models.

Furthermore, current SVS results are primarily confined to modeling the timbre of singing voices. While there are several good vocoders [11, 21, 34] and acoustic models [10, 35] for SVS based on Ground-Truth control signals (e.g., inputting F0 control signals to the model), the truly creative and artistic aspects of singing, such as expressive performance control, singing styles, vocal techniques, and creative improvisation, have yet to be explored. Again, data limitations play a significant role in this, as most datasets consist of amateur performances or have not

© S. Dai, Y. Wu, S. Chen, R. Huang and R. B. Dannenberg. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** S. Dai, Y. Wu, S. Chen, R. Huang and R. B. Dannenberg, "SingStyle111: A Multilingual Singing Dataset With Style Transfer", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

Dataset	Language	Style	#Hour	#Singer	Quality	Musicality	Score	Align-ment	Style Transfer
Opencpop [41]	Chinese	Pop	5.25	1	Studio	Ama.	Perform. MIDI	✓	✗
M4Singer [42]	Chinese	Pop	29.77	20	Studio	50% Ama. 50% Prof.	Perform. MIDI	✓	✗
Children Song [43]	Korean English	Children	4.86	1	Studio	Prof. but plain	Perform. MIDI	word	✗
Tohoku Kiritan [44]	Japanese	Pop	0.95	1	Studio	Prof.	Score	✓	✗
PopCS [28]	Chinese	Pop	5.89	6	Not Clean	Ama.	✗	✗	✗
Open-Singer [35]	Chinese	Pop	50	66	Studio	Ama.	✗	✗	✗
VocalSet [45] Annotated [46]	Five Vowels	Opera	10.1	20	Studio	Prof.	Score	✓	technique transfer
NHSS [47]	English	Pop	3.5	10	Studio	Ama.	✗	✓	✗
NUS-48E [48]	English	Pop Children	1.41	12	Studio	Ama.	✗	✓	✗
RWC [49]	Japanese English	Pop	4	27	Not Solo	Prof.	Both	✗	✗
TONAS [50]	Spanish	Flamenco	0.34	> 40	Not Clean	Prof.	✗	✗	✗
Vocadito [51]	Seven Languages	Pop Children	0.23	29	Not Clean	Ama.	✗	✗	✗
MIR-1K [52]	Chinese	Pop	2.22	19	Not Solo	Ama.	✗	✗	✗
StyleSing111 (Ours)	English Chinese Italian	Opera Pop Folk Jazz etc.	12.8	8	Studio	Prof.	Both	✓	✓

Table 1. A comparison of existing singing datasets. Score means if there is score or performance MIDI file provided. "Perform. MIDI" stands for "Performance MIDI". "Both" means both performance MIDI files synchronized with the singing audio and sheet music scores are provided. Alignment means whether or not there is duration annotation at the phoneme level for lyrics. "Ama." stands for "Amateur," and "Prof." stands for "Professional."

yet begun to address the issue of artistic expression.

For example, Style Transfer [36, 37] is a popular technique in deep learning that combines the content of one image or sound with the style of another. For audio processing, some researchers [38, 39] have recently transferred the timbre from one audio source to another while preserving the speech content (similar to SVC). However, the transfer of expressive performance styles embedded below the timbre level remains elusive, mainly because (1) disentangling performance style is much more challenging than timbre features [40] and (2) the scarcity of relevant datasets providing examples of performance styles.

To help address these issues, we introduce a new singing corpus, SingStyle111. We summarize the main contributions as follows:

(1) SingStyle111 is a large and high-quality singing dataset. It contains 111 songs performed by eight professional singers, spanning 12.8 hours of clean monophonic vocal recordings in studio quality.
(2) It is a diverse dataset with creative singing. It covers English, Chinese, and Italian songs and incorporates various singing styles, such as bel canto opera, Chinese folk, pop, jazz, and children. Some performances are creative improvisations based on the original score.
(3) It demonstrates style transfer in both performance and timbre levels. 80 songs contain at least two distinct singing styles performed by the same singer.
(4) It includes proper annotations and extracted features. We manually segmented voices into phrases, labeled Performance MIDI files and music score notes and aligned them with the phonemes of lyrics, extracted acoustic features such as Mel-Spectrogram, F0 contour, and loudness curves.
(5) It applies to different MIR tasks such as SVS, SVC, Singing Transcription, Score Following, Expressive Performance, Lyrics Detection, Singing Style Transfer.
(6) It is publicly available for research purposes for free.

The rest of this paper is organized as follows: after a brief review of related works, we describe how we collect and process the dataset (Section 3) and show the annotations and analysis (Section 4). Finally, we discuss potential applications in Section 5 followed by conclusions.

2. RELATED WORK

Existing singing voice datasets still have many limitations in fulfilling the requirements for singing research tasks such as Singing Voice Synthesis (SVS) [24, 26–28] and Singing Voice Conversion (SVC) [29]. Table 1 provides an overview of the available public datasets. Datasets such as MIR-1K [52], TONAS [50], and Vocadito [51] are restricted by the absence of separated solo vocal tracks or suffer from subpar recording environments with noise, reverberation, and other interferences. These issues hinder their usability in SVS-related tasks. While NHSS [47] and OpenSinger [35] contain clean and dry human vocals, they lack essential musical scores or phoneme-level duration alignment. Consequently, these datasets are unsuitable for training end-to-end synthesis models that convert scores to vocals. Moreover, datasets such as Opencpop [41] and M4singer [42] offer good annotations and recording quality but primarily focus on Mandarin songs and a limited range of pop styles. Additionally, the singing proficiency of performers is inconsistent, with many being amateurs, which affects the overall quality of the dataset.

Another issue that has long been overlooked and misunderstood in singing voice datasets is the difference between Performance MIDI and the actual sheet music score. In Table 1, only Tohoku Kiritan [44], Vocalset [45, 46] and RWC [49] have music scores, while other datasets claimed to have scores that are indeed performance MIDI files. Performance MIDI features expressive performance timings rather than score timings with regular note durations in beats. The melodic pitches in performance MIDI can also differ from those in score melody. Utilizing performance MIDI for singing voice synthesis and claiming it as score-based is, in reality, a deceptive approach that takes advantage of real singing data.

As for the Style Transfer task, Vocalset [45, 46] provides relevant examples, but its scope is limited to singing technique transfer within the bel canto singing style. Furthermore, the dataset predominantly consists of scale exercises using only five vowels and includes only three short songs, which restricts its applicability. Given the limitations of existing datasets, there is a need for a large-scale, high-quality, professional, multilingual, and diverse singing dataset that caters to various styles and includes style transfer examples. In this paper, we introduce a novel dataset designed to address these requirements and facilitate research in SVS-related tasks and style transfer.

3. DATASET DESCRIPTION

3.1 Overview

SingStyle111 is a multilingual singing dataset with style transfer demonstrations. Figure 1 illustrates the data collection pipeline. Following the completion of the recording process, we post-process all recordings and retain all high-quality segments. Thus, our dataset offers two versions: the first version consists of edited full-length songs, and the second version comprises all usable, high-quality vocal segments, incorporating redos from the recording process.

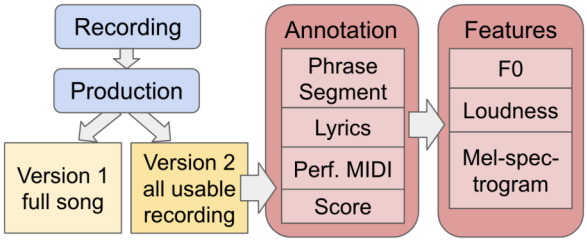


Figure 1. Data collection pipeline.

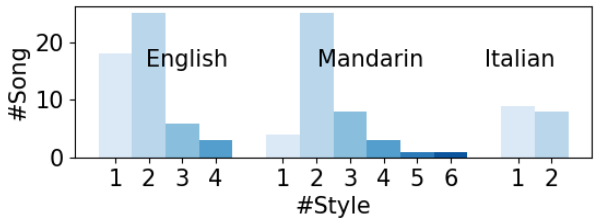


Figure 2. Distribution of songs according to languages and the number of style demonstrations. For example, English songs have 18 songs with only one style version, 25 songs with two different styles, six songs with three styles, and three songs with four styles.

We preserve these redos for two primary reasons. First, during recording, singers often need to restart due to minor errors, resulting in many redos that far exceed the quantity required for a single song. The high-quality vocals in these redo segments are perfect for segmented training in deep learning and effectively augmenting the dataset. Second, even when the same singer performs the same song using the same style, each rendition exhibits subtle differences. Capturing these variations provides valuable training data for learning multi-modes in singing performance and disentangling a singer's style with music content. This paper focuses on describing the second version of the dataset.

Upon obtaining the clean and dry vocal segments in audio, we manually annotate them into phrases (music sentences), provide lyrics and score alignment with audio at the phoneme level. We then extract acoustic attributes such as F0 contour, loudness curve, and Mel-spectrogram. Finally, we partition and package the data, incorporating relevant attributes. Section 4 describes this process in detail.

In the following subsections, we delve into the dataset's repertoire and styles, singer profiles, recording environments, and post-production methods, accompanied by pertinent statistics.

3.2 Repertoire and Style

SingStyle111 comprises 111 songs, of which 80 have at least two different versions performed in distinct styles by the same singer, resulting in a total of 224 song versions. The dataset encompasses three languages: English (372 minutes), Chinese Mandarin (307 minutes), and Italian (88 minutes). Figure 2 illustrates the number of song versions for each language. During song selection, we sought to diversely represent various styles, singing techniques, tempos, and eras.

Figure 3 presents the styles of the original songs and all

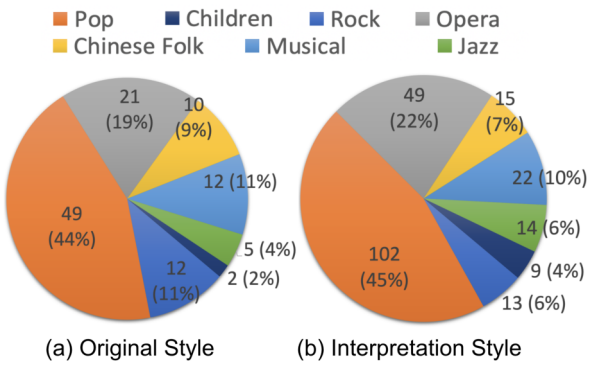


Figure 3. Distribution of song styles. Chart(a) describes the original style of the 111 songs, while chart(b) indicates the 224 different style interpretations in the dataset.

style demonstrations. We consolidated several sub-genres into seven broader styles to streamline the pie chart. For instance, Country, Western folk, Chinese pop, and other pop styles were combined into a single pop genre. Likewise, the Rock category contains Soft Rock, Hard Rock, Alternative Rock, etc.

Throughout the data collection process, we instructed singers to exhibit significant differences in style transfer. Sometimes they made appropriate adaptations or improvisations to the original song for better style transfer while preserving the original lyrics, melody, and structure. For example, it is easier for singers to transfer vocal timbres when the key changes. Also, tempo changes and rhythmic variations can dramatically help alter styles, such as transferring a fast and happy song into a slow and melancholic one. Converting singing techniques or adding ornamentations are also prevalent in our style transfer examples. For instance, the dataset includes many demonstrations interchanged among pop, bel canto, and Chinese traditional folk singing; or singing the same song in the distinct pop styles of Adele Adkins and Teresa Teng. In addition, some styles include deliberate emotional changes, for example, contrasting a "plain and lyrical style" with an "exaggerated and highly emotional style."

3.3 Singers

We paid eight professional singers (Table 2) to sing the songs. They have diverse vocal ranges, singing styles, and vocal techniques. They are aged 20 to 63, and all have received formal musical training for more than six years. Six of them are graduates or current students in the voice major at music conservatories. “Male1” is a native American English speaker, and all the others are Chinese. “Female1” has lived in the US for more than five years and received formal English singing training at a music academy. We also removed the English song phrases that have strong foreign accents. All singers have signed agreements to release the dataset for research purposes.

3.4 Recording

We recorded the songs in a professional recording studio with little reverberation or noise. We use a Shure Model SM81-LC microphone, an Apollo X8 Thunderbolt 3 audio interface, Heritage Audio 73jr as the pre-amplifier, and Pro Tools Studio as DAW software. All singings are pure vocal only and recorded at 44,100 Hz sampling rate with 24 bits per sample in wav format.

In most recording sessions, singers wear headphones to listen to the accompaniment. However, in some style transfer demonstrations, accompaniments and headphones may not always be used. Despite this, singers must ensure they maintain the correct key and consistently stay within it throughout the performance.

Singer	Language	Style	#Hour	Range
Female1	en, cn	P. C. O. R. F. M. J.	3.73	F#3-A5
Female2	it, en, cn	O. F. M.	1.24	E4-C6
Female3	cn, en	P. C. O. R. F. M. J.	1.58	F#3-F5
Female4	cn, en	P.	1.63	D3-C5
Male1	en	P. R. M.	0.59	D2-G4
Male2	cn, en	P. M. J.	1.35	A2-C5
Male3	it, cn	O. M.	1.16	C4-G5
Male4	cn	P. O. F.	1.51	D#3-A4

Table 2. Singer Information. Here the vocal range is the used range in the dataset. en: English, cn: Chinese, it: Italian, P: Pop, C: Children, R: Rock, O: Opera, F: Chinese Traditional Folk, M: Musical, J: Jazz.

3.5 Production

We employed several essential post-production techniques to refine and clean the recorded data. First, we edited the raw recordings to retain only high-quality clips, filtering out noisy sections, mistakes, and mispronunciations. A small portion of singer Male3’s singing clips were further edited with pitch-tuning. To achieve a consistent volume balance, we applied different gain levels in each clip. Moreover, we incorporated a compressor for all recording clips to prevent extreme dynamic fluctuations. Lastly, we maximized the output volume using a limiter, setting the output ceiling at -0.6 dB. After production, we obtained clean and dry vocal tracks with similar output volumes.

4. ANNOTATION AND ANALYSIS

This section presents the annotation process, including both manual annotation and automatic analysis. We first segment the audio clips into music phrases, for which we then manually identify corresponding lyrics and music scores. By combining automatic algorithms and manual efforts, we align lyrics phonemes and score notes to their corresponding audio. Next, we utilize algorithms to extract acoustic attributes such as F0 contour, loudness curve, and Mel-spectrogram. Finally, we highlight the key attributes and explain dataset partitioning and packaging.

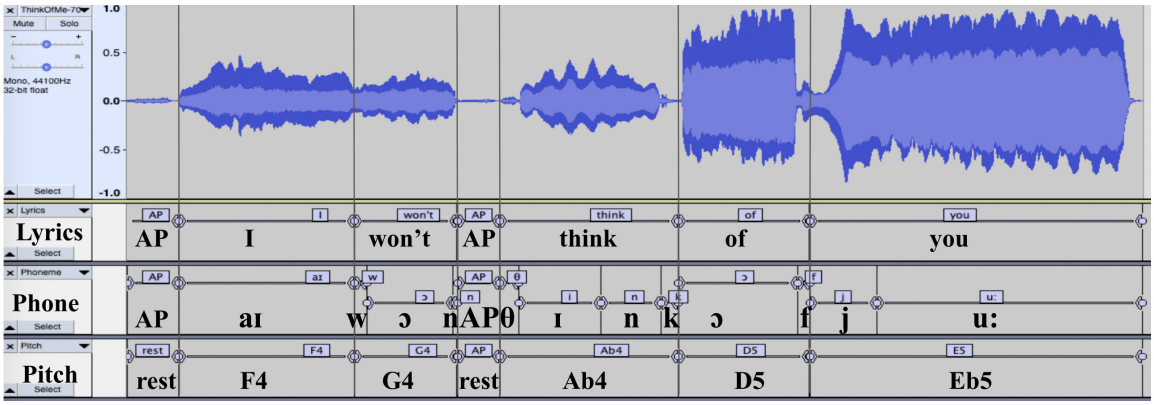


Figure 4. An example of phoneme-level annotation using Audacity. The lyrics word, IPA phoneme, and pitch label tracks are aligned with the corresponding audio. "AP" here stands for "aspirate".

4.1 Phrase-level Segmentation

We further divide the audio segments into smaller musical phrases for two reasons. First, this additional segmentation accelerates model training. Second, from a music perspective, phrases serve as one of the basic music structure units, with emotional expressions and performance controls being highly related to phrase-level structure. Inappropriate segmentation might compromise musical expression due to insufficient phrase structure information. Given the low accuracy of automatic algorithms for phrase segmentation, we manually label them. The result shows that most segmented phrases have lengths between 3 and 12 seconds with one to three breaths. We obtain 6588 phrases in total. No silence is included at the beginning or end of the phrases, except for breath events.

4.2 Lyrics and Score Alignment at Phoneme Level

In this subsection, we describe the lyrics and score annotation process.

Lyrics annotation We first manually find lyrics for each song online and then segment and align the lyrics with each phrase. We manually correct the lyrics to match the actual singing in the data. Secondly, we employ algorithms to translate the lyrics into phonemes. For English [1] and Italian [2], we utilize tools to translate them into International Phonetic Alphabet (IPA) phonemes [53]. For Mandarin, we use Pinyin [3] for phoneme-level alignment and provide a mapping of Pinyin to IPA phonemes for later phoneme-set processing for model training. We did not directly convert Chinese to IPA due to annotation complexity. Thirdly, we obtain an approximate phoneme alignment with audio using the Montreal Forced Aligner [54] and output it into TextGrid files. Finally, we (1) use *Praat* software [55], or (2) convert the TextGrid into txt files and input them to *Audacity* [56] for further manual adjustment of phoneme text and boundaries, as well as breath and silence event annotation (Figure 4).

Performance MIDI and Score annotation We annotated the performance MIDI file and music score for each singing phrase in the dataset as follows:

(1) We manually input performance MIDI files that strictly align to singing audio using MIDI piano, including multiple rounds of correction.
(2) We automatically align MIDI notes with phonemes based on their corresponding time stamps in the audio.
(3) We search online for music score MIDI files; if no reliable sources are found, we quantize and derive the score from annotated performance MIDI file.
(4) For online score files, we develop an algorithm that automatically matches each singing phrase's performance MIDI data to the corresponding phrase in the score MIDI file. Manual matching is required for non-original-style style transfer versions.
(5) We use the Dynamic Time Warping algorithm to match the performance MIDI data with the score MIDI file within each phrase. We manually verify the mapping results for non-original-style style transfer versions.

All these above steps allow us to annotate the lyrics, performance MIDI, and music score at the phoneme level for our singing voice dataset, ensuring accurate and comprehensive representations of the musical content.

4.3 Acoustic Feature Extraction

F0, or fundamental frequency, is the lowest frequency of a periodic waveform. F0 contour is critical in singing synthesis as it determines the pitch variations of singing performance and largely influences singing quality. It can capture pitch modulations in various singing techniques, such as vibrato, ornaments, and glissando. Many current SVS systems still require the input of ground-truth F0 as a condition to guide the synthesis process. To ensure accurate F0 extraction, we employ a combination of two widely-used models, pYIN [57] and PENN [58]. First, we use pYIN algorithm to identify unvoiced parts, including breaths, silence, and consonants. Then, the PENN algorithm is applied to extract F0 for the voiced parts.

Loudness represents the energy of a sound. It is crucial in singing performance since it largely reflects the dynamic and emotional changes that contribute to the expressiveness of the singing voice. To extract loudness, we first calculate the root-mean-square (RMS) amplitude values from audio and then convert them to decibels. We further ap-

[1] https://github.com/mphilli/English-to-IPA
[2] https://espeak.sourceforge.net/
[3] https://github.com/mozillazg/python-pinyin

ply a moving average window of frame size 30 to obtain a smoother loudness curve.

Finally, we use the Short-Time Fourier Transform (STFT) with a window size of 1024, FFT size of 1024, and hop size of 256 to extract the mel-spectrogram, which shares the same settings with loudness extraction.

5. POTENTIAL APPLICATIONS

This dataset is intended to promote research into a number of different MIR tasks. We consider a variety of interesting relevant problems in this section.

5.1 Singing Style Transfer

Style transfer has to do with music interpretation. Here, "style" refers to performance details that are not constrained by symbolic representations such as traditional notation. If notation gives a song its "identity," styles are performance characteristics that are shared across performances of different songs. Styles are often associated with genre, e.g., a song can be interpreted in rock, pop, or jazz styles. Styles can be more or less specific than genre, e.g., the style of Louis Armstrong (more specific) or symphonic (less specific). Style transfer is a process of identifying the style of one or more performances and applying it to a new song to create a stylistic performance. SingStyle111 contains many performances where a single singer performs in multiple styles, offering the potential to abstract styles from other information (singer identity, melodies) which is held constant. In the multi-style recordings, singers were asked to exaggerate differences, which should help to learn features that characterize different styles.

5.2 Singing Voice Synthesis

A large motivation for SingStyle111 is the difficulty of finding high-quality musical examples of singing. In particular, the presence of accompaniment and reverberation complicate the process of learning to create the sound of singing voices. Furthermore, lower recording and singing quality are a barrier to learning high-fidelity sounds of professional singing. In addition, SingStyle111 also provides necessary phoneme-level annotations for score-based SVS.

5.3 Singing Voice Conversion

In SVC, we hope to substitute the sound of one voice with the sound of another while maintaining the same melody and style. To promote progress in this area, SingStyle111 has performances of the same song by multiple singers, including male and female voices. Since we have performances of the same song in the same style, SVC can be cast as a sequence-to-sequence problem analogous to many other machine learning tasks such as language translation.

5.4 Expressive Performance

Expressive performance is the general problem of creating a musical performance given a symbolic description such as a melody in common music notation. Notation omits many details, including loudness, vibrato, pitch variations, changes in vocal timbre, the details of pronouncing lyrics within pitch and rhythmic constraints, and breathiness. Often, connections and transitions from one note to the next are as important as how notes are performed. To learn expressive performance, it helps to have symbolic notation, which can be considered as input constraints, context, or conditioning. In addition, it helps if the notated events are aligned with corresponding time points in the audio. SingStyle111 includes symbolic representations (performance MIDI files and music scores) aligned with audio. The data is especially designed to support machine learning using sequence-to-sequence models from notation to control signals such as pitch contours, loudness, spectrograms, or directly to audio.

5.5 Automatic Singing Transcription

Singing transcription can be regarded as the inverse of expressive performance control: Rather than converting notation to sound, we wish to convert sound into music notation. With transcriptions for all of the singing examples, SingStyle111 provides a wealth of transcription examples for training and evaluating transcription models.

5.6 Score Alignment and Following

Score following [59] is the problem of aligning an audio performance to symbolic notation. Vocal score following is particularly difficult because, unlike most other instruments, voices do not have keys, valves, or frets, so singing cannot be easily reduced to a sequence of distinct discrete states corresponding to musical notes [60]. Real-time score following is the first step in the task of computer accompaniment, in which a computer synchronizes a pre-composed accompaniment to a live performance by a soloist. Score following has also been used for automatic page turning, delivering synchronized comments via mobile phones to symphony orchestra audiences, and as a data collection method for learning music segmentation and other tasks. SingStyle111 contains accurate alignments for learning and evaluation of automatic alignment and real-time score following.

5.7 Lyrics Detection

The common task of understanding lyrics is one that even humans struggle with. SingStyle111 includes the lyrics used by the singers, and lyrics are aligned to the audio down to the phoneme level, facilitating learning and evaluation of various lyrics transcription and alignment tasks.

6. CONCLUSION

In conclusion, we introduce SingStyle111, a large-scale, high-quality, multilingual singing voice dataset that caters to various styles and includes style transfer examples. We provided detailed annotations of lyrics and scores at the phoneme level, together with extracted acoustic features. We will make the dataset freely available for research purposes to facilitate relevant MIR tasks.

7. REFERENCES

[1] Y. Cao, S. Li, Y. Liu, Z. Yan, Y. Dai, P. S. Yu, and L. Sun, "A comprehensive survey of ai-generated content (aigc): A history of generative ai from gan to chatgpt," *arXiv preprint arXiv:2303.04226*, 2023.

[2] 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 *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.

[3] 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.

[4] A. Q. Nichol, P. Dhariwal, A. Ramesh, P. Shyam, P. Mishkin, B. Mcgrew, I. Sutskever, and M. Chen, "Glide: Towards photorealistic image generation and editing with text-guided diffusion models," in *International Conference on Machine Learning*. PMLR, 2022, pp. 16 784–16 804.

[5] 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.

[6] 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.

[7] OpenAI, "Gpt-4 technical report," 2023.

[8] 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.

[9] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan *et al.*, "Natural tts synthesis by conditioning wavenet on mel-spectrogram predictions," in *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2018, pp. 4779–4783.

[10] Y. Ren, C. Hu, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "Fastspeech 2: Fast and high-quality end-to-end text to speech," in *International Conference on Learning Representations*.

[11] Z. Kong, W. Ping, J. Huang, K. Zhao, and B. Catanzaro, "Diffwave: A versatile diffusion model for audio synthesis," in *International Conference on Learning Representations*.

[12] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," in *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[13] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *International conference on machine learning*. PMLR, 2014, pp. 1278–1286.

[14] A. Van Den Oord, O. Vinyals *et al.*, "Neural discrete representation learning," *Advances in neural information processing systems*, vol. 30, 2017.

[15] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[16] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional gans," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8798–8807.

[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] 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.

[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] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," in *International Conference on Learning Representations*.

[21] J. Kong, J. Kim, and J. Bae, "Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis," *Advances in Neural Information Processing Systems*, vol. 33, pp. 17 022–17 033, 2020.

[22] K. Akuzawa, Y. Iwasawa, and Y. Matsuo, "Expressive speech synthesis via modeling expressions with variational autoencoder," *Proc. Interspeech 2018*, pp. 3067–3071, 2018.

[23] V. Aggarwal, M. Cotescu, N. Prateek, J. Lorenzo-Trueba, and R. Barra-Chicote, "Using vaes and normalizing flows for one-shot text-to-speech synthesis of expressive speech," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6179–6183.

[24] 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.

[25] M. Bińkowski, J. Donahue, S. Dieleman, A. Clark, E. Elsen, N. Casagrande, L. C. Cobo, and K. Simonyan, "High fidelity speech synthesis with adversarial networks," in *International Conference on Learning Representations*.

[26] P. Lu, J. Wu, J. Luan, X. Tan, and L. Zhou, "Xiaoicesing: A high-quality and integrated singing voice synthesis system," *Proc. Interspeech 2020*, pp. 1306–1310, 2020.

[27] Y. Gu, X. Yin, Y. Rao, Y. Wan, B. Tang, Y. Zhang, J. Chen, Y. Wang, and Z. Ma, "Bytesing: A chinese singing voice synthesis system using duration allocated encoder-decoder acoustic models and wavernn vocoders," in *2021 12th International Symposium on Chinese Spoken Language Processing (ISCSLP)*. IEEE, 2021, pp. 1–5.

[28] J. Liu, C. Li, Y. Ren, F. Chen, and Z. Zhao, "Diffsinger: Singing voice synthesis via shallow diffusion mechanism," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 10, 2022, pp. 11 020–11 028.

[29] Z. Li, B. Tang, X. Yin, Y. Wan, L. Xu, C. Shen, and Z. Ma, "Ppg-based singing voice conversion with adversarial representation learning," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 7073–7077.

[30] A. Jansson, E. Humphrey, N. Montecchio, R. Bittner, A. Kumar, and T. Weyde, "Singing voice separation with deep u-net convolutional networks," 2017.

[31] D. Stoller, S. Ewert, and S. Dixon, "Adversarial semi-supervised audio source separation applied to singing voice extraction," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 2391–2395.

[32] Y.-J. Lu, Z.-Q. Wang, S. Watanabe, A. Richard, C. Yu, and Y. Tsao, "Conditional diffusion probabilistic model for speech enhancement," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 7402–7406.

[33] J. Su, Z. Jin, and A. Finkelstein, "Hifi-gan: High-fidelity denoising and dereverberation based on speech deep features in adversarial networks," *Proc. Interspeech 2020*, pp. 4506–4510, 2020.

[34] 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.

[35] R. Huang, F. Chen, Y. Ren, J. Liu, C. Cui, and Z. Zhao, "Multi-singer: Fast multi-singer singing voice vocoder with a large-scale corpus," in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 3945–3954.

[36] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232.

[37] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2414–2423.

[38] K. Qian, Y. Zhang, S. Chang, X. Yang, and M. Hasegawa-Johnson, "Autovc: Zero-shot voice style transfer with only autoencoder loss," in *International Conference on Machine Learning*. PMLR, 2019, pp. 5210–5219.

[39] K. Qian, Z. Jin, M. Hasegawa-Johnson, and G. J. Mysore, "F0-consistent many-to-many non-parallel voice conversion via conditional autoencoder," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6284–6288.

[40] S. Dai, Z. Zhang, and G. G. Xia, "Music style transfer: A position paper," *arXiv preprint arXiv:1803.06841*, 2018.

[41] 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," *Interspeech 2022*, 2022.

[42] 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," *Advances in Neural Information Processing Systems*, vol. 35, pp. 6914–6926, 2022.

[43] 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.

[44] I. Ogawa and M. Morise, "Tohoku kiritan singing database: A singing database for statistical parametric singing synthesis using japanese pop songs," *Acoustical Science and Technology*, vol. 42, no. 3, pp. 140–145, 2021.

[45] J. Wilkins, P. Seetharaman, A. Wahl, and B. Pardo, "Vocalset: A singing voice dataset." in *ISMIR*, 2018, pp. 468–474.

[46] B. Faghih and J. Timoney, "Annotated-vocalset: A singing voice dataset," *Applied Sciences*, vol. 12, no. 18, p. 9257, 2022.

[47] B. Sharma, X. Gao, K. Vijayan, X. Tian, and H. Li, "Nhss: A speech and singing parallel database," *Speech Communication*, vol. 133, pp. 9–22, 2021.

[48] Z. Duan, H. Fang, B. Li, K. C. Sim, and Y. Wang, "The nus sung and spoken lyrics corpus: A quantitative comparison of singing and speech," in *2013 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*. IEEE, 2013, pp. 1–9.

[49] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "Rwc music database: Popular, classical and jazz music databases." in *Ismir*, vol. 2, 2002, pp. 287–288.

[50] J. Mora, F. Gomez Martin, E. Gómez, F. J. Escobar-Borrego, and J. M. Díaz-Báñez, "Characterization and melodic similarity of a cappella flamenco cantes." International Society for Music Information Retrieval Conference, ISMIR, 2010.

[51] R. M. Bittner, K. Pasalo, J. J. Bosch, G. Meseguer-Brocal, and D. Rubinstein, "vocadito: A dataset of solo vocals with f_0, note, and lyric annotations," 2021.

[52] 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, 2009.

[53] I. P. Association, *Handbook of the International Phonetic Association: A guide to the use of the International Phonetic Alphabet*. Cambridge University Press, 1999.

[54] M. McAuliffe, M. Socolof, S. Mihuc, M. Wagner, and M. Sonderegger, "Montreal Forced Aligner: Trainable Text-Speech Alignment Using Kaldi," in *Proc. Interspeech 2017*, 2017, pp. 498–502.

[55] P. Boersma, "Praat, a system for doing phonetics by computer," *Glot International 5:9/10, 341-345*, 2001.

[56] D. Mazzoni and R. Dannenberg, "Audacity [software]," *The Audacity Team, Pittsburg, PA, USA*, vol. 328, 2000.

[57] 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)*. IEEE, 2014, pp. 659–663.

[58] M. Morrison, C. Hsieh, N. Pruyne, and B. Pardo, "Cross-domain neural pitch and periodicity estimation," in *Submitted to IEEE Transactions on Audio, Speech, and Language Processing*, TODO 2023.

[59] R. B. Dannenberg and C. Raphael, "Music score alignment and computer accompaniment," *Communications of the ACM*, vol. 49, no. 8, pp. 38–43, August 2006.

[60] L. Grubb, "A probabilistic method for tracking a vocalist," PhD thesis, Carnegie Mellon University, 1998.

A COMPUTATIONAL EVALUATION FRAMEWORK FOR SINGABLE LYRIC TRANSLATION

Haven Kim [1] **Kento Watanabe** [2] **Masataka Goto** [2] **Juhan Nam** [1]
[1] Graduate School of Culture Technology, KAIST, South Korea
[2] National Institute of Advanced Industrial Science and Technology (AIST), Japan
khaven@kaist.ac.kr, kento.watanabe@aist.go.jp, m.goto@aist.go.jp, juhan.nam@kaist.ac.kr

ABSTRACT

Lyric translation plays a pivotal role in amplifying the global resonance of music, bridging cultural divides, and fostering universal connections. Translating lyrics, unlike conventional translation tasks, requires a delicate balance between singability and semantics. In this paper, we present a computational framework for the quantitative evaluation of singable lyric translation, which seamlessly integrates musical, linguistic, and cultural dimensions of lyrics. Our comprehensive framework consists of four metrics that measure syllable count distance, phoneme repetition similarity, musical structure distance, and semantic similarity. To substantiate the efficacy of our framework, we collected a singable lyrics dataset, which precisely aligns English, Japanese, and Korean lyrics on a line-by-line and section-by-section basis, and conducted a comparative analysis between singable and non-singable lyrics. Our multidisciplinary approach provides insights into the key components that underlie the art of lyric translation and establishes a solid groundwork for the future of computational lyric translation assessment.

1. INTRODUCTION

Translating lyrics is a prevailing method of enhancing the global appeal and allure of music across a multitude of genres, such as theatre music, animation music, pop music, etc [1]. Furthermore, recent advancements in media technology have facilitated the exchange of intercultural products and globalized fandom culture, resulting in an increase in the popularity of user-translated lyrics across diverse social media platforms [2].

Despite its popularity, lyric translation is acknowledged as a challenging field, requiring an interdisciplinary approach [2]. As early as 1915, it was suggested that an ideal lyric translator should possess expertise in both linguistics and music, highlighting the need for a comprehensive understanding of the principles and techniques used in translation studies, coupled with a background in musicology [3]. Moreover, it is crucial to understand the cultural context of each language, such as different strategies employed for forming rhymes [4]. Because of these challenges, the systematic analysis and evaluation of lyric translation remain an under-researched topic of study. Thus far, only a few have proposed guidelines for scoring the quality of translated lyrics [4, 5]. While these principled approaches have proven successful, they lack automation. Consequently, despite the growing interest in the development of neural lyric translation, its evaluation has predominantly relied on human evaluation, making the evaluation process time-consuming, unreliable, and subjective [6–8].

Our study aims to computationally analyze and evaluate lyric translation based on a comprehensive understanding of lyrics that accounts for their musical, linguistic, and cultural elements. Unlike prior research that only proposed rhyme-scoring guidelines applicable to English [4], our framework is extendable to Japanese and Korean. Though our framework may be limited in its application to specific languages, we strive to provide valuable insights into establishing distinct evaluation rules for phoneme repetition in diverse languages. Our comprehensive framework employs a multi-faceted evaluation approach that examines lyric translation from four distinct perspectives: syllable counts, phoneme repetition, musical structure, and semantics. In the remainder of this paper, we explicate the rationale behind our selection of these perspectives by delving into the unique characteristics of lyric translation that differentiate it from general language translation tasks. In addition, we introduce the singable lyrics dataset we collected, which features line-by-line and section-by-section alignment of English, Japanese, and Korean lyrics. Moving forward, we propose robust evaluation metrics for lyric translation and analyze the results of our experiments based on the perspectives mentioned above. Finally, we conclude our paper by reflecting on the profound insights gleaned from our experiment and highlighting possible directions for future research.

2. BACKGROUND

Previous research indicates that linguistic analysis methods designed for standard text may not achieve desired outcomes when used to examine lyrics [9]. Although automated evaluation metrics, such as n-gram-based [10–12] or neural approaches [13], have proven valuable and effective in assessing conventional machine translation tasks, they

© H. Kim, K. Watanabe, M. Goto, and J. Nam. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** H. Kim, K. Watanabe, M. Goto, and J. Nam, "A Computational Evaluation Framework for Singable Lyric Translation", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

fall short in evaluating lyric translation. This is due to the unique characteristics of lyrics that render the translation process subject to many constraints and less direct [14].

One of the most significant constraints is the syllable count. This is because the original and translated lyrics must match the same melody lines, while it is a common practice to tweak the melody to accommodate minor changes in syllable count [4, 15]. In fact, conveying the same message in different languages requires vastly different syllable counts. For example, "Happy New Year" in English consists of 4 syllables, whereas 15 and 9 are required for Japanese (あけましておめでとうございます) and Korean (새해 복 많이 받으세요), respectively. For the numerical comparison, we examined PAWS-X, a dataset that contains 23,659 English sentences paired with human-translated sentences in various languages [16]. The average number of syllables per sentence in the dataset is 50.89 for Japanese, whereas 36.18 per English and 40.40 per Korean. With these statistics, it can be deduced that Japanese necessitates approximately 41% more syllables than English and 26% more syllables than Korean to express an equivalent message. This limitation forces translators to often modify the meaning of original lyrics by adding, omitting, or even tweaking the message. However, translated lyrics still aim to capture the theme, mood, and spirit of the original lyrics [17]. Therefore, while original and translated lyrics need not be semantically identical, they still need to be semantically relevant [4, 18].

It is also crucial to preserve the frequency of phoneme repetition (e.g., rhyme) in translated lyrics, particularly when the music demands it [17]. For instance, some sections, such as choruses, require a substantial degree of phoneme repetition, while others do not. Moreover, due to the inherent connection between lyrics and music, lyrics must be arranged in a way that complements the music [19]. As a result, musically similar sections should maintain resembling linguistic features, including the choice of phonemes and the frequency of phoneme repetition [20].

3. DATASET

Although some websites provide user-translated multilingual lyrics, we found that most of them lack singability, as these translations were focused on delivering the meaning of the original lyrics rather than making them performable. While there are a few singable translations available, they are often not aligned on a line-by-line nor section-by-section basis due to the subjective nature of the lyric structure that there is no universal agreement on what to call a line and what to call a section. The absence of alignment makes it difficult to compare the original lyrics with their translated versions. To address this issue, we collected a set of singable lyrics, sourced from either official lyrics of commercial songs or user-translated ones found on YouTube, meticulously aligned on a line-by-line basis in English, Japanese, and Korean. This approach ensures that lyrics on the same line share the same melodies. Moreover, the dataset divides the lyrics into sections, allowing for section-by-section analysis. Alongside the lyrics, it provides essential metadata such as genre, artist, original language, and the official status of lyrics. The dataset consists of 162 songs, each having lyrics in the three languages. It covers a diverse range of genres, including 109 K-pop, 23 animation music (e.g., Disney), 13 J-pop, 10 theatre music, and more. Table 1 shows sample data.

Section #	Line #	English (EN)	Japanese (JP)	Korean (KR)
1	1	Twinkle, twinkle, little star	きらきらひかる	반짝 반짝 작은별
	2	How I wonder what you are	おそらのほしよ	아름답게 비치네
	3	Up above the world so high	まばたきしては	서쪽 하늘에서도
	4	Like a diamond in the sky	みんなをみてる	동쪽 하늘에서도
	5	Twinkle, twinkle, little star	きらきらひかる	반짝 반짝 작은별
	6	How I wonder what you are	おそらのほしよ	아름답게 비치네
2	7	Twinkle, twinkle, little star	きらきらひかる	반짝 반짝 작은별
	8	How I wonder what you are	おそらのほしよ	아름답게 비치네
	9	When the blazing sun is gone	みんなのうたが	서쪽 하늘에서도
	10	When he nothing shines upon	とどくといいな	동쪽 하늘에서도
	11	Then you show your little light	きらきらひかる	반짝 반짝 작은별
	12	Twinkle, twinkle, all the night	おそらのほしよ	아름답게 비치네

Table 1. Sample data illustrating the original English lyrics of "Twinkle, Twinkle, Little Star" and their corresponding singable translations in Japanese and Korean, aligned on a line-by-line and section-by-section basis.

4. EVALUATING SINGABILITY

Our primary goal is to develop an evaluation framework that automatically assesses the quality of translated lyrics. One of the most important factors determining the quality is *singability*, defined as not only the ability of being sung, but also the suitability (easiness) of being sung [18]. To ensure such singability, we aim to provide metrics from three distinct perspectives by making sure that they i) maintain the song's melodic integrity, ii) preserve the degree of phoneme repetition, and iii) consider the underlying musical structure.

To substantiate the reliability of our evaluation metrics, we conducted a comparative analysis of singable lyrics versus non-singable lyrics based on each proposed evaluation metric. In all our comparative analyses, we utilized our dataset for singable lyrics, where official lyrics served as both source and target lyrics, and unofficial functioned as only target lyrics. For non-singable lyrics, we obtained pairs of original singable (source) and human-translated non-singable (target) lyrics, aligned line-wise and section-wise, for 3,642 songs from https://lyricstranslate.com/.

Source	Target	Singable	Non-singable
English	Japanese	0.17 (80 songs)	0.74 (1401 songs)
	Korean	0.11 (80 songs)	0.48 (620 songs)
Japanese	English	0.16 (162 songs)	0.39 (589 songs)
	Korean	0.11 (162 songs)	0.31 (73 songs)
Korean	English	0.09 (161 songs)	0.20 (702 songs)
	Japanese	0.12 (161 songs)	0.52 (257 songs)

Table 2. The average line syllable count distance (Dis_{syl}) between source and target languages for singable and non-singable lyrics.

Section	English	Japanese (English translation)	Korean (English translation)	*pho*
$E(A_1)$, $J(A_1)$, $K(A_1)$	Do you wanna build a snowman? Come on, let's go and play! I never see you anymore Come out the door It's like you've gone away	雪だるま作ろう (Let's build a snowman) ドアを開けて (Please open the door) 一緒に遊ぼう (Let's play together) どうして (Why) 出てこないの? (don't you come out?)	같이 눈사람 만들래? (Do you wanna build a snowman?) 제발 좀 나와봐 (Please come out) 언니를 만날 수 없어 (I can't meet you) 같이 놀자 (Let's play together) 나 혼자 심심해 (I'm lonely alone)	0.85, 0.73, 0.77
$E(B_1)$, $J(B_1)$, $K(B_1)$	We used to be best buddies And now we're not I wish you would tell me why!	前は仲良く(We were close before) してたのに (We used to be) なぜ会えないの (Why can't we meet each other?)	그렇게 친했는데 (We were close before) 이젠 아냐 (and we're not) 그 이유를 알고파 (I want to know the reason why)	0.92, 0.80, 0.91
$E(A_2)$, $J(A_2)$, $K(A_2)$	Do you wanna build a snowman? Or ride our bike around the halls? I think some company is overdue I've started talking to the pictures on the walls!	雪だるま作ろう (Let's build a snowman) 自転車に乗ろう(Let's ride a bike) ずっと一人でいると (When I'm alone all the time) 壁の絵とおしゃべりしちゃう (I'm almost talking to the pictures on the walls)	같이 눈사람 만들래? (Do you wanna build a snowman?) 아니면 자전거 탈래? (or do you wanna ride a bike?) 이제는 나도 지쳐 가나봐 (Seems I'm getting tired) 벽에다 말을 하며 놀고 있잖아 (because I've started talking to the walls)	0.79, 0.73, 0.82
$E(B_2)$, $J(B_2)$, $K(B_2)$	It gets a little lonely All these empty rooms Just watching the hours tick by	さびしい部屋で (In a lonely room) 柱時計 (the wall clock) 見てたりするの (I look at or something)	사실은 조금 외로워 (In fact, I'm a little lonely) 텅빈 방에선 (In empty rooms) 시계소리만 들려 (All I can hear is the clock's ticking)	0.90, 0.88, 0.96

Table 3. Lyric excerpt from "Do You Want to Build a Snowman" from the animation "Frozen," singable in all languages. Sections A_1 and A_2 form a musically similar pair, while B_1 and B_2 are also musically similar to each other. Each section is denoted as $E(A_1), \ldots, E(B_2)$ in English, $J(A_1), \ldots, J(B_2)$ in Japanese, and $K(A_1), \ldots, K(B_2)$ in Korean.

4.1 Line Syllable Count Distance

It is crucial to preserve the syllable counts between the original and translated lyrics for each line as similar as possible in order to maintain the integrity of a song's melody [21]. Therefore, it is unsurprising that our evaluation framework incorporates a metric to assess the differences in syllable counts. Let the line syllable counts for a pair of lyrics that consist of n lines, $\mathbf{X} = \{x_1, ..., x_n\}$ and $\tilde{\mathbf{X}} = \{\tilde{x_1}, ..., \tilde{x_n}\}$ be denoted as $\{syl(x_1), ..., syl(x_n)\}$ and $\{syl(\tilde{x_1}), ..., syl(\tilde{x_n})\}$ where each element refers to the syllable count of each line. For instance, if the first line of the English lyrics $\mathbf{X}$ is "Silent night holy night" and the corresponding line in the Korean lyrics $\tilde{\mathbf{X}}$ is "Goyohanbam-georukhanbam (고요한밤 거룩한밤)", the value of $syl(x_1)$ is 6 and $syl(\tilde{x_1})$ is 8. We define the **line syllable count distance** between a pair of lyrics $\mathbf{X}$ and $\tilde{\mathbf{X}}$ ($Dis_{syl}(\mathbf{X}, \tilde{\mathbf{X}})$) in order to evaluate the disparities in syllable counts, as follows.

$$Dis_{syl}(\mathbf{X}, \tilde{\mathbf{X}}) = \frac{1}{2n} \sum_{i=1}^{n} \left(\frac{|syl(x_i) - syl(\tilde{x_i})|}{syl(x_i)} + \frac{|syl(x_i) - syl(\tilde{x_i})|}{syl(\tilde{x_i})} \right) \quad (1)$$

We compare the line syllable count distance of singable and non-singable lyrics. As shown in Table 2, non-singable lyrics display a considerably greater $Dis_{syl}(\mathbf{X}, \tilde{\mathbf{X}})$ compared to singable lyrics due to the varying syllable count requirements across languages.

4.2 Phoneme Repetition Similarity

Rhyme, defined as the repetition of a vowel sound and any subsequent sounds [22], has historically been a fundamental element in the realm of poetry, including in Western languages like English. However, the concept of rhyme has not been as prevalent in Japanese or Korean poetry [23]. In fact, traditional Korean poetry did not incorporate this concept [24]. Despite the increasing tendency to adopt the concept of rhyme in Japanese and Korean lyrics due to intercultural exchanges, we observed that lyrics in these languages often rely more on repeating grammatical elements. For example, in section A_1 of Table 3, the Japanese pair "tsukurou (作ろう, Let's build)" and "asobou (遊ぼう, Let's play)" generates a sense of rhyme because both end with the same conjugation "ou" meaning "let's". Similarly, in Section A_2, the Korean pair "mandeullae (만들래, Do you wanna build)" and "tallae (탈래, Do you wanna ride)" creates a sense of repetition because both end with "llae" meaning "Do you wanna". Another example is the repetition of particles at the end of sentences, such as "yo (よ)" and "no (の)" in Japanese and "yo (요)" and "da (다)" in Korean, which convey cultural nuances related to formality. We therefore propose that English, Japanese, and Korean share common ground in adopting phoneme repetition for poetic expression. However, as such repetition is not necessarily called rhyme in Japanese and Korean, we will refrain from using the term "rhyme" and instead employ the term "phoneme repetition."

We noticed that each section's degree of phoneme repetition remains consistent across different languages when the lyrics are singable. For example, in Table 3, the first section of the original English lyrics ($E(A_1)$) displays a strong degree of phoneme repetition, with three rhyming pairs: "come-come", "play-away", and "anymore-door" (In this paper, we denote a section as an uppercase with a number and a line as a lower case with a number). Similarly, both the Japanese and Korean translations ($J(A_1)$, $K(A_1)$) also exhibit a substantial degree of phoneme repetition, featuring three pairs of repeated phonemes in each: "doa (ドア)"-"dou (どう)", "tsukurou (作ろう)"-"asobou (遊ぼう)", "akete (開けて)"-"shite (して)" in Japanese, and "gachi (같이)"-"gachi (같이)", "mandeul (만들)"-"eonnireul (언니를)", "mandeullae (만들래)"-"simsimhae (심심해)" in Korean. However, we realized that it is not fair to directly compare the number of phoneme repetitions when attempting to quantify the degree of phoneme repetition as each language has a different number of vowels and consonants: English has 15 vowels and 24 consonants, whereas Japanese has 5 and 15 and Korean has 21 and 19. Hence, in an attempt to minimize the differences in the number of phonemes, we treated acoustically similar vowels as the same vowel in English, such as 'IH'-'IY', 'UH'-'UW', or 'EH'-'AE'(e.g., 'mass' and 'mess') [25] because they can still form slant rhymes [26]. Conversely, we considered 'A'-'YA', 'O'-'YO', and 'U'-'YU' as sep-

Source	Target	Singable	Non-singable
English	Japanese	0.69	0.56
	Korean	0.97	0.72
Japanese	English	0.79	0.61
	Korean	0.80	0.48
Korean	English	0.97	0.78
	Japanese	0.80	0.50

Table 4. The average phoneme repetition similarity (Sim_{pho}) of singable lyrics and non-singable lyrics.

arate vowels in Japanese, as they are unlikely to function as the same grammatical components. In Korean, we regarded the perceptually similar vowels (e.g., 'AE'-'E'or 'OE'-'OI'-'OAE') as the same vowels [27, 28].

To quantitatively represent the degree of phoneme repetition, we utilized the concept of *distinct-2*, the ratio of the number of distinct bi-grams to the total number of bi-grams [29]. While the original concept formed bi-grams using two consecutive words, we used two consecutive phonemes to assess the degree of repetition because lower *distinct-2* values indicate higher repetition and vice versa. The *phoneme distinct-2* (pho) of a section X_i, is defined as follows:

$$pho(X_i) = \frac{\textit{unique bi-gram \# in } X_i}{\textit{total bi-gram \# in } X_i}. \quad (2)$$

For example, consider a section with a single line "twinkle twinkle little star", denoted as X_1. First, we decomposed the section into phonemes and added the '<eos>'to each line: 'T', 'W', 'IH', 'NG', 'K', 'AH', ..., 'S', 'T', 'AA', 'R', and '<eos>'. Next, we grouped each component into bi-grams: 'TW', 'WIH', 'IHNG', 'NGK', 'KAH', 'AHL', ..., 'ST', 'TAA', 'AAR', 'R<eos>'. Finally, we calculated the *phoneme distinct-2* of the section ($pho(X_1)$) by dividing the number of unique bi-grams by the total number of bi-grams (17/23 = 0.74). To measure the similarity between two sections in terms of the degree of phoneme repetition, we introduce the **phoneme repetition similarity** (Sim_{pho}). Given two sets of lyrics with m sections, $\mathbf{X} = \{X_1, ..., X_m\}$ and $\tilde{\mathbf{X}} = \{\tilde{X_1}, ..., \tilde{X_m}\}$, the phoneme repetition similarity between $\mathbf{X}$ and $\tilde{\mathbf{X}}$ is defined as the Spearman correlation between $\{pho(X_1), ..., pho(X_m)\}$ and $\{pho(\tilde{X_1}), ..., pho(\tilde{X_m})\}$, as shown below.

$$Sim_{pho}(\mathbf{X}, \tilde{\mathbf{X}}) = corr(\{pho(X_1), ..., pho(X_m)\}, \{pho(\tilde{X_1}), ..., pho(\tilde{X_m})\}) \quad (3)$$

We present the statistical results for the average phoneme repetition similarity of singable and non-singable lyrics in Table 4. The table clearly exhibits a higher correlation between the original lyrics and singable translated lyrics in terms of the *phoneme distinct-2* than non-singable lyrics. This result suggests that singable lyric translation takes into account the degree of phoneme repetition within each section to convey a sense of repetition for that section.

4.3 Musical Structure Distance

Upon examining our section-divided singable lyrics data, we identified two tendencies in lyrics when musical sections

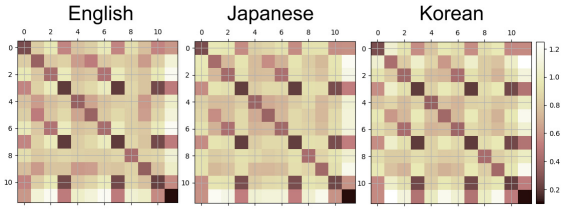


Figure 1. Musical self-dissimilarity matrices for English, Japanese, and Korean versions of the K-pop song "Icy" by ITZY. Dissimilarity between the i-th and the j-th section was computed using $diss(X_i, X_j)$.

are repeated (e.g., the repetition of the chorus). First, we observed that musically similar sections tend to share the same phonemes and, as expected, the same phrases. For instance, in Table 3, musically similar sections share the same vowels (e.g., "why" in $E(B_1)$ and "by" in $E(B_2)$) or identical phrases (e.g., "Do you wanna build a snowman" in $E(A_1)$ and $E(A_2)$) in order to create a sense of consistency. As a result, when calculating the *phoneme distinct-2* (pho) for two concatenated sections, musically similar sections are likely to have smaller values than musically different sections. For example, $pho(E(A_1 +\!\!+ A_2))$ is 0.70 ('$+\!\!+$' denotes the concatenation of text), whereas $pho(E(A_1 +\!\!+ B_1))$ is 0.82, where A_1 is musically similar to A_2 but not to B_1. However, a low value of pho does not always imply musical similarity, as a meager pho value in one section could result in a low pho of two concatenated sections despite the musical dissimilarity (e.g., "lalalalalalalalalalalalalala" $+\!\!+$ "do you wanna build a snowman?"). From this case, we derived our second observation that a significant difference in pho for each section could imply musical differences. Accordingly, we also realized that musically similar sections tend to have a similar degree of pho. For instance, in Table 3, both A_1 and A_2, a set of musically similar sections, exhibit relatively low pho, indicating a strong degree of phoneme repetition (rhyme), with similar values to each other across all languages. Likewise, both B_1 and B_2, another pair of musically similar sections, demonstrate a higher pho, with similar values to each, in all languages.

Therefore, to quantify the musical similarity between sections, we examined whether they have 1) a tendency to share the same phoneme by obtaining $pho(X_i +\!\!+ X_j)$, and 2) similar pho values by calculating $|pho(X_i) - pho(X_j)|$. Given that higher values represent dissimilarity in both cases, we define the musical dissimilarity between two sections, $diss(X_i, X_j)$, as the sum of these two values, as follows.

$$diss(X_i, X_j) = pho(X_i +\!\!+ X_j) + |pho(X_i) - pho(X_j)| \quad (4)$$

As shown in Figure 1, self-dissimilarity matrices employing our definition of musical dissimilarity look highly similar across English, Japanese, and Korean, where all are singable, visually representing musical structure.

Finally, we quantitatively evaluated the distance between matrices. We refer to this distance between matrices as the musical structure distance, as it represents the structural element of lyrics. In summary, the **musical structure distance**

Source	Target	Singable	Non-singable (Human)	Non-singable (Machine)
English	Japanese	0.14	0.26	0.30
	Korean	0.10	0.15	0.15
Japanese	English	0.13	0.20	0.18
	Korean	0.11	0.13	0.14
Korean	English	0.10	0.10	0.10
	Japanese	0.11	0.10	0.17

Table 5. The average musical structure distance (Dis_{mus}) of singable lyrics and non-singable lyrics.

between lyrics in different languages $\mathbf{X}$ and $\tilde{\mathbf{X}}$, each consisting of m sections, $Dis_{mus}(\mathbf{X}, \tilde{\mathbf{X}})$, is defined as follows:

$$Dis_{mus}(\mathbf{X}, \tilde{\mathbf{X}}) = \frac{1}{m^2}\sqrt{\sum_{i,j=1}^{m} (diss(X_i, X_j) - diss(\tilde{X}_i, \tilde{X}_j))^2} \quad (5)$$

In Table 5, we provide a summary of the average musical structure distance for singable lyrics, human-translated non-singable lyrics, and machine-translated non-singable lyrics generated by automatically translating official singable lyrics from 80 English, 162 Japanese, and 161 Korean songs using Google Translator. Our findings show that singable lyrics exhibit the lowest Dis_{mus} values, while machine-translated non-singable lyrics display the highest, suggesting that machine-translated ones lack structural coherence the most. As human-translated non-singable lyrics maintain structural coherence in aspects such as word choice and nuances, they demonstrate lower distances than machine-translated counterparts.

5. EVALUATING SEMANTICS

Semantic relatedness to the original lyrics is by no means less fundamental than syllable counts, phoneme repetition, and structural factors [18]. We therefore introduce a fourth metric, semantic similarity, to ensure the semantic relevance of translated lyrics to the original.

5.1 Semantic Similarity

To numerically assess the semantic textual similarity (sts) between a pair of lyrics, we first obtained the contextual embeddings of each text from lyrics using a pre-trained Sentence BERT model [1] [31] and then calculated the cosine similarity between the embeddings. As this model was trained for English, the Japanese and Korean lyrics were automatically translated using Google Translator before obtaining the embeddings.

We started by examining hierarchical semantic similarity using cross-scape plots [32], as shown in Figure 2. Given a pair of lyrics $\mathbf{X} = \{x_1, \ldots, x_n\}$ and $\tilde{\mathbf{X}} = \{\tilde{x_1}, \ldots, \tilde{x_n}\}$ with n lines each, the first (leftmost) block of the lowest line represents the semantic textual similarity between x_1 and $\tilde{x_1}$ (denoted as $sts(x_1, \tilde{x_1})$) while the last (rightmost) block signifies $sts(x_n, \tilde{x_n})$. The first (leftmost) block of the second-lowest line denotes $sts(x_1 \mathbin{+\!\!+} x_2, \tilde{x_1} \mathbin{+\!\!+} \tilde{x_2})$, and the second block corresponds to $sts(x_2 \mathbin{+\!\!+} x_3, \tilde{x_2} \mathbin{+\!\!+} \tilde{x_3})$. Lastly,

[1] We used `all-MiniLM-L6-v2` [30].

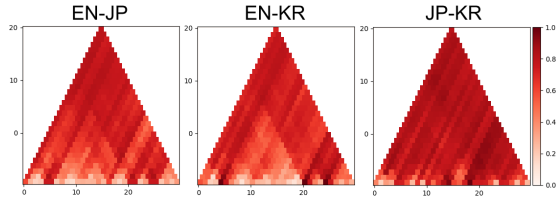


Figure 2. Semantic similarity cross-scape plot for the J-pop song "A Thousand Winds" between English and Japanese (**Left**), English and Korean (**Middle**), and Japanese and Korean (**Right**). Any value less than 0 was regarded as 0.

Line #	English	Japanese (English translation)	sts
1	please do not stand at my grave and weep.	私のお墓の前で (in front of my grave)	0.56
2	I am not there, I do not sleep	泣かないでください (please stop crying)	0.27
1, 2	please do not stand at my grave and weep. I am not there, I do not sleep	私のお墓の前で 泣かないでください (in front of my grave. please stop crying.)	**0.76**

Table 6. Semantic textual similarity (sts) between English and Japanese versions of "A Thousand Winds".

the highest block (line) represents the similarity between the entire lyrics, $sts(x_1 \mathbin{+\!\!+} \cdots \mathbin{+\!\!+} x_n, \tilde{x_1} \mathbin{+\!\!+} \cdots \mathbin{+\!\!+} \tilde{x_n})$.

In each plot of Figure 2, there are semantic disparities at lower levels, but similarities increase at higher (broader) levels. We have two explanations for this. First, the number of musical notes within a single lyric line may be adequate to deliver a specific message in one language but insufficient in another language. Therefore, it is common for a message conveyed in one line in one language to span two lines in another language. As an example, we provide Table 6, which presents the semantic textual similarity (sts) between Japanese and English lyrics of the J-pop song "A Thousand Winds (千の風になって)". As demonstrated in the table, the similarity between Japanese and English at a broader level ($sts(x_1 \mathbin{+\!\!+} x_2, \tilde{x_1} \mathbin{+\!\!+} \tilde{x_2})$) can be higher than at the line level ($sts(x_1, \tilde{x_1})$, $sts(x_2, \tilde{x_2})$) because Japanese generally requires more syllables than English and it often takes two lines in Japanese to express a single-line message in English. Second, the semantic similarities at broader levels can be higher because of grammatical/linguistic differences. Each language has its own natural word order patterns. For example, in the phrase "I'm going to travel to find the gold," it is natural in English to mention "I'm going to travel" before "to find the gold." However, in Japanese and Korean, expressing "to find the gold (金を探しに, 금을 찾으러)" before "I'm going to travel (旅に出る, 떠난다)" is a more typical and natural construction. Table 7 shows that these differences between languages make line-level semantic assessments insufficient. Since lines 1, 2, and 3 in the English version correspond to lines 3, 1, and 2 respectively in the Japanese version, these pairs exhibit low semantic similarities at the line level ($sts(x_1, \tilde{x_1})$, $sts(x_2, \tilde{x_2})$, $sts(x_3, \tilde{x_3})$), while demonstrating higher similarity when considered as a whole ($sts(x_1 \mathbin{+\!\!+} x_2 \mathbin{+\!\!+} x_3, \tilde{x_1} \mathbin{+\!\!+} \tilde{x_2} \mathbin{+\!\!+} x_3)$).

Considering these factors, it becomes evident that

Line #	English	Japanese (English translation)	sts
1	Dare to try and reach out for heaven	望むように生きるなら (If you want to become what you're meant to be)	0.22
2	You must become what you're meant to be	星からの金を求め (to find the gold from stars)	0.27
3	And bring the gold of heaven to the world	一人旅に出るのよ (Dare to embark on a solo journey)	0.13
1-3	Dare to try and reach out for heaven You must become what you're meant to be And bring the gold of heaven to the world	望むように生きるなら星からの金を求め一人旅に出るのよ (Dare to embark on a solo journey if you want to become what you're meant to be to find the gold from stars)	**0.53**

Table 7. Semantic textual similarity (sts) between English and Japanese versions of "Gold von den Sternen".

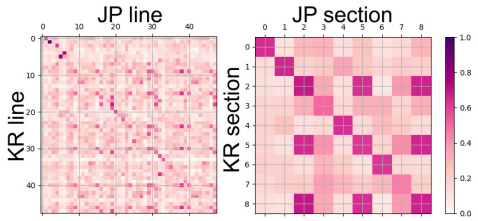


Figure 3. The line-wise (**Left**) and section-wise (**Right**) semantic similarity matrices between Japanese and Korean versions of "Wie wird man seinen Schatten los?"

singable lyric translations do not prioritize line-wise semantic similarity. Rather, we observed that singable translations aim to preserve semantic connections at the section level since the organization of the lyric storyline follows a section-wise approach. To illustrate this, we present Figure 3, which displays both line-wise and section-wise semantic similarity matrices for the Japanese and Korean versions of "How do you get rid of your shadow? (Wie wird man seinen Schatten los?)" from the German musical "Mozart!". As shown in the Figure, the section-wise matrix represents the semantic relatedness more clearly than the line-wise matrix.

Therefore, we propose assessing section-wise semantic relatedness for evaluating singable lyric translation. To achieve this, we define the **semantic similarity** between a pair of lyrics $\mathbf{X} = \{X_1, ..., X_m\}$ and $\tilde{\mathbf{X}} = \{\tilde{X_1}, ..., \tilde{X_m}\}$, consisting of m sections and $n = n(X_1) + \cdots + n(X_m)$ lines, where $n(X_i)$ denotes the number of lines in the i-th section, as follows:

$$Sim_{sem}(\mathbf{X}, \tilde{\mathbf{X}}) = \sum_{i=1}^{m} (\frac{n(X_i)}{n} sts(X_i, \tilde{X_i})). \quad (6)$$

Table 8 compares singable and non-singable lyrics in terms of line-wise semantic similarities ($\frac{1}{n}\sum_{i=1}^{n} sts(x_i, \tilde{x_i})$) and section-wise similarities, using our proposed metric (Sim_{sem}). The table reveals that non-singable translations exhibit high semantic similarity for both line-wise and section-wise measures, with similar values for each. In contrast, singable translations show low line-wise similarity, as expected, since they do not prioritize line-wise semantic similarity. However, when evaluated using section-wise similarity, they display a level of similarity comparable to that between "Machine learning is so easy" and "Deep learning is so straightforward", which is 0.623 when measured with the same pre-trained model [30].

Source	Target	Singable		Non-singable	
		line	section	line	section
English	Japanese	0.40	0.54	0.64	0.74
	Korean	0.42	0.55	0.70	0.76
Japanese	English	0.47	0.59	0.66	0.72
	Korean	0.52	0.61	0.77	0.79
Korean	English	0.53	0.63	0.78	0.81
	Japanese	0.52	0.61	0.73	0.75

Table 8. The average line-wise semantic similarity and section-wise semantic similarity (Sim_{sem}) of singable and non-singable lyrics.

6. DISCUSSIONS AND CONCLUSIONS

In this paper, we introduced a computational evaluation framework for singable lyric translation, grounded in the musical, linguistic, and cultural understanding of lyrics, comprised of four evaluation metrics, line syllable count distance (Dis_{syl}), phoneme repetition similarity (Sim_{pho}), musical structure distance (Dis_{mus}), and semantic similarity (Sim_{sem}). These metrics are designed to ensure that the translated lyrics maintain the integrity of melodies, degree of phoneme repetition, structural coherence, and semantics of the original lyrics. Our framework is automated, guaranteeing objectivity and efficiency in terms of time and cost. We showed the efficacy of our evaluation metrics by offering comparative statistics between singable and non-singable lyrics. In addition, our analysis revealed that the degree of phoneme repetition in the original lyrics is frequently mirrored in the translated lyrics, musically similar sections tend to share the same phonemes and display comparable degrees of phoneme repetition, and section-wise analysis is better suited for evaluating semantic similarity for lyric translation than line-wise analysis.

Nonetheless, there remains room for improvement. Although we have assembled a singable lyrics dataset, aligned across English, Japanese, and Korean, our dataset has some limitations; it lacks musical information and its volume is limited. As a result, we have not been able to incorporate musical notes into our experiment or conduct comparative studies across various genres. We recognize that an ideal lyric translation evaluation system should take into account the relationship between musical notes and phonemes, as well as adapt to different genres. Moreover, although we have endeavored to incorporate cultural understandings of poetry in different languages, we acknowledge the need for deeper cultural considerations. For example, we noticed that cultural similarities might have an impact on the extent of semantic similarities. This is demonstrated in an English translation of "MIC Drop", a K-pop song by BTS originally written in Korean, made by YouTuber Iris Phuong. The translated singable lyrics do not include a translation of the term "hyodo (효도, taking care of parents)" as there is no direct equivalent in English, while the Japanese version of the song effortlessly conveys the concept as "koukou (孝行)". In the future, we aim to expand our dataset to contribute more to lyric translation studies and to further explore the impact of genre and cultural influences on lyric translation.

7. ACKNOWLEDGMENTS

This work was supported in part by JST CREST Grant Number JPMJCR20D4.

8. REFERENCES

[1] M. Mateo, "Music and translation," *Handbook of translation studies*, vol. 3, pp. 115–121, 2012.

[2] Ş. Susam-Sarajeva, "Translation and music: Changing perspectives, frameworks and significance," *The Translator*, vol. 14, no. 2, pp. 187–200, 2008.

[3] S. Spaeth, "Translating to music," *The Musical Quarterly*, vol. 1, no. 2, pp. 291–298, 1915.

[4] P. Low, "Translating songs that rhyme," *Perspectives: Studies in translatology*, vol. 16, no. 1-2, pp. 1–20, 2008.

[5] ——, "The pentathlon approach to translating songs," in *Song and significance*. Brill, 2005, pp. 185–212.

[6] 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)*. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 729–743. [Online]. Available: https://aclanthology.org/2022.findings-acl.60

[7] L. Ou, X. Ma, M.-Y. Kan, and Y. Wang, "Songs across borders: Singable and controllable neural lyric translation," *arXiv preprint arXiv:2305.16816*, 2023.

[8] 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," *arXiv preprint arXiv:2303.15705*, 2023.

[9] K. Watanabe and M. Goto, "Lyrics information processing: Analysis, generation, and applications," in *Proceedings of the 1st Workshop on NLP for Music and Audio (NLP4MusA)*, 2020, pp. 6–12.

[10] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: A method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting of the Association for Computational Linguistics (ACL)*, 2002, pp. 311–318.

[11] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Text summarization branches out*, 2004, pp. 74–81.

[12] S. Banerjee and A. Lavie, "METEOR: An automatic metric for MT evaluation with improved correlation with human judgments," in *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, 2005.

[13] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "BERTScore: Evaluating text generation with bert," in *International Conference on Learning Representations (ICLR)*, 2019.

[14] I. Marc, "Travelling songs: On popular music transfer and translation," *IASPM Journal*, vol. 5, no. 2, pp. 3–21, 2015.

[15] E. C. Hui-tung, "Translation of songs," *An Encyclopedia of Practical Translation and Interpreting*, p. 351, 2019.

[16] Y. Yang, Y. Zhang, C. Tar, and J. Baldridge, "PAWS-X: A cross-lingual adversarial dataset for paraphrase identification," 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. 3687–3692.

[17] H. S. Drinker, "On translating vocal texts," *The Musical Quarterly*, vol. 36, no. 2, pp. 225–240, 1950.

[18] J. Franzon, "Three dimensions of singability. An approach to subtitled and sung translations," *Text and Tune. On the Association of Music and Lyrics in Sung Verse. Bern: Peter Lang*, pp. 333–346, 2015.

[19] J. P. G. Mahedero, A. MartÍnez, P. Cano, M. Koppenberger, and F. Gouyon, "Natural language processing of lyrics," in *Proceedings of the 13th Annual ACM International Conference on Multimedia*, ser. MULTIMEDIA '05. Association for Computing Machinery, 2005, pp. 475–478.

[20] K. Watanabe and M. Goto, "A chorus-section detection method for lyrics text." in *Proceedings of the 21th International Conference on Music Information Retrieval (ISMIR)*, 2020, pp. 351–359.

[21] R. Apter and M. Herman, "Translating art songs for performance: Rachmaninoff's six choral songs," *Translation Review*, vol. 84, no. 1, pp. 27–42, 2012.

[22] A. Bain, "English composition and rhetoric,(2nd american ed.)," *New York: D. Appleton and Company*, 1867.

[23] N. Manabe, "Globalization and Japanese creativity: Adaptations of Japanese language to rap," *Ethnomusicology*, vol. 50, no. 1, pp. 1–36, 2006.

[24] Y. B. Yoon and B. L. Derwing, "A language without a rhyme: Syllable structure experiments in Korean," *Canadian Journal of Linguistics/Revue canadienne de linguistique*, vol. 46, no. 3-4, pp. 187–237, 2001.

[25] A. K. Syrdal and H. S. Gopal, "A perceptual model of vowel recognition based on the auditory representation of American English vowels," *The Journal of the Acoustical Society of America*, vol. 79, no. 4, pp. 1086–1100, 1986.

[26] K. Hanson, "Formal variation in the rhymes of Robert Pinsky's the inferno of dante," *Language and Literature*, vol. 12, no. 4, pp. 309–337, 2003.

[27] C. Ito, Y. Kang, and M. Kenstowicz, "The adaptation of Japanese loanwords into Korean," *MIT Working Papers in Linguistics*, vol. 52, no. 2006, pp. 65–104, 2006.

[28] S.-E. Chang, "Enhancement effects of clear speech and word-initial position in Korean glides." *The Journal of the Acoustical Society of America*, 2017.

[29] J. Li, M. Galley, C. Brockett, J. Gao, and W. B. Dolan, "A diversity-promoting objective function for neural conversation models," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics (NACCL) : Human Language Technologies*, 2016, pp. 110–119.

[30] W. Wang, F. Wei, L. Dong, H. Bao, N. Yang, and M. Zhou, "MiniLM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, pp. 5776–5788, 2020.

[31] 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, pp. 3982–3992.

[32] 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 Conference on Music Information Retrieval (ISMIR)*, 2019, pp. 423–430.

CHORUS-PLAYLIST: EXPLORING THE IMPACT OF LISTENING TO ONLY CHORUSES IN A PLAYLIST

Kosetsu Tsukuda **Masahiro Hamasaki** **Masataka Goto**
National Institute of Advanced Industrial Science and Technology (AIST), Japan
{k.tsukuda, masahiro.hamasaki, m.goto}@aist.go.jp

ABSTRACT

When people listen to playlists on a music streaming service, they typically listen to each song from start to end in order. However, what if it were possible to use a function to listen to only the choruses of each song in a playlist one after another? In this paper, we call this music listening concept "chorus-playlist," and we investigate its potential impact from various perspectives such as the demand and the objectives for listening to music with chorus-playlist. To this end, we conducted a questionnaire-based online user survey involving 214 participants. Our analysis results suggest reusable insights, including the following: (1) We show a high demand for listening to existing playlists with the chorus-playlist approach. We also reveal preferred options for chorus playback, such as adding crossfade transitions between choruses. (2) People listen to playlists with chorus-playlist for various objectives. For example, when they listen to their own self-made playlists, they want to boost a mood or listen to music in a specific context such as work or driving. (3) There is also a high demand for playlist creation on the premise of continuous listening to only the choruses of the songs in a playlist. The diversities of artists, genres, and moods are more important when creating such a playlist than when creating a usual playlist.

1. INTRODUCTION

The chorus of a song is one of the most distinctive parts in the song. In terms of acoustic aspects, it has been reported that the chorus tends to have louder sound, contain heavier instrumentation and additional vocals, and include the highest-pitch vocal note in a song [1–3]. In terms of cognitive aspects, the chorus tends to be the catchiest, most memorable, and most salient part of a song for emotional expression [4–7]. Moreover, the chorus is often characterized by the property of being a song's most repeated section [8, 9]. Because of these characteristics, the chorus has attracted academic attention. For example, research has been conducted on music structure analysis including chorus detection [8–35] and its use for music summary generation [36, 37]. In addition, music datasets specializing in choruses have been made publicly available [38].

© K. Tsukuda, M. Hamasaki, and M. Goto. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).
Attribution: K. Tsukuda, M. Hamasaki, and M. Goto, "Chorus-Playlist: Exploring the Impact of Listening to Only Choruses in a Playlist", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.*, Milan, Italy, 2023.

As for general music listening habits beyond choruses, the amount of time spent listening to music through playlists on music streaming services has increased [39–42]. It has also been reported that this listening time is longer than the time of listening to music via albums [43]. On the services, both playlists that are created by general users and those that are created by professional curators or automatically generated are widely available [42, 44–46]. As a result, Spotify has over 4 billion playlists, for example [42]. With the popularity of playlists, many studies have been conducted on playlist recommendation, generation, and analysis [40, 42, 44, 47–76].

Given the importance of choruses and playlists, we focus on a listening approach in which only the choruses of the songs in a playlist are played one after another. Certain smartphone music player applications such as Vocacolle App by DWANGO Co., Ltd., MIXTRAX App by Pioneer Corporation, and KENWOOD Music Control have provided a function to continuously play the choruses of songs or parts including the choruses. However, there has been no academic discussion on the impact of this listening approach. In this paper, we refer to the concept of continuous listening to only the choruses of songs in a playlist as "chorus-playlist." This concept can be applied not only when a user listens to playlists that she created but also when she listens to playlists created by other users. Under this concept, a user could create a playlist on the premise of continuous listening to only the choruses of the playlist's songs. Hence, the goal of this paper is to reveal the usefulness of chorus-playlist and provide reusable insights.

To achieve this goal, we conducted a questionnaire-based online user survey involving 214 participants. Our contributions can be summarized as follows.

- To our knowledge, this is the first study investigating the impact of continuous listening to only the choruses of songs in a playlist.
- We reveal user preferences for chorus playback in a playlist (*e.g.*, users prefer to add crossfade transitions between choruses). We also show a high demand and certain user objectives for listening to playlists with chorus-playlist. For example, people often want to listen to their own self-made playlists to boost a mood or in a specific context such as work or driving.
- We show that people tend to be willing to create a playlist for continuously listening to only the choruses of the songs in the playlist. We also reveal important properties in creating such playlists (*e.g.*, the diversity of moods, genres, and artists).

- According to the survey results, we suggest new research topics for the music information retrieval (MIR) community (*e.g.*, song recommendation for users who listen to music with chorus-playlist). We also make several proposals for music streaming platforms to attract users (*e.g.*, when using the chorus-playlist approach, people would be willing to listen to playlists containing hit songs to efficiently check them out).
- We have made a portion of the survey results publicly available on the web to support future studies [1].

2. RELATED WORK

2.1 Chorus Analysis and Detection

The chorus is distinctive compared to other sections of a song in terms of acoustic, structural, and cognitive aspects. In terms of acoustic aspects, it has been known that choruses tend to be louder, to contain heavier instrumentation and more vocals, and to have the highest-pitch vocal note in a song [1–3]. More recently, Balen et al. [77] revealed that choruses have a smaller dynamic range and a greater variety of MFCC-measurable timbres, as compared to other sections. Regarding structural aspects, choruses are usually repeated more than other sections such as intros and verses [8, 9]. As for cognitive aspects, choruses are the catchiest and most memorable sections for listeners [4, 5, 7]. For artists, too, choruses are distinctive in that they are the most salient sections for emotional expression [6]. Given these characteristics, we focus here on choruses as song excerpts, rather than other sections. As we will show in section 7, the chorus is more preferred than other sections for continuous listening in a playlist.

Because of the importance of choruses, many studies have addressed musical structure analysis, including chorus detection, based on music audio signals [8–15, 17–24, 26,27,29–32,34,35] or lyrics [16,25,28,33]. The accuracy of identifying the chorus section is approximately 80-90% in terms of the F-measure [12,22,29,33]. Accordingly, the feasibility of implementing the chorus-playlist concept on music streaming services is sufficiently high. For songs in which the correct chorus cannot be detected, it is also possible to have users on a service manually correct choruses through a collective intelligence approach [78].

2.2 Playlist Analysis and Recommendation

On today's music streaming services, playlists have become a central way to listen to music [40–42]. The majority of playlists on services are created by general users rather than professional music enthusiasts [44]. Users create playlists not only for their own listening but also to share their musical preferences with other users such as friends and followers [44, 53, 70]. It has also been reported that playlists can be characterized by certain properties [40, 62, 68, 69, 74, 76] such as song order [42] and low diversity in terms of both artists and genres [70]. In this paper, too, we report objectives for listening to music with chorus-playlist (section 5) and important properties for creating a playlist to continuously listen to only the choruses of songs in the playlist (section 6).

Although playlists are actively created by users, it is time consuming to manually create a playlist [44]. To ease the process, two approaches have been studied: assisted playlist creation [49–52, 54, 55, 71] and automatic playlist generation [47, 48, 56, 58–61, 63–67, 72, 73, 75]. In song recommendation for a playlist or generation of a playlist, it is typical to consider the song order and/or the audio similarity between songs. Furthermore, there are several studies that automatically extract prominent sections (not only limited to choruses) from individual songs and generate DJ mixes [79] or medleys [80] by connecting them. Music streaming services such as Spotify and Deezer provide functions for automatic playlist generation to promote song discovery by users [45, 46]. In this paper, according to our survey results, we discuss not only new approaches for these research topics but also how to encourage users on music streaming services to more actively interact with playlists and discover novel songs.

3. PARTICIPANTS

We recruited participants for our survey via an online research company in Japan. We limited the participants to those who are Japanese, listened to music an average of at least one day per week via any music streaming service, and had created at least 10 playlists on the service. We paid 51.6 USD (7,000 JPY) to each participant. Although 222 participants answered the questionnaire in sections 4, 5, 6, and 7 through a web browser, to make the analysis results more reliable, we removed the answers from eight participants who submitted improper responses to a free-response question. The remaining 214 participants were diverse in both gender and age range: 89 males (10s: 1; 20s: 29; 30s: 35; 40s: 17; 50s: 7), and 125 females (10s: 1; 20s: 45; 30s: 38; 40s: 25; 50s: 16).

4. PREFERENCE FOR CHORUS PLAYBACK

4.1 Chorus Playback Choices

As explained in section 1, the concept of chorus-playlist enables a user to continuously listen to only the choruses of songs in a playlist. However, some users may prefer to add crossfade transitions between choruses. In this section, we investigate the preferences for chorus playback in chorus-playlist in terms of the following three choices.

- TimePreChorus: the playback time before the chorus. The options are "no playback," "5 seconds," and "10 seconds." "No playback" means that only the choruses are continuously played, without any part of the song before the chorus.
- Crossfade: whether 1-second crossfade transitions are added between songs. The options are "on" and "off."
- TimeChorus: the playback time for the chorus. The options are "15 seconds," "30 seconds," and "adaptive." In the case of "15 (resp. 30) seconds," 15 (resp. 30) seconds on a song is played from the beginning of the

[1] The data can be downloaded from `https://github.com/ktsukuda/chorus_playlist`.

first chorus [2] . In the case of "adaptive," the first chorus is played from beginning to end regardless of its length.

Hereafter, we refer to a combination of these three choices in the form of a triplet such as (TimePreChorus, Crossfade, TimeChorus) = (5 seconds, off, adaptive).

4.2 Dataset

In this survey, the participants listened to playlists that we provided. To reduce bias due to the played songs, we used 120 songs created by professional musicians that we commissioned. That is, we guaranteed that the participants had never listened to any of the 120 songs. Instead of using chorus detection methods introduced in section 2.1, a music expert manually labeled the start and end times of each song's first chorus to prevent detection errors. For 26 songs that started with the chorus, the expert labeled the start and end times of the second chorus, because the mood and/or beat of such leading choruses are sometimes different from those of the second and subsequent choruses. The average and standard deviation of chorus lengths for the 120 songs were 29.0 and 7.66 seconds, respectively, and 52 songs had choruses longer than 30 seconds.

As the participants had various music preferences, we created diverse playlists by sampling the songs to be included in playlists as follows. First, the 120 songs were plotted in the valence-arousal (VA) space according to their audio features. Next, we applied the k-means algorithm and classified the 120 songs in the VA space into three clusters. We created three playlists by sampling five songs at random for each playlist from one cluster. Similarly, we created three more playlists from another cluster. Finally, we created three playlists containing diverse songs in terms of their moods by randomly selecting one song from each of the two previous clusters and three songs from the remaining cluster. Each playlist's song order was also determined randomly. In total, we created nine playlists that each comprised five songs. Note that there were no song overlaps between any pairs of playlists.

For each playlist, there were 18 total option combinations (3 options for TimePreChorus $\times$ 2 options for Crossfade $\times$ 3 options for TimeChorus). For example, for (5 seconds, on, adaptive), we first created an MP3 file by cutting each song in a playlist from five seconds before the first chorus to the end of the chorus and then connecting the songs with crossfade transitions. We then created an MP4 file to simulate the participants' experience of listening to the playlist on a smartphone application. Specifically, given an MP3 file, we created an MP4 file in which the MP3's playlist was played and images changed at the same time the song changed in the playlist. Each image contained a song's title and artist name like in the music playback screen of a music player application (Figure 1) [3] .

4.3 Procedure

First, we investigated the participants' preferred option combinations. To this end, three dropdown lists with the

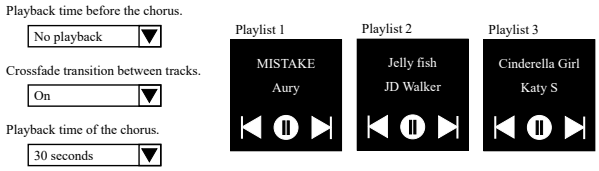


Figure 1. Interface example from our user survey. In this example, when a participant selects the options (no playback, on, 30 seconds), three playlists satisfying this combination are displayed.

options for each choice were presented to the participants. For each participant, three playlists were selected at random from the set of playlists described in section 4.2. Once the participant had selected an option for each choice, the three playlists (MP4 files) satisfying that option combination were displayed (Figure 1). When the participant changed the combination, the displayed playlists were also changed to those satisfying the new option combination [4] . After listening to playlists for any option combination, the participants reported their most preferred combination such as (10 seconds, on, 30 seconds), by selecting those options from the dropdown lists [5] .

Even if a participant chose "adaptive" as the preferred option for TimeChorus, she may have liked "30 seconds" almost as much. Accordingly, after the above investigation, we investigated the option preferences including such subtle differences. To this end, we displayed each of the 18 options with a six-point Likert scale ranging from "not preferred at all" to "very preferred," and we asked the participants to rate their preferences for each option.

4.4 Results

Table 1 and Figure 2 indicate the results for the first and second investigations, respectively. In Table 1, we can see that the most popular combination was (5 seconds, on, adaptive). Even participants who chose other combinations also tended to prefer each of these options. In fact, for the results shown in Figure 2, paired Wilcoxon signed-rank tests with Bonferroni correction revealed that the median of "5 seconds" was statistically higher than the other two options for TimePreChorus at $p < 0.01$ [6] . Similarly, "on" for Crossfade and "adaptive" for TimeChorus were statistically higher than the other options at $p < 0.01$. In particular, it was not obvious that "5 seconds" was the most preferred option for TimePreChorus, making this a useful, reusable insight for realizing chorus-playlist.

A music streaming service could offer the concept of chorus-playlist by implementing a function that enables users to listen to only choruses for all existing playlists on the service. If a service provided this function, it would be ideal to enable users to play playlists with arbitrary option combinations, as we did, to reflect users' preferences. If it

[2] Thus, if the first chorus is less than 15 (resp. 30) seconds long, the song continues play after the chorus until the total playback time reaches 15 (resp. 30) seconds.

[3] The images also include icons of pause, next, and previous buttons.

[4] Note that the songs contained in the three playlists did not change; only the options for playing them were changed.

[5] It was not mandatory to listen to the playlists for all 18 option combinations. In fact, most participants reported their most preferred combination by narrowing down their preferences while switching options and listening to the corresponding playlists.

[6] Throughout this paper, ** (*) in a figure denotes a statistical difference at $p < 0.01$ ($p < 0.05$).

Table 1. Preference distribution for option combinations.

TimePreChorus	Crossfade	TimeChorus	# participnats
No playback	On	15 seconds	10 (4.67%)
		30 seconds	3 (1.40%)
		adaptive	38 (17.76%)
	Off	15 seconds	8 (3.74%)
		30 seconds	0
		adaptive	8 (3.74%)
5 seconds	On	15 seconds	26 (12.15%)
		30 seconds	14 (6.54%)
		adaptive	**49 (22.90%)**
	Off	15 seconds	1 (0.47%)
		30 seconds	1 (0.47%)
		adaptive	16 (7.48%)
10 seconds	On	15 seconds	3 (1.40%)
		30 seconds	6 (2.80%)
		adaptive	22 (10.28%)
	Off	15 seconds	1 (0.47%)
		30 seconds	1 (0.47%)
		adaptive	7 (3.27%)

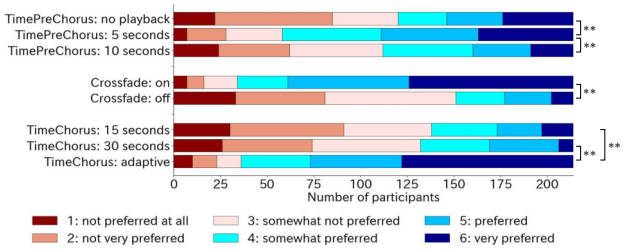


Figure 2. Preference distributions for each option.

is difficult to implement such a flexible function, chorus-playlist should be provided with the option combination (5 seconds, on, adaptive), which should maximize the average user satisfaction according to this section's results.

5. DEMAND FOR CHORUS-PLAYLIST

We described above how to implement a chorus-playlist function. In this section, we investigate the demand for listening to existing playlists with such a function.

5.1 Procedure

First, we showed the following description: "Suppose that a function to play only the choruses of the songs in a playlist has become available on the music streaming service that you usually use. Please rate on a scale from 1 (unwilling) to 6 (very willing) how much you would like to use this function to listen to existing playlists that you created." When the answer was "unwilling," they were asked to respond freely on why he/she did not want to use it. Otherwise, when the answer was one of the remaining five items, they were asked to respond freely with at least one objective for listening to playlists with the function. We did not set a cap on the number of responses.

Next, in a similar way, we asked the participants to indicate their willingness on a 6-point scale to use the function to listen to existing playlists created by other users. According to their willingness, they were asked to provide free responses as they did for the first question.

5.2 Results

Figure 3 shows the answer distribution. We refer to playlists created by the participant and by other users "self-made playlists" and "others' playlists." Only 11 and 3 participants answered "unwilling" for self-made playlists and

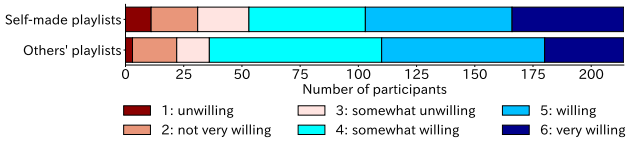


Figure 3. Distribution of the willingness to listen to playlists with the chorus-playlist function.

others' playlists, respectively. The most popular reason for unwillingness was "I believe that there is value in listening to the entire song, including parts other than the chorus." On the other hand, because 75.2 % (161) and 82.7 % (177) participants for the two respective types of playlists answered "very willing," "willing," or "somewhat willing," we conclude that there is a sufficiently high demand for chorus-playlist.

We manually grouped the free responses on their objectives. When a response included multiple objectives corresponding to different groups, it was assigned to multiple groups. Table 2 lists the top 10 objectives in terms of the group sizes for each of the two kinds of playlists. Each number in parentheses indicates the number of participants who gave that objective. Below, we discuss the results.

In the case of self-made playlists, the top three objectives could be achieved just by continuously listening to the choruses of songs in a playlist. For example, the first objective was "boost a mood." As self-made playlists usually contain songs that match the user's own music preferences, the user's mood would be boosted even when listening to playlists in the usual way [68]. Nevertheless, it is interesting that the participants answered that they wanted to further boost their mood by listening to only choruses. It would be beneficial to recommend songs for a playlist that are suitable for boosting a user's mood when the user listens to only the choruses in the playlist. For the second objective, the participants answered with various contexts such as "work" and "driving." When listening to a playlist with the chorus-playlist function in a specific context, there could be various reasons such as "increasing concentration" and "relaxing." It would be an interesting future work to conduct a more detailed analysis of the contexts in which the conventional playlist listening approach or the chorus-playlist approach are preferred. As for the fourth objective, it is known that people consider it valuable to listen to music with others and let others listen to their favorite songs [81–84]. However, because it takes much time for others to listen to all the songs in a playlist, people may hesitate to introduce their favorite songs. Thus, the participants answered that they wanted to efficiently introduce others such as friends or family to their favorite songs when they listen to music together in person. That is, chorus-playlist could encourage people to interact with others through music in the real world.

On the other hand, regarding the top six objectives for others' playlists, although those objectives could be achieved by conventional playlist listening, the participants wanted to use the chorus-playlist function to achieve the objectives more efficiently in a shorter time. In particular, as seen from the first, second, fourth, and fifth objectives, there is a strong demand for efficiently discovering and lis-

Table 2. Top 10 free-response objectives for listening to playlists with the chorus-playlist function.

Rank	Self-made playlist	Others' playlist
1	Boost a mood (79)	Find unfamiliar songs that suit my preference (109)
2	Listen to a playlist in a specific context (68)	Listen to hit songs (48)
3	Listen to a playlist within a limited time (46)	Learn other people's music preferences (44)
4	Recommend my favorite songs to others (36)	Listen to songs by unfamiliar artists (37)
5	Explore desired songs (35)	Listen to songs in unfamiliar genres (26)
6	Listen to many songs (23)	Preview a playlist (18)
7	Listen to various songs (20)	Listen to a playlist in a specific context (13)
8	Recall songs listened to in the past (18)	Refer to a playlist for creating my playlists (12)
9	Listen to only the choruses of my favorite songs (19)	Listen to a playlist for a change of pace (9)
10	Sing songs in a playlist (12)	Boost a mood (6)

tening to unfamiliar songs. Accordingly, if a music streaming service provides playlists consisting of hit songs of the past week, songs by a specific artist, or songs in a specific genre with the chorus-playlist function, many users would likely listen to those playlists by using the function. This would enable users to find more new favorite songs and help increase their music listening activity.

In summary, we have revealed that chorus-playlist can generate new interactions between people and music especially when they listen to self-made playlists. Moreover, as discussed above, the results in Table 2 can provide useful insights for both researchers and music streaming services.

6. IMPORTANT PROPERTIES FOR CREATING PLAYLISTS IN CHORUS-PLAYLIST CONCEPT

6.1 Procedure

In section 5, we assumed application of the chorus-playlist concept to existing playlists. However, by taking the concept a step further, a user could create a playlist on the premise of continuous listening to only the choruses in the playlist's songs (for simplicity, we refer to creating such a playlist as "creating a chorus-playlist"). Therefore, we asked the participants how much they would like to create chorus-playlists on the music streaming service that they used regularly. They answered with their willingness on a 6-point scale from "unwilling" to "very willing." We also told the participants that they could also listen to each song from beginning to end through, not just the chorus.

When users create playlists, they consider certain properties such as the diversity of artists and the song order. Hence, we wondered whether there are any differences regarding the importance of these properties when creating a chorus-playlist as compared to creating a usual playlist. To answer this question, we considered the following 11 properties derived from past studies [42, 70]. (1) SongHit: including songs with high popularity. (2) SongNew: including new songs in terms of the release dates. (3) ArtistSame: including as many songs by the same artist as possible. (4) ArtistDiv: including songs by as many different artists as possible. (5) GenreSame: including as many songs in the same genre as possible. (6) GenreDiv: including songs in as many different genres as possible. (7) MoodSame: including as many songs with the same musical mood as possible. (8) MoodDiv: including songs with as many different moods as possible. (9) SongOrder: the song order in the playlist. (10) SongTop: the first song in the playlist. (11) SongLast: the last song in the playlist.

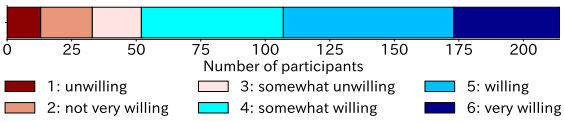


Figure 4. Willingness to create a chorus-playlist.

The participants who answered the first question with anything other than "unwilling" rated the importance of each property in creating a chorus-playlist and in creating a usual playlist on a 6-point scale from "not at all important" to "very important." The 11 properties were displayed in a random order to each participant.

6.2 Results

Figure 4 shows that chorus-playlist creation has the potential to be a new way of enjoying music, because 75.7% (162) participants answered "very willing," "willing," or "somewhat willing," while only 6.07% (13) participants answered "unwilling." Next, as shown in Figure 5, paired Wilcoxon signed-rank tests indicated that statistical differences between the two playlist types were observed for nine properties [7]. Hence, we can say that people tended to emphasize different properties when creating a chorus-playlist as compared to creating a usual playlist. Existing studies on song recommendation for playlists or playlist generation have proposed various methods focusing on the song order in a playlist [54, 61, 64, 73]. However, for chorus-playlist, the SongOrder, FirstSong, and LastSong properties were relatively less important. In contrast, hit songs and new songs were more important for chorus-playlist. Furthermore, the results revealed the importance of diversity in terms of artists, genres, and moods for chorus-playlist. It has been reported that the diversities of artists and genres tend to be low in usual playlists [70]; however, to support users creating chorus-playlists, it would be important to recommend songs to diversify such properties. These results thus open up new recommendation approaches in the MIR community.

7. PLAYBACK METHOD COMPARISON

We have revealed a high demand to try chorus-playlist. In this section, we investigate whether the chorus-playlist playback method is preferred to other playback methods.

7.1 Procedure

For comparison, we used the following two types of playlists. (1) Head-playlist: a user continuously listens to

[7] Figure 5 shows the results for the 201 participants besides the 13 participants who answered "unwilling." The same statistical differences were obtained even with only the aforementioned 162 participants.

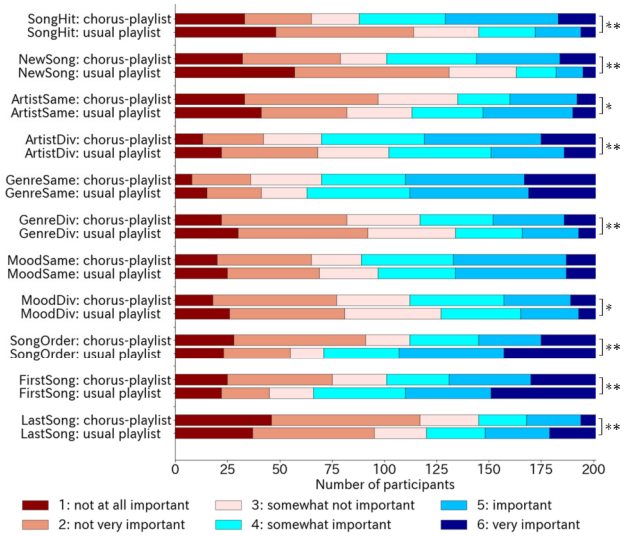


Figure 5. Property importance for playlist creation.

only the head sections of each song in a playlist. (2) 30sec-playlist: a user continuously listens to only the parts after the first 30 seconds of each song in a playlist. We adopted 30 seconds according to the preview samples on a music streaming service (Deezer) [85].

Similarly to the investigation described in section 4, we had the participants listen to each type of playlist by selecting options for two choices. In the head-playlist case, the two choices were 1-second crossfade (options: "on" and "off") and the playing time from the head of each song (options: "15 seconds" and "30 seconds"). In the case of 30sec-playlist, the two choices were 1-second crossfade (options: "on" and "off") and the playing time after the first 30 seconds of each song (options: "15 seconds" and "30 seconds"). The participants listened to playlists with each playback method by using their favorite option combinations. Here, each participant was assigned three playlists containing the same songs as those in section 4. Then, they were asked to rate their willingness to listen to self-made playlists with each method on a 6-point scale.

7.2 Results

Figure 6 shows the results. Note that the chorus-playlist results are repeated from "self-made playlists" in Figure 3. Paired Wilcoxon signed-rank tests with Bonferroni correction revealed that the median for chorus-playlist was statistically higher than the medians for head-playlist and 30sec-playlist. It thus became clear that it was not enough to simply play any part of the songs in a playlist continuously, but that it was important for users to be able to play choruses continuously.

8. DISCUSSION AND CONCLUSION

In this paper, we have studied the concept of chorus-playlist. The reusable insights obtained from our user survey can be summarized as follows.

- We showed that there is a high demand for chorus-playlist. When the participants listened to songs in a playlist with chorus-playlist, they tended to prefer to listen to 5 seconds before the chorus, add crossfade transitions between songs, and listen to the chorus from beginning to end. As discussed in section 7.2, it is more important to play choruses continuously than to play other sections continuously.
- As seen in Table 2, the objectives for listening to music with chorus-playlist were largely different between self-made playlists and others' playlists. In particular, certain objectives for self-made playlists were unexpected, in that people wanted to enjoy music in a new way with chorus-playlist for objectives such as boosting their mood. These results could provide guidelines for researchers and services to consider new research topics and activate user interaction, respectively.
- We revealed a high demand for creating a chorus-playlist. As in Figure 5, hit songs, new songs, and the diversities of artists, genres, and moods are more important when creating a chorus-playlist than when creating a usual playlist. These results also provide new viewpoints for studies on assisted playlist creation.

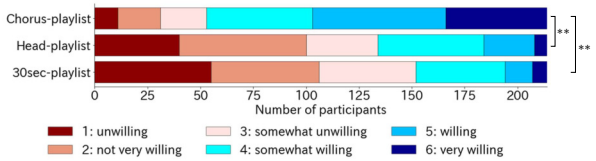


Figure 6. Willingness for three playlist types.

We acknowledge a limitation of this paper in that all the participants in our user survey were Japanese. Because peoples' music preferences, listening behaviors, and music itself vary widely from country to country [86–90], not all of the findings reported here can be generalized. Nevertheless, we believe that this study provides a worthwhile contribution as a first step toward understanding the impact of the chorus-playlist concept. At the same time, this limitation indicates further possibilities such as investigating the differences among countries and cultures. The publicly available dataset of results from our user survey will enable researchers to perform such comparisons.

Another limitation is that the participants did not experience chorus-playlist on the music streaming services they usually used. However, because they answered the survey after experiencing the chorus-playlist concept by listening to the playlists that we provided, we think that they could sufficiently imagine the situation of listening to self-made playlists and others' playlists with the chorus-playlist approach. We currently provide the chorus-playlist function in a music-related smartphone application (Vocacolle App) and web service (Kiite [8]). In the future, we will investigate the function's usage in those more realistic environments.

Finally, although we considered only the first chorus of a song (except when a song started with the chorus), the final chorus tends to be longer and contain heavier instrumentation than other choruses [1]. Therefore, it would also be an interesting future work to investigate the impact of differences between choruses in chorus-playlist listening. Moreover, the concept of listening to only choruses can be applied not only to playlists but also to other song lists such as album track lists, which could further enrich and diversify people's music listening experience.

[8] https://kiite.jp

9. ACKNOWLEDGMENTS

This work was supported in part by JST CREST Grant Number JPMJCR20D4 and JSPS KAKENHI Grant Number JP21H04917, Japan.

10. REFERENCES

[1] K. Stephenson, *What to listen for in rock: A stylistic analysis*. Yale University Press, 2002.

[2] J. Covach, "Form in rock music," *Engaging music: Essays in music analysis*, pp. 65–76, 2005.

[3] W. Everett, *The foundations of rock: from "Blue suede shoes" to "Suite: Judy blue eyes"*. Oxford University Press, 2008.

[4] R. Middleton, "Song form," *The Continuum Encyclopedia of Popular Music of the World*, vol. 2, pp. 513–519, 2003.

[5] A. Eronen, "Chorus detection with combined use of MFCC and chroma features and image processing filters," in *Proceedings of the 10th International Conference on Digital Audio Effects*, ser. DAFx 2007, 2007, pp. 229–236.

[6] S. Hult, L. B. Kreiberg, S. S. Brandt, and B. T. Jónsson, "Analysis of the effect of dataset construction methodology on transferability of music emotion recognition models," in *Proceedings of the 2020 International Conference on Multimedia Retrieval*, ser. ICMR 2020, 2020, pp. 316–320.

[7] 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, pp. 1–18, 2021.

[8] J. Paulus, M. Müller, and A. Klapuri, "State of the art report: Audio-based music structure analysis," in *Proceedings of the 11th International Society for Music Information Retrieval Conference*, ser. ISMIR 2010, 2010, pp. 625–636.

[9] M. Müller, P. Grosche, and N. Jiang, "A segment-based fitness measure for capturing repetitive structures of music recordings," in *Proceedings of the 12th International Society for Music Information Retrieval Conference*, ser. ISMIR 2011, 2011, pp. 615–620.

[10] J. Foote, "Automatic audio segmentation using a measure of audio novelty," in *Proceedings of the 2000 IEEE International Conference on Multimedia and Expo*, ser. ICME 2000, 2000, pp. 452–455.

[11] M. Cooper and J. Foote, "Summarizing popular music via structural similarity analysis," in *Proceedings of the 2003 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, ser. WASPAA 2003, 2003, pp. 127–130.

[12] M. Goto, "A chorus section detection method for musical audio signals and its application to a music listening station," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 5, pp. 1783–1794, 2006.

[13] M. Müller and S. Ewert, "Joint structure analysis with applications to music annotation and synchronization," in *Proceedings of the 9th International Conference on Music Information Retrieval*, ser. ISMIR 2008, 2008, pp. 389–394.

[14] J. Paulus and A. Klapuri, "Music structure analysis using a probabilistic fitness measure and a greedy search algorithm," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 6, pp. 1159–1170, 2009.

[15] J. Serrà, M. Müller, P. Grosche, and J. L. Arcos, "Unsupervised detection of music boundaries by time series structure features," in *Proceedings of the 26th AAAI Conference on Artificial Intelligence*, ser. AAAI 2012, 2012, pp. 1613–1619.

[16] A. Baratè, L. A. Ludovico, and E. Santucci, "A semantics-driven approach to lyrics segmentation," in *Proceedings of the 8th International Workshop on Semantic and Social Media Adaptation and Personalization*, ser. SMAP 2013, 2013, pp. 73–79.

[17] O. Nieto and T. Jehan, "Convex non-negative matrix factorization for automatic music structure identification," in *Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, ser. ICASSP 2013, 2013, pp. 236–240.

[18] F. Kaiser and G. Peeters, "A simple fusion method of state and sequence segmentation for music structure discovery," in *Proceedings of the 14th International Society for Music Information Retrieval Conference*, ser. ISMIR 2013, 2013, pp. 257–262.

[19] H. Grohganz, M. Clausen, N. Jiang, and M. Müller, "Converting path structures into block structures using eigenvalue decompositions of self-similarity matrices," in *Proceedings of the 14th International Society for Music Information Retrieval Conference*, ser. ISMIR 2013, 2013, pp. 209–214.

[20] G. Peeters and V. Bisot, "Improving music structure segmentation using lag-priors," in *Proceedings of the 15th International Society for Music Information Retrieval Conference*, ser. ISMIR 2014, 2014, pp. 337–342.

[21] B. McFee and D. Ellis, "Analyzing song structure with spectral clustering," in *Proceedings of the 15th International Society for Music Information Retrieval Conference*, ser. ISMIR 2014, 2014, pp. 405–410.

[22] C.-H. Yeh, W.-Y. Tseng, C.-Y. Chen, Y.-D. Lin, Y.-R. Tsai, H.-I. Bi, Y.-C. Lin, and H.-Y. Lin, "Popular music

representation: Chorus detection & emotion recognition," *Multimedia Tools and Applications*, vol. 73, pp. 2103–2128, 2014.

[23] 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*, ser. ISMIR 2015, 2015, pp. 531–537.

[24] J. B. L. Smith and M. Goto, "Using priors to improve estimates of music structure," in *Proceedings of the 17th International Society for Music Information Retrieval Conference*, ser. ISMIR 2016, 2016, pp. 554–560.

[25] K. Watanabe, Y. Matsubayashi, N. Orita, N. Okazaki, K. Inui, S. Fukayama, T. Nakano, J. B. L. Smith, and M. Goto, "Modeling discourse segments in lyrics using repeated patterns," in *Proceedings of the 26th International Conference on Computational Linguistics*, ser. COLING 2016, 2016, pp. 1959–1969.

[26] G. Sargent, F. Bimbot, and E. Vincent, "Estimating the structural segmentation of popular music pieces under regularity constraints," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 2, pp. 344–358, 2017.

[27] 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 2018 IEEE International Conference on Acoustics, Speech and Signal Processing*, ser. ICASSP 2018, 2018, pp. 106–110.

[28] M. Fell, Y. Nechaev, E. Cabrio, and F. Gandon, "Lyrics segmentation: Textual macrostructure detection using convolutions," in *Proceedings of the 27th International Conference on Computational Linguistics*, ser. COLING 2018, 2018, pp. 2044–2054.

[29] Y. Huang, S. Chou, and Y. Yang, "Pop Music Highlighter: Marking the emotion keypoints," *Transactions of the International Society for Music Information Retrieval*, vol. 1, no. 1, pp. 68–78, 2018.

[30] A. Maezawa, "Music boundary detection based on a hybrid deep model of novelty, homogeneity, repetition and duration," in *Proceedings of the 2019 IEEE International Conference on Acoustics, Speech and Signal Processing*, ser. ICASSP 2019, 2019, pp. 206–210.

[31] G. Shibata, R. Nishikimi, E. Nakamura, and K. Yoshii, "Statistical music structure analysis based on a homogeneity-, repetitiveness-, and regularity-aware hierarchical hidden semi-markov model," in *Proceedings of the 20th International Society for Music Information Retrieval Conference*, ser. ISMIR 2019, 2019, pp. 268–275.

[32] 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*, ser. ISMIR 2020, 2020, pp. 23–29.

[33] K. Watanabe and M. Goto, "A chorus-section detection method for lyrics text," in *Proceedings of the 21st International Society for Music Information Retrieval Conference*, ser. ISMIR 2020, 2020, pp. 351–359.

[34] 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 2021 IEEE International Conference on Acoustics, Speech and Signal Processing*, ser. ICASSP 2021, 2021, pp. 566–570.

[35] 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 2022 IEEE International Conference on Acoustics, Speech and Signal Processing*, ser. ICASSP 2022, 2022, pp. 416–420.

[36] S. Gao and H. Li, "Popular song summarization using chorus section detection from audio signal," in *Proceedings of the IEEE 17th International Workshop on Multimedia Signal Processing*, ser. MMSP 2015, 2015, pp. 1–6.

[37] Y. Gao, Y. Shen, X. Zhang, S. Yu, and W. Li, "Music summary detection with state space embedding and recurrence plot," in *Proceedings of the 6th Conference on Sound and Music Technology*, ser. CSMT 2019, 2019, pp. 41–51.

[38] K. Zhang, H. Zhang, S. Li, C. Yang, and L. Sun, "The PMEmo dataset for music emotion recognition," in *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval*, ser. ICMR 2018, 2018, pp. 135–142.

[39] J. H. Lee and N. M. Waterman, "Understanding user requirements for music information services," in *Proceedings of the 13th International Society for Music Information Retrieval Conference*, ser. ISMIR 2012, 2012, pp. 253–258.

[40] L. Porcaro and E. Gomez, "20 years of playlists: A statistical analysis on popularity and diversity," in *Proceedings of the 20th International Society for Music Information Retrieval Conference*, ser. ISMIR 2019, 2019, pp. 130–136.

[41] I. Kamehkhosh, G. Bonnin, and D. Jannach, "Effects of recommendations on the playlist creation behavior of users," *User Modeling and User-Adapted Interaction*, vol. 30, no. 2, pp. 285–322, 2020.

[42] H. V. Schweiger, E. Parada-Cabaleiro, and M. Schedl, "Does track sequence in user-generated playlists matter?." in *Proceedings of the 22nd International Society*

for Music Information Retrieval Conference, ser. ISMIR 2021, 2021, pp. 618–625.

[43] K. Sakurai, R. Togo, T. Ogawa, and M. Haseyama, "Controllable music playlist generation based on knowledge graph and reinforcement learning," *Sensors*, vol. 22, no. 10, pp. 135–142, 2022.

[44] G. Bonnin and D. Jannach, "Automated generation of music playlists: Survey and experiments," *ACM Computing Surveys*, vol. 47, no. 2, pp. 1–35, 2014.

[45] K. Jacobson, V. Murali, E. Newett, B. Whitman, and R. Yon, "Music personalization at Spotify," in *Proceedings of the 10th ACM Conference on Recommender Systems*, ser. RecSys 2016, 2016, p. 373.

[46] T. Bontempelli, B. Chapus, F. Rigaud, M. Morlon, M. Lorant, and G. Salha-Galvan, "Flow Moods: Recommending music by moods on deezer," in *Proceedings of the 16th ACM Conference on Recommender Systems*, ser. RecSys 2022, 2022, pp. 452–455.

[47] J. J. Aucouturier and F. Pachet, "Scaling up music playlist generation," in *Proceedings of the 2002 IEEE International Conference on Multimedia and Expo*, ser. ICME 2002, 2002, pp. 105–108.

[48] B. Logan, "Content-based playlist generation: Exploratory experiments," in *Proceedings of the 3rd International Conference on Music Information Retrieval*, ser. ISMIR 2002, 2002, pp. 295–296.

[49] S. Pauws and B. Eggen, "PATS: Realization and user evaluation of an automatic playlist generator," in *Proceedings of the 3rd International Conference on Music Information Retrieval*, ser. ISMIR 2002, 2002, pp. 222–230.

[50] R. V. Gulik and F. Vignoli, "Visual playlist generation on the artist map," in *Proceedings of the 6th International Conference on Music Information Retrieval*, ser. ISMIR 2005, 2005, pp. 520–523.

[51] E. Pampalk, T. Pohle, and G. Widmer, "Dynamic playlist generation based on skipping behavior," in *Proceedings of the 6th International Conference on Music Information Retrieval*, ser. ISMIR 2005, 2005, pp. 634–637.

[52] S. Pauws and S. V. D. Wijdeven, "User evaluation of a new interactive playlist generation concept," in *Proceedings of the 6th International Conference on Music Information Retrieval*, ser. ISMIR 2005, 2005, pp. 638–643.

[53] S. J. Cunningham, D. Bainbridge, and A. Falconer, ""more of an art than a science": Supporting the creation of playlists and mixes," in *Proceedings of the 7th International Conference on Music Information Retrieval*, ser. ISMIR 2006, 2006, pp. 240–245.

[54] E. Pampalk and M. Gasser, "An implementation of a simple playlist generator based on audio similarity measures and user feedback," in *Proceedings of the 7th International Conference on Music Information Retrieval*, ser. ISMIR 2006, 2006, pp. 389–390.

[55] N. Oliver and L. Kreger-Stickles, "PAPA: Physiology and purpose-aware automatic playlist generation," in *Proceedings of the 7th International Conference on Music Information Retrieval*, ser. ISMIR 2006, 2006, pp. 250–253.

[56] S. Pauws, W. Verhaegh, and M. Vossen, "Fast generation of optimal music playlists using local search," in *Proceedings of the 7th International Conference on Music Information Retrieval*, ser. ISMIR 2006, 2006, pp. 138–143.

[57] B. Fields, C. Rhodes, M. A. Casey, and K. Jacobson, "Social playlists and bottleneck measurements: Exploiting musician social graphs using content-based dissimilarity and pairwise maximum flow values," in *Proceedings of the 9th International Conference on Music Information Retrieval*, ser. ISMIR 2008, 2008, pp. 559–564.

[58] A. Flexer, D. Schnitzer, M. Gasser, and G. Widmer, "Playlist generation using start and end songs," in *Proceedings of the 9th International Conference on Music Information Retrieval*, ser. ISMIR 2008, 2008, pp. 173–178.

[59] K. Bosteels, E. Pampalk, and E. E. Kerre, "Evaluating and analysing dynamic playlist generation heuristics using radio logs and fuzzy set theory," in *Proceedings of the 10th International Society for Music Information Retrieval Conference*, ser. ISMIR 2009, 2009, pp. 351–356.

[60] F. Maillet, D. Eck, G. Desjardins, and P. Lamere, "Steerable playlist generation by learning song similarity from radio station playlists," in *Proceedings of the 10th International Society for Music Information Retrieval Conference*, ser. ISMIR 2009, 2009, pp. 345–350.

[61] B. McFee and G. R. G. Lanckriet, "The natural language of playlists," in *Proceedings of the 12th International Society for Music Information Retrieval Conference*, ser. ISMIR 2011, 2011, pp. 537–542.

[62] J. H. Lee, "How similar is too similar?: Exploring users' perceptions of similarity in playlist evaluation," in *Proceedings of the 12th International Society for Music Information Retrieval Conference*, ser. ISMIR 2011, 2011, pp. 109–114.

[63] B. McFee and G. R. G. Lanckriet, "Hypergraph models of playlist dialects," in *Proceedings of the 13th International Society for Music Information Retrieval Conference*, ser. ISMIR 2012, 2012, pp. 343–348.

[64] J. L. Moore, S. Chen, T. Joachims, and D. Turnbull, "Learning to embed songs and tags for playlist prediction," in *Proceedings of the 13th International Society for Music Information Retrieval Conference*, ser. ISMIR 2012, 2012, pp. 349–354.

[65] G. Bonnin and D. Jannach, "Evaluating the quality of generated playlists based on hand-crafted samples," in *Proceedings of the 14th International Society for Music Information Retrieval Conference*, ser. ISMIR 2013, 2013, pp. 263–268.

[66] D. Jannach, L. Lerche, and I. Kamehkhosh, "Beyond "hitting the hits": Generating coherent music playlist continuations with the right tracks," in *Proceedings of the 9th ACM Conference on Recommender Systems*, ser. RecSys '15, 2015, pp. 187–194.

[67] T. Nakano, J. Kato, M. Hamasaki, and M. Goto, "PlaylistPlayer: An interface using multiple criteria to change the playback order of a music playlist," in *Proceedings of the 21st International Conference on Intelligent User Interfaces*, ser. IUI 2016, 2016, pp. 186–190.

[68] M. Pichl, E. Zangerle, and G. Specht, "Understanding playlist creation on music streaming platforms," in *Proceedings of the 2016 IEEE International Symposium on Multimedia*, ser. ISM 2016, 2016, pp. 475–480.

[69] C. Chung, Y. Chen, and H. H. Chen, "Exploiting playlists for representation of songs and words for text-based music retrieval," in *Proceedings of the 18th International Society for Music Information Retrieval Conference*, ser. ISMIR 2017, 2017, pp. 478–485.

[70] R. Dias, D. Gonçalves, and M. J. Fonseca, "From manual to assisted playlist creation: A survey," *Multimedia Tools and Applications*, vol. 76, no. 12, pp. 14 375–14 403, 2017.

[71] L. F. Pontello, P. H. F. Holanda, B. Guilherme, J. P. V. Cardoso, O. Goussevskaia, and A. P. C. D. Silva, "Mixtape: Using real-time user feedback to navigate large media collections," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 13, no. 4, pp. 1–22, 2017.

[72] R. M. Bittner, M. Gu, G. Hernandez, E. J. Humphrey, T. Jehan, H. McCurry, and N. Montecchio, "Automatic playlist sequencing and transitions," in *Proceedings of the 18th International Society for Music Information Retrieval Conference*, ser. ISMIR 2017, 2017, pp. 442–448.

[73] S. Shih and H. Chi, "Automatic, personalized, and flexible playlist generation using reinforcement learning," in *Proceedings of the 19th International Society for Music Information Retrieval Conference*, ser. ISMIR 2018, 2018, pp. 168–174.

[74] S. Y. Park, A. Laplante, J. H. Lee, and B. Kaneshiro, "Tunes together: Perception and experience of collaborative playlists," in *Proceedings of the 20th International Society for Music Information Retrieval Conference*, ser. ISMIR 2019, 2019, pp. 723–730.

[75] A. Patwari, N. Kong, J. Wang, U. Gargi, M. Covell, and A. Jansen, "Semantically meaningful attributes from co-listen embeddings for playlist exploration and expansion," in *Proceedings of the 21st International Society for Music Information Retrieval Conference*, ser. ISMIR 2020, 2020, pp. 527–533.

[76] Z. Li, M. Song, S. Duan, and Z. Wang, "Are users attracted by playlist titles and covers? Understanding playlist selection behavior on a music streaming platform," *Journal of Innovation & Knowledge*, vol. 7, no. 3, pp. 1–14, 2022.

[77] J. V. Balen, J. A. Burgoyne, F. Wiering, and R. C. Veltkamp, "An analysis of chorus features in popular song," in *Proceedings of the 14th Society of Music Information Retrieval Conference*, ser. ISMIR 2013, 2013, pp. 107–112.

[78] M. Goto, K. Yoshii, H. Fujihara, M. Mauch, and T. Nakano, "Songle: A web service for active music listening improved by user contributions," in *Proceedings of the 12th International Society for Music Information Retrieval Conference*, ser. ISMIR 2011, 2011, pp. 311–316.

[79] A. Kim, S. Park, J. Park, J.-W. Ha, T. Kwon, and J. Nam, "Automatic DJ mix generation using highlight detection," in *Proceedings of the 18th International Society for Music Information Retrieval Conference*, ser. ISMIR 2017, 2017, pp. 1–2.

[80] Y.-S. Huang, S.-Y. Chou, and Y.-H. Yang, "Generating music medleys via playing music puzzle games," in *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, ser. AAAI 2018, 2018, pp. 2281–2288.

[81] A. Bassoli, J. Moore, S. Agamanolis, and H. C. Group, "tunA: Local music sharing with handheld Wi-Fi devices," in *Proceedings of the 5th Wireless World Conference 2004*, ser. WWC 2004, 2004, pp. 1–23.

[82] M. Håkansson, M. Rost, and L. E. Holmquist, "Gifts from friends and strangers: A study of mobile music sharing," in *Proceedings of the 10th European Conference on Computer-Supported Cooperative Work*, ser. ECSCW 2007, 2007, pp. 311–330.

[83] M. Håkansson, M. Rost, M. Jacobsson, and L. E. Holmquist, "Facilitating mobile music sharing and social interaction with Push!Music," in *Proceedings of the 40th Annual Hawaii International Conference on System Sciences*, ser. HICSS 2007, 2007, pp. 87–96.

[84] K. Tsukuda, K. Ishida, M. Hamasaki, and M. Goto, "Kiite Cafe: A web service for getting together virtually to listen to music." in *Proceedings of the 22nd*

International Society for Music Information Retrieval Conference, ser. ISMIR 2021, 2021, pp. 697–704.

[85] K. M. Ibrahim, J. Royo-Letelier, E. V. Epure, G. Peeters, and G. Richard, "Audio-based auto-tagging with contextual tags for music," in *Proceedings of the 2020 IEEE International Conference on Acoustics, Speech and Signal Processing*, ser. ICASSP 2020, 2020, pp. 16–20.

[86] X. Hu and J. H. Lee, "A cross-cultural study of music mood perception between American and Chinese listeners," in *Proceedings of the 13th International Society for Music Information Retrieval Conference*, ser. ISMIR 2012, 2012, pp. 535–540.

[87] Y. 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*, ser. ISMIR 2012, 2012, pp. 19–24.

[88] X. Hu, J. H. Lee, K. Choi, and J. S. Downie, "A cross-cultural study on the mood of K-POP songs," in *Proceedings of the 15th International Society for Music Information Retrieval Conference*, ser. ISMIR 2014, 2014, pp. 385–390.

[89] M. Liu, X. Hu, and M. Schedl, "Artist preferences and cultural, socio-economic distances across countries: A big data perspective," in *Proceedings of the 18th International Society for Music Information Retrieval Conference*, ser. ISMIR 2017, 2017, pp. 103–111.

[90] C. Bauer and M. Schedl, "Global and country-specific mainstreaminess measures: Definitions, analysis, and usage for improving personalized music recommendation systems," *PLOS ONE*, vol. 14, no. 6, pp. 1–36, 2019.

Papers – Session VII

SUPPORTING MUSICOLOGICAL INVESTIGATIONS WITH INFORMATION RETRIEVAL TOOLS: AN ITERATIVE APPROACH TO DATA COLLECTION

David Lewis[1] **Elisabete Shibata**[2] **Andrew Hankinson**[3] **Johannes Kepper**[4]
Kevin R. Page[1] **Lisa Rosendahl**[2] **Mark Saccomano**[4] **Christine Siegert**[2]
[1] University of Oxford, UK [2] BeethovenHaus Bonn, Germany
[3] RISM Digital Centre, Switzerland [4] Paderborn University, Germany
david.lewis@oerc.ox.ac.uk

ABSTRACT

Digital musicology research often proceeds by extending and enriching its evidence base as it progresses, rather than starting with a complete corpus of data and metadata, as a consequence of an emergent research need.

In this paper, we consider a research workflow which assumes an incremental approach to data gathering and annotation. We describe tooling which implements parts of this workflow, developed to support the study of nineteenth-century music arrangements, and evaluate the applicability of our approach through interviews with musicologists and music editors who have used the tools. We conclude by considering extensions of this approach and the wider implications for digital musicology and music information retrieval.

1. INTRODUCTION

Digital humanities research often extends and enriches an evidence base – in the form of digital, machine-accessible corpora – as it progresses, mirroring a methodological process of evidence gathering and preparation that is common and accepted in analogue research. Rather than assuming complete corpus encoding as a prerequisite for digital scholarship, we anticipate that research subjects will more usually be found in un-transcribed and only minimally-catalogued documents. A researcher or team can thereby more effectively support their work by digitising, transcribing, and annotating a corpus incrementally. Resource limitations will generally mean that this is most efficiently carried out in an incomplete way, producing partial editions of short extracts or individual instrumental parts, instead of a complete corpus as an outcome of the investigation. To support this mode of digital scholarship, we propose that incremental workflows, which manipulate and analyse incomplete sources, should be an explicit consideration for applied MIR assemblies.

We present an example of this approach, from the *Beethoven in the House* project, where musical arrangements and miscellaneous music publications aimed at a domestic market are the subject of the scholarship. In this case, little of the music has been published in modern editions, and no digital editions existed at the start of the research process. Some sources had been photographed and published online before the project began, and the remainder were digitised at the request of the project. Our data model abstracts the musical structures from the surface presented by digital representations themselves, so that our tools can switch transparently between working with digital scores and facsimile images, with measure detection supporting the transition. We also use Linked Data and user-authored, web-based storage, which supports the enrichment of institutional data resources, such as library images, without requiring that scholars have write access to those servers. We focus on chained components and data compatibility rather than trying to build end-to-end tools. Our ambition is that, at the end of the process, the digital tools support our own research, as well as supporting reusability and transparency, since the 'working materials' can be published along with the finished results.

In this paper, we consider a research workflow which assumes an incomplete and incremental approach to data gathering and annotation. We describe tooling implementing this workflow, and evaluate the applicability of the approach through interviews with musicologists and music editors who have used the tools. We conclude by considering extensions of this approach and the wider implications for digital musicology and MIR.

2. MUSICOLOGISTS AS DIGITAL RESEARCHERS

Most Information Retrieval implementations are optimised from the perspective of a 'whole' or 'complete' corpus, produced by some prior acts of digitisation, being interrogated by a user motivated by a single, explicit information need. This approach facilitates the optimisation of retrieval tool engineering, since the elements of the system are well known, and the quality of tools can be transparently

© D. Lewis, E. Shibata, A. Hankinson, J. Kepper, K. Page, L. Rosendahl, M. Saccomano, and C. Siegert. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** D. Lewis, E. Shibata, A. Hankinson, J. Kepper, K. Page, L. Rosendahl, M. Saccomano, and C. Siegert, "Supporting musicological investigations with information retrieval tools: an iterative approach to data collection", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

quantified, assessed, evaluated, and compared. Meanwhile Bates's model of berry picking [1] is based on the observation that information needs often develop during the user's interactions with a system, as a part of a research process that takes new findings into account in the search. Different information-seeking strategies and their modes of search and scope of application (whether based on content, features or metadata) are further teased out by Weigl et al [2]. While this does not replace or reject the engineering of MIR tools based on concrete requirements, design, and evaluation, it does suggest we should consider such tools being recomposed as components within a multitude of individualised workflows – where the overall object of the composite workflow cannot be determined a priori. We reflect that this is especially true when MIR tools are used as a means to undertake curiosity-driven research, as they often are in support of digital musicology.

A similar pattern is identified for data as well as tools. Fenlon et al. [3] note the role of selecting and gathering data in the research process so that, as the investigation evolves, so may the subset of the corpus being studied. More recently, Oberbichler et al. [4] have observed that the separation between the management of digital materials and their analysis is less clear for humanities scholarship. They note that the clarity and separation of workflows and responsibilities in digitising, organising, and interrogating collections that make for efficient, maintainable solutions may be problematic in these domains.

For notated music, where data entry remains expensive in terms of time and effort, separating between digitisation, digital editing, metadata organisation, and research can mean that much musical heritage is ruled out from digital research, as digital editors become unwitting gatekeepers of our history. This can have the effect of channelling research into canonical composers and works, and diverting it from less-well-represented areas and niche and regional music [5]. The lack of encoded corpora appropriate to their research has long been identified as an important problem for musicologists [6, 7]. Although it is true that these issues could be addressed by comprehensive and complete mass digitisation and encoding, in the absence of this, an alternative strategy may be required.

We have seen that berry picking can be extended to accept that the research process involves partial and changing research questions, and even that during the investigation, the researcher may add to, correct or enrich the metadata [2]. An alternative interaction model might extend berry picking to acknowledge that this is true for the data itself. Clearly, this may pose problems for statistical evaluation of IR tools, and necessitates consideration of alternative approaches to system and workflow design. Nonetheless we can demonstrate that it is a mode of use aligned with the needs – and limited resources – of digital musicologists.

Given limited resources, we cannot assume that a single scholar, or even a funded research project, can transcribe the complete corpus of music that might be relevant to their investigations – including any comparison or control groups – prior to research commencing, and even producing a complete digitisation by the end of their investigations may prove impractical. Creating an expectation of the prior existence of these primary objects of study may feed the sense of "disconnect between this research strand and musicological users' needs and requirements" identified by Inskip and Wiering [8]. A better approach would accommodate images or partial editions – transcriptions of only a few bars or one instrumental part – created incrementally as the research progresses.

Many of the basic tools that already exist could be made to accommodate this approach well, indeed the extra information that may be available in a digital environment at a later research stage may help them, supporting a bootstrapping approach to training or parameter tuning. Without musicologist-facing, high-level tools built on these, researchers are more likely to resort to less machine-accessible approaches, such as pen and paper or local spreadsheets.

In this paper we explore this interaction model through a set of prototypes. In the next section, we describe a workflow and tooling designed to support musicological research in previously digitally unavailable music, and discuss how an incremental approach can be supported, before evaluating the approach in subsequent sections.

3. SUPPORTING RESEARCH WITH INCREMENTAL AND INCOMPLETE CORPORA

Musicology, and indeed research more broadly, may involve many activities and strategies, whose selection will be informed both by research topic (see [2]) and the stage at which the research stands. For example, a researcher may start with a literature exploration, then start reviewing music scores through a catalogue, selecting a set of potential subjects to look at more closely and then focus down later. The researcher may scan through the scores, selecting works or passages for further consideration, and rejecting others. This might be followed by closer engagement with the chosen texts, often relating them to extra-musical information. Finally, their investigations will be written up formally.

Teasing apart the steps of this example, and when they are most likely to happen in a research life cycle, we can see the following:

1. Literature exploration (early phase)
2. Catalogue exploration (early phase)
3. Workset selection (early phase)
4. Content exploration (mid phase)
5. Content analysis (mid phase)
6. Connecting music with extra-musical material (mid phase)
7. Visualisation and reporting (end phase)

This is not intended as a complete catalogue of research steps, but illustrates common components, and helps ground our observations. Each of these steps will decompose into tasks that may or may not be carried out

Phase	Step	Example activity	Example tools and media
Early	1. Literature exploration	Makes up-to-date literature survey	RILM, JSTOR Google Scholar Physical browsing
	2. Catalogue exploration	Explores the repertory; identifies a superset for more attention	Library catalogues RISM IMSLP, CPDL Physical browsing
	3. Workset selection	Looks at the music, scans through scores to identify works or passages for detailed consideration	RISM IMSLP, CPDL Specialised corpora Physical sources Image digitisation
Mid	4. Content exploration	Close reading of scores, identifying distinctive parameters that support an emerging thesis	OMR Measure detection Sonic visualiser Piano
	5. Content analysis	Lists spacing and instrumentation of chords at cadences	Humdrum toolkit Music21 Sonic visualiser Spreadsheet Paper
	6. Making Connections	Associates particular orchestration approaches with review and theory texts	Spreadsheet Paper
End	7. Visualisation & reporting	Writes and publishes a journal article	Journal, Published edition Recording, Dataset

Table 1. A typical set of steps in a research lifecycle, with example activities and tools. Although this appears as a list, scholars may jump between these, or pursue several at the same time. The **Beethoven in the House Annotator** supports stages 4 and 5, producing data suitable for stage 7.

in a digital environment, and although broadly sequential, a musicologist may jump backwards at any point to supplement the data they already have. To support such flexible research patterns, we believe it is important to create an ecosystem of tools that read or write compatible data, facilitating researcher-directed methodologies for tool selection and task ordering.

4. THE BEETHOVEN IN THE HOUSE ANNOTATOR: A TOOL SUPPORTING MID PHASE RESEARCH

To investigate the feasibility of an 'incremental' interaction model with MIR tools, we have developed a tool to support an active musicological investigation which also embodies the 'mid phase' of the research life cycle described above, focussing particularly on steps 4 and 5. The tool's main purpose is to bring together digitised resources in the form of images and digital scores, and allow a musicologist to view them in a browser, selecting specific extracts for study and then annotating those with scholarly commentary. The resulting annotations are stored, and can be shared and published, including references to the pertinent selections from the digital music resources.

A user entering the **Beethoven in the House Annotator** first selects items to explore in a 'library' view – a listing which displays metadata about available musical works, their arrangements and digital resources available. Because the annotator is designed to handle comparisons of the same passage of music as it is realised in different versions, the selected resources are displayed one above another to aid analysis.

Once works are selected and loaded into the display pane, a musicologist can point and click on individual notes and measures, or click and drag to select larger regions, whether the resource is a facsimile image or a rendered score encoding. Individual selections can be annotated, but also parallel passages in different versions of a work ('Musical Material') can be identified (figure 1, left), and these structures themselves annotated (figure 1, right).

Previous annotations can also be viewed and themselves annotated. Thus, the tool can be used for quick browsing or juxtaposition of music and metadata and for detailed labelling of the content.

The **Beethoven in the House Annotator** is built as a web application, and implemented as a MELD (Music Encoding and Linked Data) application [9] [1]. As a baseline provision we assume the materials underpinning the musicologist's investigation are available in image form via IIIF [2] as welll as digital MEI scores when these are available. We further assume that the musicologist has the tools and skills to optionally transcribe whole pieces or extracts and save or convert them as MEI (this can be carried out using music typesetting packages such as Sibelius or Mus-

[1] More precisely, we use data models and the graph traversal library from MELD, with Vue-based application code.

[2] The International Image Interoperability Framework provides for standardised image delivery through APIs for content and presentation. Although increasingly widespread in use by research collections in particular, it is not yet comprehensively adopted. For the purposes of our research project, required digitized images were provisioned via a local (private) IIIF server where they were not already available over IIIF from the holding collection.

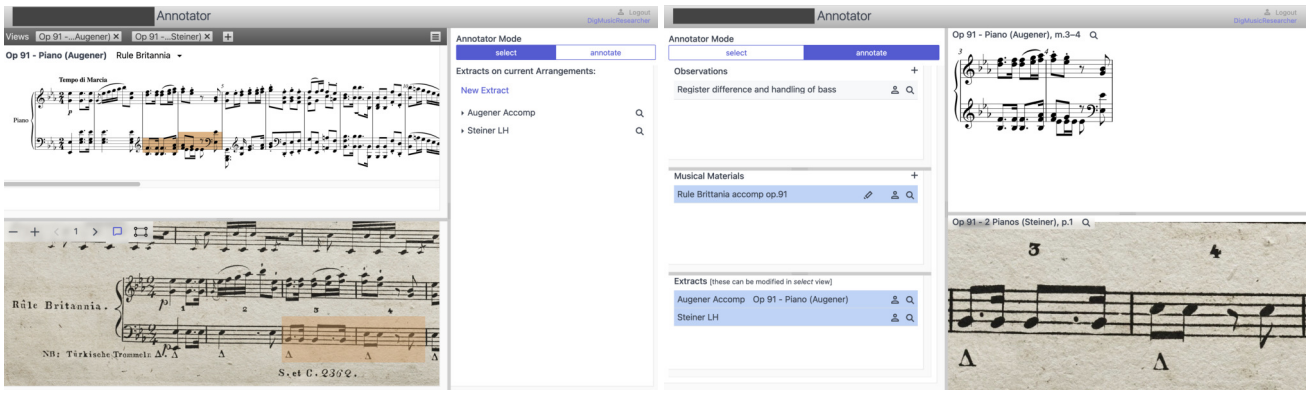


Figure 1. Two screenshots from the **Beethoven in the House Annotator**. **Left:** 'Musical material' – Parallel passages recorded as occurring in two different arrangements of Beethoven's *Wellington's Sieg* (Op. 91). Selections need not be contiguous or limited to a single part. The upper version here has been retrieved from an MEI file and is displayed using Verovio. The lower version is from a IIIF file for which measure locations have been separately detected using the Cartographer tool and stored, along with links to the image, in an otherwise minimal MEI file. **Right:** The annotation view, showing an observation recorded about the musical material shown left. In both cases, structures are saved to the musicologists personal Solid pod, with their login shown upper right.

eScore with the help of plugins).

We also assume the prior existence of well-formed and self-describing catalogue metadata, and we base our prototype on the Linked Open Data published by the Gemeinsame Normdatei (GND) of the Deutsche Nationalbibliothek. We do this with the intention that these could in future be loaded directly where records exist. [3]

Direct image annotation is possible within our tool. The musicologist may prefer to use a labour-saving measure detection tool, such as Cartographer [4] or MEI Friend [10], both of which can output MEI with empty measures and image co-ordinates, and which have been successfully tested with our tool. When provided with MEI and IIIF resources such as these, our annotation tool allows the researcher to annotate the image measure by measure – giving a semantically-richer anchor for the annotation with relatively low input of manual intervention (see the lower pane in figure 1, left). If the researcher needs a finer level of annotation, then they may fill in additional music notation in the MEI, and can indicate the selective nature of the encoding in the MEI header, a process supported by tools such as MEI Friend.

Our application supports textual Web Annotations [11] made onto conceptually abstracted musical extracts rather than directly onto elements or regions of the image or encoding, allowing parallel material occurring in different arrangements of a work to be annotated together and, at a more basic level, allowing the model to remain agnostic to the different types of media used as evidence (figure 1, right, illustrates an example of an annotation on a passage that has been identified in two arrangements, in one case using the MEI transcript, and in the other a IIIF image after a process of measure detection). This uses the Music Annotation Ontology described by Lewis et al [12].

In order to promote data sharing between tools rather than a single monolithic application, user data is stored as Linked Data in Solid Pods [13], distributed online data storage with fine-grained access control, and for which the user can choose provider. This provides a simple mechanism for data portability between applications, given compatible data structures. The structures written can refer to resources anywhere on the web, and traversal carried out by the MELD library will draw them into the application.

In summary, the **Beethoven in the House Annotator** described above supports our proposed workflow in several ways. Firstly, it is conceived as part of a pipeline of tools publishing compatible Linked Data and MEI, and is already interoperable with existing tools. Secondly, it is intended to provide a low barrier for including evidence materials, allowing the use of any web-published IIIF images, complete or partial MEI files, and GND metadata rather than requiring extensive data entry and local servers. Thirdly, it supports the sharing of source data and metadata, along with intermediate observations, within a research team. Finally, as currently implemented, annotations are minimally structured. This supports an evolving research agenda, trading expressiveness against semantic structures.

5. EVALUATING THE BEETHOVEN IN THE HOUSE ANNOTATOR AND ITS WIDER APPLICATION

Whilst the tool's internal development was aimed at satisfying researcher needs within our own project, two rounds of wider evaluation were carried out, timed to coincide with two phases of application development. These evaluation rounds were carried out as semi-structured interviews following shortly after a combination of a presentation about the Annotator and period of time freely ex-

[3] In practice, the GND is not currently usable for client-side applications due to access control headers. This would still allow the use of a server-cached version of the data. Where other metadata is needed, we draw on the WikiData model.

[4] `https://cartographer-app.zenmem.de`

ploring its functionality over a pre-loaded musical library. In the first round interviews were conducted with musicologists recruited via a Studienkolleg (summer school) located at Beethoven Haus, Bonn, in September 2022. In the second round in March 2023, volunteers from staff at the Beethoven Haus were interviewed. In the first round, we interviewed 9 scholars, and 7 in the second round, of whom 2 had previously been interviewed. This allowed us to assess progress with new and returning users.

5.1 Workflow as data pipeline, low barriers for evidence gathering

The application was regarded by all interviewees as useful in the context of larger musicological research projects and editorial work. Since our interviewees were musicologists and editors rather than engineers and, since we did not demonstrate or present any tools for other steps in the process, this support is based primarily on the interviewer's description of the intended wider context for the app rather than concrete experience. Interviewees did raise important concerns regarding the workflow itself, and these are discussed in 5.5 below, and as further work.

5.2 Sharing of evidence and findings

Users that we spoke to were strongly attracted both by the idea of sharing data and annotations and the option of keeping these private or controlling access – either during the research process or separating draft and publishable work. They immediately identified the equivalence of this approach to paper based methods of publication and regarded using publicly shared annotations as "similar to quoting published books", although there are concerns about how to verify and attest its quality. It is clear that these features would be easier to realise given user interfaces optimised for these tasks, since the default management interfaces of Solid providers, our principle medium for publication, generally present usability barriers to newcomers. Nonetheless, one user evaluates that the application has the potential to "bring everything together in a way I haven't experienced before" in terms of gathering and sharing knowledge about musical works. This would support the "Nachprüfbarkeit", or verifiability (literally reviewability) of a conclusion by collecting the evidence in a single place.

Although musicology can appear – at least from its outputs – as the activity of lone scholars, sharing between scholars in an informal way is common, as is the use of student assistance, both of which can benefit from controlled data sharing. Certainly, several participants were explicitly open to a wider set of contributors, one noting that, depending on the quality of the community, "more knowledge can be obtained". Beyond this, other musicological use cases identified by participants are more commonly team or group activities, such as scholarly music editing or pedagogical uses, with sharing either between teacher and student or between students within a class.

This sharing approach is well supported for our own Linked Data structures, but there are concerns with the boundaries of that sharing. For example, a Linked Data structure that is publicly shared could annotate a part of an image or score that is not itself publicly available (perhaps for copyright reasons). This would not render the information in the Linked Data unusable, and the URI itself would remain uniquely identifiable, but for some uses would become unavailable. There is no clear way to deduce that one identified element in an MEI file occurs earlier in the piece than another purely from the URI since these semantics are located in the MEI score. Our use of the Music Annotation Ontology brings more aspects of musical selection into the Linked Data domain, but we do not attempt to export musical meaning encoded in MEI into RDF.

5.3 Minimally-structured annotations

The open nature of the annotations and the **Beethoven in the House Annotator** more generally was very clearly important in allowing the musicologists to identify a wide range of contexts in which it would be useful to them. These covered the full range from studying stages in the development of a particular music edition ("Plattenstadien"), systematic musicology, historical approaches, philology and pedagogy. Participants also identified the ease of linking material, both music and annotation material, which is evidence that our low structure approach may have reduced barriers to use. Beyond this (sometimes implicit) validation of our approach, participants identified some structures in annotations to support navigation and discovery.

In the **Beethoven in the House Annotator**, annotations are edited and viewed separately from the score view. In our first version, this view was purely textual, making them harder to navigate, and placing a strong reliance on user-provided labels. Adding musical previews for the second version enhanced findability, but multiple participants noted that an informal taxonomic labelling, such as tags, would enhance this – especially where annotations are shared between users. Data currently available to the application includes metadata and musical locations (where annotations are made on transcribed sources or images on which measure detection has been run). These are not currently used in the annotation listings, but could be, allowing the navigation by measure and source requested by several participants.

5.4 Application-specific responses

The **Beethoven in the House Annotator** builds on rich underlying data models and a complex range of data sources and technologies. An aspect that emerged from the interviews is that the terms chosen for defining key elements in the model did not translate well when designing a user interface. Most users had difficulty navigating the application because their expectations about the terms used did not match with the meaning given to them in the context of the model. Although the learning curve can be conquered, the interviewees expressed that substituting and simplifying the language (in certain cases, hiding structures) would be more beneficial to a quick acclimation into

the application. Although the general-purpose nature of the tool makes the choice of task-specific language difficult, use of clear domain-specific and task-appropriate terms would have been better received by the musicologists and required less detailed briefing.

Although the functionality of the **Beethoven in the House Annotator** is distinctive in ways that were recognised and appreciated, those participants who have worked with comparable applications commented on affordances that they missed from the other tools. In particular, familiarity with EDIROM tools left some participants missing the more advanced navigation system, with, for example, jumping to measure numbers.

5.5 The workflow outside the application

Our workflow acknowledges the poverty of encoded scores but does not, currently, accommodate the lack of digitised images. These, too, have been created according to particular priorities, which may not reflect those of researchers. Libraries and archives must weigh up rarity, value, appearance, physical condition, use and public impact among many other factors when deciding their digitisation policy. Interviewees expressed particular concern for sources located in institutions for whom the burden posed by digitisation in the first place and publication as IIIF in the second is too great, while private collectors may have no desire to engage in digitisation at all. Even following the suggestion of one interviewee, and supporting user upload of static images – whether to their own Solid Pods, or some public IIIF server operating for the common good – could fall foul of institutional restrictions. This may indicate a need to point our structures at musical regions even where no digital proxy exists at all, something which would require a semantic representation of musical location. Although some progress has been made towards such a representation (see, for example, [14]), further modelling is needed to make a robust system.

Similarly, interviewees speculated about how additions are made to the library that the application presents. Currently, we have no application to support the selection of items from a published catalogue to create and operate on a selected workset (steps 2 and 3 in table 1) or the discovery and data transformation this would require. This has not been the focus of the current research, but it does mean that we have relied on some manual technical interventions that would be unsuitable for the sort of musicologists we target here.

6. CONCLUSIONS AND FURTHER WORK

The research workflow we describe here is one in which a scholar adds and edits data and metadata, and in which research priorities develop throughout. We assert this is expresses, albeit schematically, a common approach in musicological research. Rather than trying to create tools to manage the whole process, we have advocated for smaller tools that can comfortably handle mixed, incomplete and partial data, and accumulate results in a way that is data-compatible with other applications and IR tools that the researcher might use.

The musicologists interviewed identified a range of contexts for the **Beethoven in the House Annotator**. That these went not only beyond our design for it, but also beyond its capabilities provides evidence of the need for and dearth of tools that support such activities and the diversity of approaches that can and should be considered.

Our interviews also point clearly to further work, with early-phase support – in the form of digitisation, search and retrieval, and workset gathering – being priorities that would help researchers prepare their materials for use with mid phase applications such as our own. Candidates for components of such tooling, such as Cartographer, but also Sonic Annotator and MEI Friend, often already exist, and often have elements that directly support their role in an ecosystem of tools, particularly in terms of data compatibility, but are often seen either as entirely standalone tools or built into workflows in task-specific ways that are not generalised.

Our investigation demonstrates that the workflow into which the **Beethoven in the House Annotator** fits is recognised and valued by musicologists. The flexibility of the annotator tool, in terms of data and functionality, presents many opportunities in support of musicological research. Importantly, we show that it supports or replaces activities currently taking place in forms – such as Word documents, spreadsheets or on paper – that provide few opportunities for scholars to take advantage of either MIR tools in data analysis on the one hand or digital transparency and sharing of results on the other. Thus, it is recognised as going beyond reproducing existing methods, by enhancing and extending them.

7. ACKNOWLEDGMENTS

This research was undertaken by the project 'Beethoven in the House: Digital Studies of Domestic Music Arrangements', supported by a UK-Germany funding initiative: in the UK by the Arts and Humanities Research Council (AHRC) project number AH/T01279X/1 and in Germany funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) project number 429039809; with additional support from the UK Software Sustainability Institute Phase 3, funded by the UK Engineering and Physical Sciences Research Council (EPSRC) project number EP/S021779/1.

8. REFERENCES

[1] M. J. Bates, "The design of browsing and berrypicking techniques for the online search interface," *Online Review*, vol. 13, no. 5, pp. 407–424, May 1989. [Online]. Available: https://www.emerald.com/insight/content/doi/10.1108/eb024320/full/html

[2] D. M. Weigl, K. R. Page, P. Organisciak, and J. S. Downie, "Information-seeking in large-scale digital libraries: Strategies for scholarly workset creation,"

in *2017 ACM/IEEE Joint Conference on Digital Libraries, JCDL 2017, Toronto, ON, Canada, June 19-23, 2017*. IEEE Computer Society, 2017, pp. 253–256.

[3] K. Fenlon, M. Senseney, H. E. Green, S. Bhattacharyya, C. Willis, and J. S. Downie, "Scholar-built collections: A study of user requirements for research in large-scale digital libraries," in *Connecting Collections, Cultures, and Communities - Proceedings of the 77th ASIS&T Annual Meeting, ASIST 2014, Seattle, WA, USA, October 31 - November 5, 2014*, ser. Proc. Assoc. Inf. Sci. Technol., vol. 51, no. 1. Wiley, 2014, pp. 1–10. [Online]. Available: https://doi.org/10.1002/meet.2014.14505101047

[4] S. Oberbichler, E. Boroş, A. Doucet, J. Marjanen, E. Pfanzelter, J. Rautiainen, H. Toivonen, and M. Tolonen, "Integrated interdisciplinary workflows for research on historical newspapers: Perspectives from humanities scholars, computer scientists, and librarians," *Journal of the Association for Information Science and Technology*, vol. 73, no. 2, pp. 225–239, 2022. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/asi.24565

[5] A. Kijas, "What does the data tell us?: Representation, canon, and music encoding," 2018, Keynote at Music Encoding Conference, Maryland, 24 May 2018. [Online]. Available: https://medium.com/@kijas/https-medium-com-kijas-what-does-the-data-tell-us-926ba830702f

[6] F. Wiering, "User needs and challenges in digital musicology," 2014, Digital Music Lab Workshop on Analysing Big Music Data, City University London, 19 March 2014. [Online]. Available: https://webspace.science.uu.nl/~wieri103/presentations/WieringLondonDigitalMusicLabFinal.pdf

[7] N. Cook, "Towards the compleat musicologist?" 2005, Invited Keynote for the 6th International Conference on Music Information Retrieval (ISMIR), London 2005. [Online]. Available: https://ismir2005.ismir.net/documents/Cook-CompleatMusicologist.pdf

[8] C. Inskip and F. Wiering, "In their own words: Using text analysis to identify musicologists' attitudes towards technology," in *Proceedings of the 16th International Society for Music Information Retrieval Conference, ISMIR 2015, Málaga, Spain, October 26-30, 2015*, 2015, pp. 455–461.

[9] D. M. Weigl and K. R. Page, "A framework for distributed semantic annotation of musical score: "take it to the bridge!"," in *Proceedings of the 18th International Society for Music Information Retrieval Conference, ISMIR 2017, Suzhou, China, October 23-27, 2017*, S. J. Cunningham, Z. Duan, X. Hu, and D. Turnbull, Eds., 2017, pp. 221–228. [Online]. Available: https://ismir2017.smcnus.org/wp-content/uploads/2017/10/190_Paper.pdf

[10] D. M. Weigl and W. Goebl, "Alleviating the last mile of encoding: The mei-friend package for the atom text editor," in *Proceedings of the Music Encoding Conference, Alicante, 2021*. Humanities Commons, 2022. [Online]. Available: https://hcommons.org/deposits/item/hc:45977/

[11] R. Sanderson, P. Ciccarese, and B. Young, "Web annotation data model," W3C, W3C Recommendation, Feb. 2017, https://www.w3.org/TR/2017/REC-annotation-model-20170223/.

[12] D. Lewis, E. Shibata, M. Saccomano, L. Rosendahl, J. Kepper, A. Hankinson, C. Siegert, and K. R. Page, "A model for annotating musical versions and arrangements across multiple documents and media," in *DLfM '22: 9th International Conference on Digital Libraries for Musicology, Prague Czech Republic, 28 July 2022*, L. Pugin, Ed. ACM, 2022, pp. 10–18.

[13] D. M. Weigl, W. Goebl, A. Hofmann, T. Crawford, F. Zubani, C. C. S. Liem, and A. Porter, "Read/write digital libraries for musicology," in *7th International Conference on Digital Libraries for Musicology, Montréal, Canada, October 16, 2020*. ACM, 2020, pp. 48–52. [Online]. Available: https://doi.org/10.1145/3424911.3425519

[14] R. Viglianti, "The music addressability api: A draft specification for addressing portions of music notation on the web," in *Proceedings of the 3rd International Workshop on Digital Libraries for Musicology*, ser. DLfM 2016. New York, NY, USA: Association for Computing Machinery, 2016, p. 57–60.

OPTIMIZING FEATURE EXTRACTION FOR SYMBOLIC MUSIC

Federico Simonetta[1] **Ana Llorens**[2] **Martín Serrano**[1]
Eduardo García-Portugués[3] **Álvaro Torrente**[1,2]
[1] ICCMU - Instituto Complutense de Ciencias Musicales, Madrid
[2] Universidad Complutense de Madrid, Madrid
[3] Universidad Carlos III, Madrid
didone@iccmu.es

ABSTRACT

This paper presents a comprehensive investigation of existing feature extraction tools for symbolic music and contrasts their performance to determine the set of features that best characterizes the musical style of a given music score. In this regard, we propose a novel feature extraction tool, named musif, and evaluate its efficacy on various repertoires and file formats, including MIDI, MusicXML, and **kern. Musif approximates existing tools such as jSymbolic and music21 in terms of computational efficiency while attempting to enhance the usability for custom feature development. The proposed tool also enhances classification accuracy when combined with other sets of features. We demonstrate the contribution of each set of features and the computational resources they require. Our findings indicate that the optimal tool for feature extraction is a combination of the best features from each tool rather than those of a single one. To facilitate future research in music information retrieval, we release the source code of the tool and benchmarks.

1. INTRODUCTION

Feature extraction is a pivotal task in contemporary machine learning. Music features can be categorized into two main types: symbolic and audio. While audio features have been subject to extensive research, computational techniques for symbolic music remain comparatively underexplored.

In recent years, there has been an increasing interest in analyzing symbolic scores in music. This encompasses studies on composer [1] and style recognition [2], affective computing [3], music generation [4], analysis of performance [5], and interpretation [6]. The symbolic dimension of music concerns the conceptual representation of musical data [7]. This level has been used in the field of Music Information Retrieval (MIR), with particularly successful outcomes when employed to support multimodal approaches [8], which integrate both audio and symbolic levels through audio-to-score alignment techniques [9]. The symbolic level is also crucial for musicologists, as music scores are the most common source for historical music studies. Musicologists rely on computational tools to extract and analyze musical scores on a large scale [10, 11]. However, traditional manual annotations, such as harmony [12] and cadence [13], are time-consuming and prone to errors. Therefore, computational tools are essential for efficient and accurate musicological analysis. Presently, two primary tools are available for extracting features from symbolic music: jSymbolic [14] and music21 [15]. Although both tools are open-source and widely employed, no comprehensive comparison between them has been conducted yet.

In this paper, we propose a novel set of features that is specifically, although not exclusively, tailored for the analysis of 18th-century Italian opera. We have developed a tool for extracting these features, named musif, that is being used for the analysis of operatic music in the Didone project [1] [16]. Here, we conduct a comparative study between musif and other existing tools, thus providing valuable insights into the strengths and weaknesses of each of them. Additionally, we evaluate the efficiency of each tool and demonstrate that musif adds useful features to both music21 and jSymbolic. We observe that, in most cases, a combination of features from multiple tools yields the most powerful feature set. To validate our findings, we test all three tools on various repertoires. We aim to compare the feature sets on file formats with varying levels of representation abilities, such as MIDI, MusicXML, and **kern. While MIDI is widespread in computational studies, it is relatively simplistic for written music; MusicXML and **kern, instead, are less commonly utilized in MIR but provide more accurate representations when dealing with music scores.

The main contributions of this paper are, therefore, threefold. Firstly, we present a new set of features designed for the study of an under-represented repertoire in music computing literature, i.e., 18th-century Italian opera. Secondly, we introduce musif, a new efficient, extensible, and open-source Python tool for feature extraction from symbolic music. Finally, we provide a benchmark of music21,

© F. Simonetta, A. Llorens, M. Serrano, E. García-Portugués, and Á. Torrente. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** F. Simonetta, A. Llorens, M. Serrano, E. García-Portugués, and Á. Torrente, "Optimizing Feature Extraction for Symbolic Music", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

[1] https://didone.eu

jSymbolic, and musif on a variety of repertoires and file formats.

The whole code used for this study, as well as the code used for the proposed tool, is available at `https://github.com/DIDONEproject/music_symbolic_features/`.

2. FEATURE EXTRACTION TOOLS

In this study, we compare three tools for feature extraction from symbolic music: jSymbolic [14], music21 [15], and musif. Other tools such as Humdrum [2] may be used for feature extraction, but they would require a larger effort for assembling different features from various toolkits and organizing them in a usable tabular format. We will describe each one in detail in the following subsections.

2.1 jSymbolic

The jSymbolic tool was initially introduced in 2006 [17] and subsequently updated in 2018 [14]. It is an open-source, Java-based software designed to extract features from both MIDI and MEI files. The latest iteration of jSymbolic is capable of extracting 246 distinct features, some of which are multidimensional and account for a total of 1022 values. However, the actual number of extracted features may vary depending on the user's configuration and the musical composition itself. jSymbolic features relate to pitch statistics, melodic intervals, chords and vertical intervals, rhythm, instrumentation, texture, and dynamics. In addition to these features, jSymbolic is capable of computing certain characteristics that are not readily available in MIDI files. To achieve this, jSymbolic utilizes the MEI file format to determine the number of slurs and grace notes in a given piece. While MEI and other high-informative file formats offer additional features such as pitch names, harmonic analysis, and written dynamic or agogic indications, jSymbolic does not take these into consideration.

The jSymbolic software provides users with the flexibility to customize configurations and features, facilitating the integration of previously existing feature values into newer features. Furthermore, users can extract windowed features by specifying window size and overlap in seconds. jSymbolic does not provide pre-built methods for parallel processing of large corpora, thereby requiring the user to implement a suitable strategy. Lastly, jSymbolic provides output options in both CSV and Weka's ARFF format.

The software is accessible as a self-contained program featuring a Graphical User Interface (GUI) and a Command Line Interface (CLI), as well as as a Java library.

2.2 music21

music21 is a Python toolkit designed for computational music analysis, which was first introduced in 2010 [18]. One of its remarkable features is the capability to parse a wide range of file formats, including MIDI, MusicXML, **kern, ABC, and various others. The music information is represented in an object-oriented hierarchical structure that is aimed at facilitating the development of novel tools.

After its initial academic publication, music21 was further developed with a set of features presented in 2011 [15]. The latest version of music21 includes 69 features introduced by jSymbolic, as well as 20 characteristics computed using the information parsed from high-informative file formats. These characteristics are related to key, cadence, harmony, and lyrics. Regardless of the input file format, music21 consistently outputs 633 features. However, the number of extracted features may vary since some features are zeroed out when they are not computable.

music21 is a Python module that lacks a CLI or a GUI. It does not have a configuration format; rather, it offers a broad range of methods for developing custom pipelines for different types of music information processing. These methods encompass the creation of new features and some automated high-level inference of music characteristics, such as key [19], as well as tools for windowed analysis.

One disadvantage of music21 is that large music scores may result in deeply nested Python objects with numerous non-picklable attributes attached. This makes the programming process challenging, particularly due to the difficulty of saving these objects to a file.

In this study, we have developed a CLI for utilizing music21 feature extraction tools in a manner comparable to musif. This implementation facilitates parallel processing by distributing the extraction of features across numerous files simultaneously.

2.3 musif

Our software is named musif [20]. It is implemented in Python and built upon the music21 library, and offers an Application Programming Interface (API) with no default settings of significance and a CLI with default settings optimized for most common use cases.

We leverage music21's internal representation, enabling us to extract features from any file format supported by music21. musif is highly customizable and allows users to add custom features as required. After creating the internal representation of the musical score using music21, we extract multiple features and store them in `pandas` dataframes. This facilitates exporting results in various formats, making musif easily integrable into diverse pipelines.

One limitation of music21 is its restricted ability to serialize complex and large music scores. This restriction also affects the possibility of parallel processing, as Python's single-thread approach necessitates parallelization via processes, which in turn requires context copying and data serialization. Furthermore, parsing large XML files is one of the slowest steps in the feature extraction process. To optimize this procedure, a more favorable strategy would be to store the parsed XML files' logical structure on disk as a cache. We have thus implemented a caching system capable of caching and serializing any music21 object. A restriction to note about the caching system is that the cached scores are read-only. However, this feature enables the writing of parsed scores onto disk and caching of the output from resource-intensive music21 functions into memory.

musif can extract harmony-related features by utilizing standardized harmonic analyses annotated in the Mus-

[2] `https://github.com/humdrum-tools/humdrum-tools`

Table 1. Computational efficiency of the three feature extraction tools. Each run was repeated twice and the second run times are indicated between parentheses.

File format	Tool	Avg CPU Time (s)	Avg Real Time (s)	Avg RAM (GB)	Max RAM (GB)	Tot. errored files	Tot. files
MIDI	*musif*	66.30 (13.30)	5.62 (**1.14**)	9.10 (10.1)	14.2 (19.6)	1	16734
	music21	55.3 (55.2)	4.72 (4.71)	**7.12 (7.12)**	**9.87 (9.94)**	**0**	
	jSymbolic	**2.20 (2.20)**	**1.98** (1.97)	7.97 (7.14)	16.1 (11.7)	14	
MusicXML	*musif*	15.4 (**6.63**)	1.32 (**0.57**)	5.87 (5.12)	12 (10.1)	4	14712
	music21	**10.8** (10.8)	**0.91** (0.91)	**4.30 (4.33)**	**5.68 (5.55)**	**0**	
***kern*	*musif*	26 (**13.1**)	2.26 (**1.14**)	5.20 (4.14)	5.60 (4.92)	**0**	472
	music21	**14.0** (14.1)	**1.21** (1.21)	**3.08 (3.04)**	**4.12 (4.18)**	**0**	

eScore file format [12, 13]. Besides, it encompasses a wide range of features, including melodic intervals, harmony, dynamics, tempo, density and texture, lyrics, instrumentation, scoring, and key. Notably, dynamics and tempo are determined by the composer's text notation rather than by MIDI parameters. Furthermore, our implementation includes all features provided by music21 with the exception of 14 features that utilized the caching system in writing mode. The number of extracted features depends on the complexity of the score and is influenced by both the number of parts and musif's compatibility with the encoding.

NaN values are used to represent non-computable features in a score. For example, when processing datasets with varying instrumentations, some features may not be available for all scores. These values can be replaced with a default value (e.g., 0) or removed from the corpus by deleting either the score or the related feature. In the CLI, we have implemented a heuristic to determine whether a score should be removed from the extracted corpus if it contains too many NaNs. Specifically, we define r as the ratio between the number of columns without NaN and the total number of rows in the output table. If $r < 0.1$, we compute n_i, which is the number of NaNs in the ith row. We remove rows with n_i greater than $\frac{1}{0.99}q_{0.99}$, where $q_{0.99}$ is the 99% quantile of $\{n_1, n_2, \ldots\}$, indicating that 99% of rows are not deleted. The factor $\frac{1}{0.99}$ can be better understood as dividing the $Q_{0.99}$ by 99, thus obtaining an estimate of $Q_{0.01}$, and multiplying it by 100, thus obtaining the expected value of $Q_{1.00}$ based on the first 99% of the data. Put differently, it computes the maximum n_i that we expect if the remaining 1% of rows has a number of NaN "similar" to the previous 99%. Larger values are thus considered outliers. This method was empirically tested on the corpuses used in this work (see Section 3), revealing that only a few scores were generally removed while most lines of the output table were retained. In case a score is not deleted, the CLI removes from the tale the features that are NaN in that score.

musif also incorporates a post-processing module that facilitates the removal, merging, or substitution of values in specific columns or groups of columns within the extracted data. This functionality proves especially advantageous when dealing with large tables generated by musif from a substantial set of scores, as it minimizes the computational effort required for processing such tables.

Like the other tools, we have implemented the capability to extract features at a window level. However, unlike jSymbolic, in our implementation, the window length is specified in musically relevant units such as score measures rather than seconds. This provides more pertinent information for processing music scores.

In contrast to other tools, our solution provides an out-of-the-box capability for processing large corpora through parallel processing, resulting in a reduction of the required time.

The design principles and the features included in musif were presented in a previous publication [20]. The code and documentation of musif is available online [3] .

3. BENCHMARKING METHODOLOGY

To assess the performance of musif in comparison to other tools, we devised a benchmarking methodology. Initially, we identified several datasets that enable testing of diverse file formats. Subsequently, we developed a standardized protocol based on an AutoML pipeline [21]. We evaluated the computational resources utilized by each tool during extraction and their respective efficacy in various classification tasks.

3.1 Datasets

We selected five datasets to evaluate the performance of the tools in analyzing both Standard MIDI Files (SMFs) and highly informative music score formats. For MIDI analysis, we aimed to test both music scores and performances. As for highly informative file formats for music scores, we chose MusicXML and **kern due to their popularity, availability of large datasets, various conversion tools, and compatibility with common music score editing software such as Finale, Sibelius, and MuseScore. While MEI was considered as an option, the limited availability of datasets in this format led us to leave it for future studies.

In this study, we considered the following datasets:

- **ASAP** [22]: This dataset contains music performances derived from the Maestro dataset [23] and is synchronized with a corresponding score obtained from the MuseScore's crowd-sourced online library. The dataset comprises 222 music scores in MusicXML and MIDI formats, as well as 1068 music performances in MIDI format. The authors have rectified any significant notation errors found in the music scores. We used this dataset for composer recognition based on music scores and music performances.
- **EWLD** [24]: It contains lead sheets obtained from Wikifonia, a crowd-sourced archive. To reduce errors in music score transcription by inexperienced users, the authors applied algorithmic selection criteria to the dataset.

[3] `https://github.com/DIDONEproject/musif`, `https://musif.didone.eu`

Table 2. Resulting task size for each dataset and feature set.

Extension	Dataset	Classification task	Samples	Classes	Features				
					musif		music21		jSymbolic
					musif	musif native	music21	music21 native	
MIDI	*ASAP performances*	Composer	211	10	710	91	633	602	225
	ASAP scores	Composer	211	7	710	91	633	602	225
	EWLD	Genre	2645	11	710	91	633	602	225
	JLR	Attribution	109	3	732	113	633	602	226
	Quartets	Composer	363	3	1593	974	633	602	225
	Didone	Decade	1622	8	745	126	633	602	225
MusicXML	*ASAP scores*	Composer	211	7	710	91	633	602	
	EWLD	Genre	3197	11	724	105	633	602	
	JLR	Attribution	109	3	739	120	633	602	
	Didone	Decade	1636	8	971	352	633	602	
***kern*	*Quartets*	Composer	363	3	734	115	633	602	

Specifically, they retained only scores with simple notation, without modulations and with a single melodic part. Moreover, all scores contained key signatures and chords throughout. The dataset was augmented by incorporating genre and composer details, as well as the year of first performance, composer birth and death dates, precise title, and additional metadata. This was achieved by cross-referencing the dataset with information sourced from `secondhandsong.com` and `discogs.com`. We used this dataset for genre recognition.

- **Josquin-La Rue** [25]: This dataset was created within the context of the Josquin Research Project and includes 59 Josquin duos and 49 duos by La Rue. The musical scores underwent a meticulous musicological transcription process. Moreover, the music scores were assigned to two labels based on the security of the attribution, thus resulting in four labels (Josquin secure, La Rue secure, Josquin not secure, La Rue not secure). The musical scores are provided in various file formats including MIDI, MusicXML, **kern, Sibelius, and PDF. We used this dataset for composer classification in a real-world attribution problem.
- **Quartets** [26]: We retrieved a selection of files from the `kern.humdrum.org` website, consisting of all available string quartets in **kern format by Mozart, Haydn, and Beethoven. While the original sources of these musical scores are not always declared, the encoding quality is generally considered to be at a musicological level. In total, we obtained 363 files. We used this dataset for composer classification.
- **Didone** [16]: With the aim of filling an under-studied repertoire, we curated, analyzed, and transcribed over 1600 arias from 18th-century opera, written by dozens of composers. The music scores were transcribed into MusicXML format using Finale Music software and revised by three musicologists independently. Harmonic analyses were added by expert musicologists using MuseScore software in accordance with a prior standard [12, 13] and were reviewed automatically using the ms3 tool [27]. We also included various metadata in the database such as year and place of premiere, composer, and high-level formal analysis. This database is an ongoing project and will be made freely available in 2024. We utilized this dataset for classifying the period of composition of each piece, each period being defined in decades (i.e., 1720s, 1730s, 1740s, etc.).

3.2 Experimental setup

After selecting the datasets, a standardized protocol was developed for benchmarking the three aforementioned tools. The protocol is based on an AutoML pipeline [21] and comprises the following steps:

1. **Conversion to MIDI**: The datasets were selected and subsequently converted into MIDI format, resulting in two or three file formats for each dataset: MIDI and either MusicXML or **kern. This step aims to evaluate the impact of notational file formats, such as MusicXML or **kern, on classification tasks. Indeed, although MIDI has limited capacity for representing notational aspects of music, it remains uncertain the extent to which these aspects can determine the accuracy of machine-learning algorithms for music symbolic analysis. MusicXML files were converted using MuseScore 3, and **kern files were processed with the Humdrum toolkit [4].
2. **Feature extraction**: Features were extracted from MIDI, MusicXML, and **kern files using the methods detailed in Section 2 with default settings and without the use of windows, resulting in one array of features for each file. The purpose of this step was to measure the computational cost of the tools. Therefore, all available files in the datasets were used to obtain a larger number of samples and a more accurate estimation of the computational cost, even if they were discarded in later steps. For instance, MIDI scores were already provided in the ASAP dataset; however, we additionally converted them from the MusicXML files. As a result, we extracted features from more files than necessary. We created a CLI tool in Python for music21 while we utilized the official CLI tools for jSymbolic and musif. Each file format was processed individually, resulting in CSV files for each format. We calculated the average time and RAM usage of each tool. Furthermore, CPU time was collected as a measure of the required time without parallel processing. Lastly, we documented the number of files for which each tool produced errors.
3. **AutoML**: A state-of-the-art machine learning approach was employed using the Python module `auto-sklearn` [21]. The method utilizes Bayesian

[4] See footnote 2 .

Table 3. Accuracies of AutoML using 10-fold cross-validation on the first ten principal components. The best-performing tool is underlined. The best-performing combination is shown in bold.

Extension	Dataset	Dummy guessing	Tools					Combinations			
			musif	musif native	music21	music21 native	jSymbolic	musif native + music21 native	musif native + jSymbolic	music21 native + jSymbolic	musif native + music21 native + jSymbolic
MIDI	ASAP performances	.100	.960	.715	.978	.976	.916	.972	.962	**.980**	.979
	ASAP scores	.146	.743	.644	.781	.751	.780	.791	.819	.819	**.857**
	EWLD	.091	.201	.157	.212	.204	.257	.219	.245	.242	**.259**
	JLR	.344	.700	.642	.779	.751	.722	.711	**.751**	.742	.741
	Quartets	.340	.678	.668	.725	.711	.810	.768	**.831**	.791	.822
	Didone	.125	.359	.362	.403	.380	.443	.414	.451	**.479**	.462
MusicXML	ASAP scores	.171	.773	.669	.759	.745		**.785**			
	EWLD	.091	.216	.185	.215	.201		**.231**			
	JLR	.334	**.793**	.663	.768	.756		**.793**			
	Didone	.126	.398	**.399**	.384	.374		.392			
**kern	Quartets	.340	.713	.711	.767	.763		**.810**			

optimization with surrogate models based on random forests and generates ensembles of models by exploring a vast array of possible architectures. 10-fold cross-validation was used, and the balanced accuracy averaged across the test folds was observed. The best-performing model's result was used for comparison. To initiate the AutoML process, a list of valid files for each dataset was initially defined, discarding those processed in the previous step but unsuitable for validating the classification task. Subsequently, files were selected for which all tools succeeded in extraction, creating comparable datasets for validation. Finally, classes with a number of samples less than twice the number of cross-validation splits were eliminated from each dataset. Consequently, the number of files and categories used in our study differs from the numbers officially provided by each dataset. The classification task performed depended on the dataset, as shown in Table 2.

We conducted two primary experiments: one utilizing all of the extracted features and another using only the first ten principal components. To achieve this, we standardized the features and applied PCA to obtain the ten first principal components. The rationale for the latter experiment is that a larger feature space typically requires a longer AutoML optimization process and affects the performance of the trained classifiers. As the tools extract varying numbers of features, this experiment enables a principled comparison of the usefulness of the non-redundant information generated by the different tools by homogenizing the number of variables in the AutoML process. In other words, it helps decouple the AutoML optimization capabilities from the number of features.

Due to the overlap between the features extracted with musif and those with music21 with jSymbolic, we also analyzed the concatenation of music21, jSymbolic, and our features. We also observed the performance of musif and music21 when only the native features were used, i.e. when musif was utilized without music21 features and when music21 was run without jSymbolic features. In the following, we denote these feature sets as "native". We run each feature extraction and AutoML experiment on a Linux machine with 32 GB of RAM and an i7-8700 CPU, ending the AutoML procedure after 30 minutes. We also experimented with longer AutoML processes and more powerful machines for the first 5 columns of tables 4 and 3, but we noticed no significant change in accuracy.

4. RESULTS

Table 1 summarizes the comparative computational efficiency of the three tools. It is observed that jSymbolic outperforms the other tools when no parallel processing is employed. This can be attributed to the superior performance of Java language, which facilitates faster I/O operations and parsing of byte-level structures such as MIDI files. musif's caching system significantly reduces the time required for feature extraction during multiple runs, such as those performed during the development and debugging of newly added features. For MIDI files, the extraction process can be accelerated by a factor of five. When comparing the time needed for extraction, jSymbolic is still faster than musif. However, our caching system is advantageous when a cache is available. Regarding MusicXML and **kern files, musif and music21 use the same parser engine, making their time values more comparable. In this case, music21 is slightly faster than musif but also attempts to extract a smaller number of features. Nevertheless, musif's the caching system allows for a 50% reduction in extraction times. The music21 tool proves to be the optimal choice when taking into account RAM utilization.

Table 2 presents the dataset sizes used in our experiments, which are obtained through the protocol detailed in Section 3.2. The sample sizes vary from 109 to 3197, while the number of classes ranges from 3 to 11, depending on the dataset. The music21 feature extraction process produces a fixed set of 602 native features, supplemented by an additional 31 features re-implemented from the jSymbolic feature set. In contrast, jSymbolic consistently extracts a set of 225 features with minor variations. musif extracts a variable number of features depending on its ability to parse different music structures, ranging from 91 to 974 extracted features. The remaining features extracted by musif are computed using the music21 feature extraction methods. It is worth noting that music21 always converts non-computable features to zero, whereas musif allows users to assign different values or perform other operations.

Tables 3 and 4 demonstrate the effectiveness of feature sets in representing significant aspects of music analysis across various repertoires. The results in Table 4 must be interpreted with caution due to the longer AutoML process required by accurate models when using a higher number of features. Overall, music21 and jSymbolic are effective tools for extracting features from MIDI files, while musif

Table 4. Accuracies of AutoML using 10-fold cross-validation on all the extracted features. The best-performing tool is underlined. The best-performing combination is shown in bold.

Extension	Dataset	Dummy guessing	Tools					Combinations			
			musif	musif native	music21	music21 native	jSymbolic	musif native + music21 native	musif native + jSymbolic	music21 native + jSymbolic	musif native + music21 native + jSymbolic
MIDI	ASAP performances	.100	.983	.839	.983	.984	.985	.983	.985	**.990**	.988
	ASAP scores	.146	.843	.626	.877	.887	.886	.911	.898	**.912**	.937
	EWLD	.0912	.224	.180	.249	.227	.248	.236	.250	.249	**.251**
	JLR	.344	.746	.697	**.806**	.761	.747	.789	.787	.751	.774
	Quartets	.340	.828	.771	.843	.813	.901	.843	.896	.880	**.904**
	Didone	.125	.480	.429	.525	.508	.586	.515	.572	**.596**	.557
MusicXML	ASAP scores	.171	.830	.710	**.880**	.841		.847			
	EWLD	.091	.251	.200	**.266**	.253		.245			
	JLR	.334	.797	.704	**.815**	.806		.750			
	Didone	.126	.510	.504	.527	.516		**.535**			
**kern	Quartets	.340	.822	.786	.830	.820		**.842**			

Table 5. Accuracies of AutoML. Effect of harmonic features on the Didone dataset.

	Extension	Harmonic features	musif	musif native	musif native + music21 native	musif native + jSymbolic	musif native + music21 native + jSymbolic
First 10 PCs	MIDI	No	.359	.362	.414	.451	.462
		Yes	**.380**	.372	.398	.452	**.465**
	MusicXML	No	.398	.399	.392		
		Yes	.385	**.406**	**.409**		
All features	MIDI	No	**.510**	.504	.515	**.596**	.557
		Yes	.507	.437	.518	.575	.560
	MusicXML	No	.480	.429	.535		
		Yes	**.535**	.521	**.564**		

shows promising results for MusicXML files, particularly when utilizing the first ten principal components during validation. This difference in performance can be attributed to the presence of highly correlated features in musif, a consequence of its granularity. We also evaluated combinations of feature sets and found that optimal performance is achieved by employing multiple tools. For MIDI files, jSymbolic is fundamental in achieving model accuracy, but incorporating musif and music21 generally enhances performance. For MusicXML and **kern files, leveraging both musif and music21 yields optimal results, especially when considering the first ten principal components.

When comparing the efficacy of models trained on MusicXML, **kern, and MIDI files, no discernible pattern emerges indicating the superiority of highly informative file formats over SMFs for representing music scores. In fact, the only instances where the MusicXML files exhibit superior performance are in the Josquin-La Rue dataset and genre recognition on the EWLD dataset when all features are utilized. However, for all the remaining tasks, MIDI files demonstrate superior performance. This is likely due to the fact that jSymbolic can only extract features from MIDI files and is simultaneously the most important source of features for music score analysis. Consequently, in this study, the MusicXML and **kern datasets lack some relevant features that can be extracted only when converted to MIDI. Even when comparing only the proposed tool and music21's performances, MusicXML and **kern files do not show a clear advantage over MIDI files, particularly when considering the combination of both tools. It should be noted that jSymbolic can extract features from MEI as well, thus potentially allowing for better performances.

The effect of missing values on tool performance is a significant concern and may be a contributing factor to the comparatively lower results for MusicXML and **kern files. While music21 substitutes all missing values with 0, musif utilizes a hybrid strategy that entails either removing a row or column from the table (refer to Section 2). The most effective method for handling missing values remains an open issue.

We assessed the impact of harmonic features on the Didone dataset using musif. Unfortunately, due to the time-consuming nature of harmonic annotations, we were unable to evaluate these features on the other datasets used in this study. We annotated our dataset of more than 1600 opera arias using the standard established in previous works (see Section 2) and extracted melody- and accompaniment-related features with respect to the local key. The extraction of harmonic features resulted in 22 additional features beyond the 126 listed in Table 2 for MIDI files. For MusicXML files, we extracted 265 additional features, raising the total number of extracted features to 617. We observed an overall improvement in classification accuracy when incorporating harmonic features, as demonstrated in Table 5. The only instance where performance was degraded by the inclusion of harmonic features was for MIDI files when all the available features were considered (without PCA). We interpret this degradation as an indication that longer processing times are necessary for AutoML when additional, possibly highly correlated features are introduced.

5. CONCLUSION

This paper presents a comprehensive analysis of tools for extracting features from symbolic music. A strict protocol was defined to compare the tools in terms of efficiency and efficacy across various repertoires and file formats. The results indicate that using multiple tools is the most effective approach, with the optimal tool choice depending on the file format and repertoire.

The study emphasizes the importance of using file formats that are accessible by multiple tools. However, it remains open whether highly informative file formats such as MusicXML, **kern, or MEI are relevant for the automatic classification of symbolic scores. The available set of features indicates that, while these formats remain fundamen-

tal for certain types of musicological research, they do not seem to entail a significant advantage for machine learning tasks.

The problem of NaN values in extracted features from music scores remains unresolved. Further research is required to explore optimal approaches for replacing, removing, or inferring missing values in music applications.

Additionally, the new musif tool was proposed, which can process various file formats using the music21 parsing engine. The tool also includes a caching mechanism to speed up feature development. Moreover, motivated by the experiments presented in this work, we included the whole music21 and jSymbolic tools in the newer versions of musif, easing the extraction of the combined feature sets from large corpora.

6. ACKNOWLEDGEMENTS

This publication is a result of the Didone Project, which has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program, Grant agreement No. 788986. It has also been conducted with funding from Spain's Ministry of Science and Innovation (IJC2020-043969-I/AEI/10.13039/501100011033).

Part of the computational experiments were run at the FinisTerrae III cluster of the Galician Supercomputing Center (CESGA). The authors gratefully acknowledge the access to these resources.

7. REFERENCES

[1] K. C. Kempfert and S. W. K. Wong, "Where Does Haydn End and Mozart Begin? Composer Classification of String Quartets," *Journal of New Music Research*, vol. 49, no. 5, pp. 457–476, Oct. 2020.

[2] W. Herlands, R. Der, Y. Greenberg, and S. Levin, "A Machine Learning Approach to Musically Meaningful Homogeneous Style Classification," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 28, no. 1, Jun. 2014.

[3] J. Qiu, C. L. P. Chen, and T. Zhang, "A Novel Multi-Task Learning Method for Symbolic Music Emotion Recognition," 2022.

[4] M. Zeng, X. Tan, R. Wang, Z. Ju, T. Qin, and T.-Y. Liu, "MusicBERT: Symbolic Music Understanding with Large-Scale Pre-Training," *FINDINGS*, 2021.

[5] D. Jeong and J. Nam, "Note Intensity Estimation of Piano Recordings by Score-informed NMF," 2017, p. 8.

[6] F. Simonetta, S. Ntalampiras, and F. Avanzini, "Acoustics-Specific Piano Velocity Estimation," in *Proceedings of the IEEE MMSP 2022*, 2022.

[7] H. Vinet, "The Representation Levels of Music Information," in *Computer Music Modeling and Retrieval*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 193–209.

[8] F. Simonetta, S. Ntalampiras, and F. Avanzini, "Multimodal Music Information Processing and Retrieval: Survey and Future Challenges," in *Proceedings of 2019 International Workshop on Multilayer Music Representation and Processing*. Milan, Italy: IEEE Conference Publishing Services, 2019, pp. 10–18.

[9] ——, "Audio-to-Score Alignment Using Deep Automatic Music Transcription," in *Proceeddings of the IEEE MMSP 2021*, 2021.

[10] A. Llorens and A. Torrente, "Constructing *opera seria* in the Iberian Courts: Metastasian Repertoire for Spain and Portugal," *Anuario Musical*, vol. 76, pp. 73–110, Jul. 2021.

[11] F. Moss, W. Fernandes de Souza, and M. Rohrmeier, "Harmony and Form in Brazilian Choro: A Corpus-Driven Approach to Musical Style Analysis," *Journal of New Music Research*, vol. 49, pp. 416–437, 2020.

[12] 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, 2018.

[13] 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, May 2021.

[14] 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*, 2018, pp. 348–354.

[15] M. S. Cuthbert, C. Ariza, and L. Friedland, "Feature Extraction and Machine Learning on Symbolic Music Using the Music21 Toolkit," 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. 387–392.

[16] A. Torrente and A. Llorens, "The Musicology Lab: Teamwork and the Musicological Toolbox," in *Music Encoding Conference Proceedings 2021, 19–22 July, 2021 University of Alicante (Spain): Onsite & Online, 2022, ISBN 978-84-1302-173-7, págs. 9-20*. Universidad de Alicante / Universitat d'Alacant, 2022, pp. 9–20. [Online]. Available: https://dialnet.unirioja.es/servlet/articulo?codigo=8463477

[17] C. McKay and I. Fujinaga, "jSymbolic: A Feature Extractor for MIDI Files," in *Proceedings of the 2006 International Computer Music Conference, ICMC 2006, New Orleans, Louisiana, USA, November 6-11, 2006*. Michigan Publishing, 2006. [Online]. Available: https://hdl.handle.net/2027/spo.bbp2372.2006.063

[18] M. S. Cuthbert and C. Ariza, "Music21: A Toolkit for Computer-Aided Musicology and Symbolic Music Data," *International Society for Music Information Retrieval Conference (ISMIR 2010)*, pp. 637–642, 2010. [Online]. Available: http://ismir2010.ismir.net/proceedings/ismir2010-108.pdf

[19] C. L. Krumhansl, *Cognitive Foundations of Musical Pitch*. Oxford University Press, 1990.

[20] A. Llorens, F. Simonetta, M. Serrano, and Á. Torrente, "Musif: A Python Package for Symbolic Music Feature Extraction," in *Proceedings of SMC 2023 - Sound and Music Computing Conference*, Stockholm, 2023, June, pp. 132–138.

[21] M. Feurer, K. Eggensperger, S. Falkner, M. Lindauer, and F. Hutter, "Auto-Sklearn 2.0: Hands-free AutoML via Meta-Learning," arXiv, Tech. Rep. arXiv:2007.04074, Sep. 2021. [Online]. Available: http://arxiv.org/abs/2007.04074

[22] 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 21st International Society for Music Information Retrieval*, 2020, Proceedings.

[23] C. Hawthorne, A. Stasyuk, A. Roberts, I. Simon, C.-Z. 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*, 2019.

[24] 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 - AM'18*. ACM Press, 2018.

[25] J. Cumming, C. McKay, J. Stuchbery, and I. Fujinaga, "Methodologies for Creating Symbolic Corpora of Western Music Before 1600," in *Proceedings of the ISMIR*. Paris, France: ISMIR, Sep. 2018, pp. 491–498.

[26] C. S. Sapp, "Online Database of Scores in the Humdrum File Format," in *Proceedings of the 6th International Conference on Music Information Retrieval*, 2005, p. 2.

[27] J. Hentschel and M. Rohrmeier, "Creating and Evaluating an Annotated Corpus Using the Library Ms3," 2020.

EXPLORING SAMPLING TECHNIQUES FOR GENERATING MELODIES WITH A TRANSFORMER LANGUAGE MODEL

Mathias Rose Bjare[1] **Stefan Lattner**[2] **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, stefan.lattner@sony.com, gerhard.widmer@jku.at

ABSTRACT

Research in natural language processing has demonstrated that the quality of generations from trained autoregressive language models is significantly influenced by the used sampling strategy. In this study, we investigate the impact of different sampling techniques on musical qualities such as diversity and structure. To accomplish this, we train a high-capacity transformer model on a vast collection of highly-structured Irish folk melodies and analyze the musical qualities of the samples generated using distribution truncation sampling techniques. Specifically, we use nucleus sampling, the recently proposed "typical sampling", and conventional ancestral sampling. We evaluate the effect of these sampling strategies in two scenarios: optimal circumstances with a well-calibrated model and suboptimal circumstances where we systematically degrade the model's performance. We assess the generated samples using objective and subjective evaluations. We discover that probability truncation techniques may restrict diversity and structural patterns in optimal circumstances, but may also produce more musical samples in suboptimal circumstances.

1. INTRODUCTION

In recent years, developments in natural language modelling have also accelerated the field of symbolic music generation. In this context, the musical events of a music piece are represented as a sequence of symbols or tokens from a fixed vocabulary, and the goal is to learn to generate new token sequences. At present, the autoregressive transformer model [1] is the basis of many symbolic music generation models [2–5]. In this context, a conditional distribution is learned by solving a masked self-prediction task [2–5], and generation is performed with stochastic sampling techniques, e.g., ancestral sampling, or maximization-based search techniques, e.g., beam search.

© M. Bjare, S. Lattner, and G. Widmer. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).
Attribution: M. Bjare, S. Lattner, and G. Widmer, "Exploring Sampling Techniques for Generating Melodies with a Transformer Language Model", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

However, the choice of decoding technique has been shown to impact various qualitative features of generated samples substantially. In [6], the authors showed that generation with *nucleus sampling* yields natural language samples that are more contextualized than those from conventional sampling techniques, and samples of nucleus sampling score higher in human evaluations. More recently, the authors of [7] propose *typical sampling* and show that it reduces degenerate sample generation while exhibiting performance competitive with nucleus sampling. Typical sampling is based on the authors' finding that words in human language are *typical*. More specifically, the authors show that most words of human language are, in fact, not the most likely words (lowest information content (IC)), as measured with a language model, but rather *typical* words, i.e., they have an IC close to the conditional entropy of the language model. Typical sampling explicitly enforces this condition.

We hypothesize that a careful choice of sampling technique could also improve certain aspects of music generated using language models, particularly because in [8], it has been shown that musical events tend to be typical. However, we find that many music generation systems rely on ordinary sampling techniques. In addition, studies on the effect of sampling techniques on musical qualities are limited.

In this work, we study the structural and tonal properties of music generated with different sampling techniques applied to a high-capacity transformer model. Specifically, we measure the IC, long and short-term self-similarities and scale consistency of samples generated with conventional sampling, nucleus sampling, and typical sampling. We test the sampling techniques for a well-calibrated model and for under-calibrated models. We support our findings by performing a listening study. We conduct our experiments on *The Session* dataset [9], a large dataset of well-structured monophonic music in the established musical genre of Irish traditional music. We choose this dataset since we expect it to provide suitable conditions for training a well-calibrated model. Our findings suggest that truncation techniques can address inadequacies of models that are not well-fitted to the data.

2. BACKGROUND AND RELATED WORK

Although maximization-based techniques like beam search work well for directed language generation tasks [1] (such as machine translation and summarization), beam search has been shown to produce dull and repetitive samples for open-ended language generation tasks [2] [6], an effect that can be observed in music generation as well [10]. It is, therefore, more common to use stochastic sampling techniques [3] for open-ended generation tasks. The most obvious method is ancestral sampling, where one token at a time is sampled based on the predicted distribution, conditioned on the previously generated tokens. However, it has been shown that truncating the conditional distribution (by setting the probability of specific tokens to zero, followed by renormalising), can lead to better sample quality than the non-truncated variant. An example of distribution truncation is top-k sampling, where all but the k most probable tokens are zeroed. In [12], the authors showed that top-k sampling generates more coherent samples than the non-truncated variant. In [6], it is explained that the quality improvement of top-k sampling is caused by removing unreliably estimated low-probability tokens, and it is found that top-k sampling mitigates the problem. However, it is also shown that top-k sampling is sensitive to the distribution's entropy (see Section 3.3), making it hard to select a value of k that fits both high and low certainty conditions. As a solution, they propose *nucleus sampling* that assigns zero probability to the largest set of least probable tokens that together have a probability below a given threshold. The authors find that the samples produced using the technique are preferred by humans over other sampling techniques. Nucleus sampling has been used in music generation in [13–15], but its effects are difficult to quantify without comparisons to the non-truncated case. Although nucleus sampling mitigates the problem of poorly estimated low-probability tokens, it does not prevent generating degenerated repetitive sequences caused by low entropy distributions (see Section 3). As a solution, in [7], the authors propose *typical sampling* and show that this technique prevents degenerated sample generation.

3. ANCESTRAL SAMPLING

Let $p\left(x_t|x_{<t}\right)$ be the conditional probability of a symbol x_t given previously observed symbols $x_{<t}$ (i.e., the context) and let q be a model fitted to p, e.g., a neural network fitted via likelihood maximization. Given a model q, ancestral sampling samples one token at a time using $x_0 \sim q(\cdot), x_1 \sim q(\cdot|x_0), ..., x_t \sim q(\cdot|x_{<t})$.

[1] Generation with input sequence conditioning.

[2] Generation without input sequence conditioning.

[3] In the context of generative models, "*sampling techniques*" could refer to a multitude of aspects in the generative pipeline (e.g., Gibbs sampling in restricted Boltzmann machines [11]). In our work, "*sampling techniques*", refers to techniques for obtaining samples from a trained language model.

3.1 Distribution truncation sampling techniques

In distribution truncation, a truncated distribution $\widetilde{q}$ is obtained by zeroing the probability of a subset of tokens and renormalising the resulting distribution. Formally, $\widetilde{q}$ is defined by

$$\widetilde{q}(x_t|x_{<t}) = \begin{cases} \frac{q(x_t|x_{<t})}{\sum_{v\in V} q(v|x_{<t})} & \text{if } x_t \in V \\ 0 & \text{otherwise} \end{cases}, \tag{1}$$

where V is the set of tokens with nonzero probability in $\widetilde{q}$. For the remainder of this article, we use '*conventional sampling*' to denote sampling from untruncated distributions.

3.2 Nucleus sampling

In nucleus sampling, V is defined as the smallest set such that

$$\sum_{v\in V} q(v|x_{<t}) \geq \tau, \tag{2}$$

where τ is a constant determining the number of tokens to be removed.

3.3 Typical sampling

In typical sampling [7], V is defined in terms of the token information content described below.

Definition 3.1 (Conditional information content)**.** The conditional *information content* (IC) is given by

$$IC\left(x_t|x_{<t}\right) = -\log q\left(x_t|x_{<t}\right). \tag{3}$$

In computational music perception, IC has been used to model how surprising a musical event is given the musical context [16–18].

Definition 3.2 (Conditional entropy)**.** The conditional entropy is the expected conditional information content

$$H\left(x_t|x_{<t}\right) = \mathbb{E}_{x_t\sim q(\cdot|x_{<t})}\left[IC\left(x_t|x_{<t}\right)\right]. \tag{4}$$

The entropy of a distribution explains how confident a model is. It ranges from 0 to $\log n$ where n is the number of symbols in the vocabulary, with 0 indicating that the distribution is deterministic and $\log n$ indicating that the distribution is uniformly random. In typical sampling, the probabilities of tokens with the highest deviation of information from the entropy

$$\left|H\left(x_t|x_{<t}\right) - IC\left(x_t|x_{<t}\right)\right| \tag{5}$$

are set to zero. More precisely, let $U = v_1, v_2, ..., v_n$ be an ascending ordering of the vocabulary in accordance to Equation (5). Then V is defined as the smallest prefix of U such that $q(U|x_{<t}) \geq \tau$. Equation (5) implies that V is restricted by a band around the entropy as shown in Figure 1. Therefore, also the most likely token under q can have zero probability in $\tilde{q}$. The authors of [7] note that this property, however, lowers the number of degenerately repetitive samples, as opposed to nucleus sampling, without degrading preference in human evaluations.

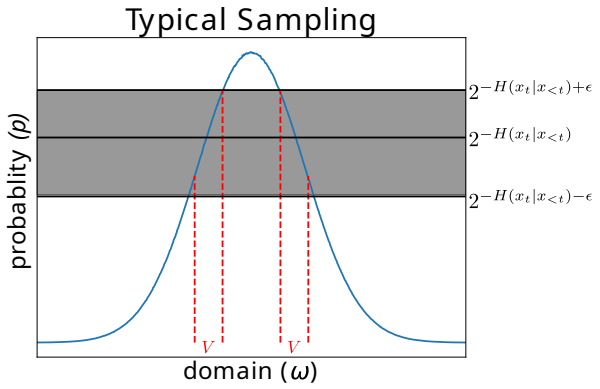


Figure 1: In typical sampling, a probability band around the entropy (dark-grey) defines the set V of tokens with non-zero probabilities in the truncated distribution.

4. EXPERIMENTS

In this section, we describe the setup for both our objective and subjective experiments, the used data, training details, degradation scenarios, and generation details, as well as objective and subjective evaluation.

4.1 Data

Our experiments are performed on monophonic symbolic music. Specifically, we use the midi-encoded version of the *The Session* dataset [9], consisting of $45,849$ traditional Irish folk tunes originally encoded in ABC notation. We discard the 5% longest sequences to lower the computational footprint of the autoregressive transformer model, and partition the dataset in training, validation, and test sets with proportions 10/12, 1/12, and 1/12, respectively. All analyses will be performed on the test set, while our generative models will be trained and optimized on the training and validation sets, respectively. The dataset contains tunes with the same name, corresponding to different versions of the same tune. We ensure that tunes with the same name appear in exactly one of the three sets.

We tokenize the sequences using a modified version of the popular REMI representation [3]. REMI serializes a score bar-wise from left to right. A bar is serialized as a sequence of tokens starting with a bar-delimiter token followed by a serialization of the notes within that bar. Each note is serialized as three tokens indicating the onset within the bar, the pitch and the duration, in that order. The position and duration tokens are quantized to 1/12th of a beat. Contrary to the original REMI implementation, we omit velocity, tempo change and chord symbols, since these are not encoded in the original ABC files either. Similar to [4], we extend the REMI representation with time-signature tokens inserted immediately after the bar token. We base our tokenization implementation on a modified version of the REMI python implementation in MidiTok [19].

4.2 Training

We train a 21-layered Transformer decoder model [20] with relative attention [2, 21] in a self-supervised prediction task. We train the model using Adam optimization [22] with a learning rate of 10^{-4} until no improvement takes place on the validation set during 10 subsequent epochs. The used batch size is 16, and the input sequence length is 512 tokens. Sequences shorter than 512 tokens are zero-padded. The negative log-likelihood (NLL) on the test dataset is measured to be $NLL = 0.30$, which is similar to the result of a recent transformer-based model trained on the same dataset [18]. We thus call this a *well-calibrated model*.

4.3 Model Degradation

In addition to the well-calibrated model, we consider two *under-calibrated models*, which we achieve by intentionally degrading the well-calibrated model. For our first degradation, we scale the logits vector h of the transformer softmax output distribution, i.e.,

$$q(x_t|x_{<t}) = \text{Softmax}(h/r), \tag{6}$$

where $r > 1.0$ is a temperature scale. This degradation increases the distribution's entropy (uncertainty) while keeping the relative ordering of the probabilities the same. Using temperature scaling, we deliberately increase the probability of token predictions x_t that fit the token context $x_{<t}$ poorly, thereby simulating the failure case of unreliably estimated tokens reported for conventional sampling (see section 2), where truncation techniques are expected to provide better results. We empirically set r to the minimal value that leads to an audible degradation of the generated sequences. This resulted in $r = 1.5$. The NLL of the test data under the temperature-degraded model is measured to be $NLL = 0.31$, which is an increase of 0.01 compared to the well-calibrated model.

Secondly, we consider an unbiased degradation where we perturb the network weights by adding a small amount of Gaussian noise. More specifically, for every weight matrix W of the well-calibrated model, we obtain a degraded weight matrix W' by adding noise z_W to W

$$W' = W + kz_W, \tag{7}$$

where $z_W \sim \mathcal{N}(0, \text{std}(W))$ and k is a constant. We sample the noise vector once and keep it fixed for all our experiments. We empirically set k to be the minimal value where sample degradations are audible, which results in $k = 0.175$. The NLL of the test data under the resulting model is measured as $NLL = 0.36$, which is an increase of 0.06.

4.4 Generation

When generating sequences with the learned models, for all models, we perform conventional sampling, nucleus sampling and typical sampling as described in section 3.

We sample until either the end-of-sequence token is encountered or a maximum length is reached. Due to computing limitations, we fix the maximum sequence length to the 80%-quantile of the dataset song-length distribution. We keep both sequences which terminate with the end-of-sequence token and sequences with the maximum length reached in our sample sets.

4.5 Objective Evaluation

The objective evaluations are performed by calculating different statistics from the generated sequences and comparing the results between different (non-)degradations, sampling types and with the original reference data.

4.5.1 Surprisal

We are interested in the degree of surprisal of the samples generated with the different sampling methods. Similar to [16–18], we measure surprisal using the IC of events. As we do not have access to the data distribution, we interpret the well-calibrated model to be an oracle that approximates the data distribution. We then use the well-calibrated model to measure the mean IC of all events from a specific sampling method and model.

4.5.2 Structural Consistency

We measure structural consistency by investigating the self-similarities of the generated pieces. Similar to [5], we compute a self-similarity distribution from samples of a given sampling method and contrast it with the similarity distribution calculated from real data. To do so, we first compute the similarity between bar pairs separated by measure lags of size t. This is done for each tune x in sample sets D according to

$$l^x_{i,i+t} = \frac{|N(i) \cap N(i+t)|}{|N(i) \cup N(i+t)|}, \tag{8}$$

where the set of notes in the i-th bar is denoted as $N(i)$, and two notes are deemed equal if their pitches, durations, and onset positions within their respective bars are identical. The similarity score $l^x_{i,j}$ between any two bars ranges from 0.0 to 1.0, with a score of 1.0 indicating that the two bars are identical. After computing the similarity for all possible lags in each tune of a sample set D, we calculate the average similarity scores of that sample set by

$$L^D_t = \frac{1}{|D|}\left(\sum_{x \in D}\sum_{j=i+t} l^x_{i,j}\right). \tag{9}$$

Note that eq. (9) does not define a probability distribution and does not, in general, sum to one. For each dataset, we then calculate an overall self-similarity score

$$SS(D) = \frac{1}{T}\sum_{t=1}^{T} L^D_t, \tag{10}$$

where T is the maximum bar lag considered. $SS(D)$ captures both short-term self-similarities, e.g., repetitions or variations of motives, and long-term self-similarities, e.g., repetitions or variations of musical segments. Similar to [5], we also consider the deviation of a sample set's similarity distribution L^D_t to the dataset's similarity distribution L_t given by

$$SE(D) = \frac{1}{T}\sum_{t=1}^{T} \left|L_t - L^D_t\right|. \tag{11}$$

We interpret this deviation as a measure of how closely the self-similarities of tunes generated with the different sampling techniques follow the self-similarities of tunes found in the dataset. We set $T = 38$ in our experiments (i.e., the smallest maximum number of bars generated by any method).

4.5.3 Tonal Consistency

We are furthermore interested in the tonality coherence of samples generated with the sampling methods. Specifically, we investigate the scale consistency [23], i.e., the maximum percentage of notes fitting a diatonic scale. The scale consistency is therefore calculated by

$$\max_{scale} \frac{\#pitch\_in\_scale(x, scale)}{\#pitches(x)}. \tag{12}$$

A scale consistency value of 1.0 indicates that all pitches are within a single scale, whereas lower values indicate more complex harmonic structures.

4.6 User Study

In addition to the objective evaluations described above, we also perform a user study to gather subjective evaluations of the tunes' musical quality, structural properties and complexity. For that, we hosted a website consisting of two pages. The first page explains the purpose of the study, specifically that it aims to evaluate sampling techniques for neural network music generation. Furthermore, the users are instructed to rate the respective tunes using the attributes *overall quality*, *short-term structure*, *long-term structure* and *complexity* using a 5-point Likert scale. The users are also asked to use appropriate headphones or loudspeakers and to announce their level of musical expertise with choices {*Beginner*, *Intermediate*, *Expert*}. On the second page, a list of 10 audio widgets is displayed, one for each tune. Below each widget, the Likert scales for the 4 different attributes (as described above) are provided for voting. In addition, the users can click on a "sheet link" that opens a window displaying the tune in staff notation. The 10 tunes for every user constitute the Cartesian product of all three sampling methods (i.e., *conventional*, *nucleus*, *typical*) and all three model modes (i.e., *well-calibrated*, *temperature degradation*, *noise-degradation*) plus a reference tune. It is ensured that every user obtains unique tunes sampled randomly from a set of 500 instances for each of the 10 types, presented in a random order. To prevent biases, every user is allowed to perform the study only once.

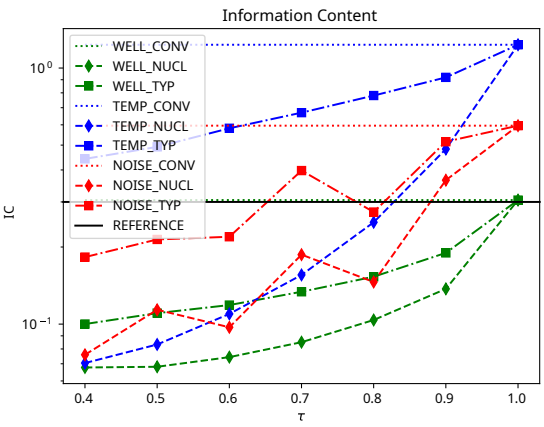


Figure 2: Information content of generated data using different sampling strategies and τ values under the well-calibrated model.

5. RESULTS AND DISCUSSION

In this section, we present the results of the experiments described in Section 4. For the figures and tables, we use the abbreviations WELL, NOISE and TEMP for the well-calibrated, noise-degraded and temperature-degraded models, respectively. To these abbreviations, we append CONV, NUCL and TYP for conventional sampling, nucleus sampling and typical sampling correspondingly.

5.1 Objective Evaluation

In the following section, we analyse and discuss the results of our objective and subjective evaluations.

5.1.1 Surprisal

We report the results of the IC estimation in Figure 2 for the truncation degrees $\tau = 0.4, ..., 1.0$. The samples from the well-calibrated model have the lowest IC and the IC of samples from the temperature-degraded model is higher than the IC of samples from the noise-degraded model. For both nucleus and typical sampling, the IC decreases with decreasing τ. For typical sampling in particular, this suggests that relatively more high information than low information tokens are pruned, similar to what is found in [8]. For most degradation scenarios and sampling methods, a τ value between 0.8 and 0.9 is shown to recover the original data distribution best.

5.1.2 Structural Consistency

We compute the self-similarity (see Equation (10)) for all models and sampling techniques and show the result in Figure 3a. Similarly, we plot the self-similarity deviation (see Equation (11)) in Figure 3b. From Figure 3a, we find that the overall self-similarity of samples produced with typical and nucleus sampling increases as τ decreases. This holds for both degraded models and the well-calibrated model. However, we find that the increase in self-similarity is more moderate for samples generated with typical sampling than those of nucleus sampling, indicating that the removal of highly probable tokens keeps the self-similarity at more moderate levels. In the temperature degradation scenario, we find that moderate levels of truncation lower the self-similarity deviation for the temperature-degraded model and thereby counteract the temperature degradation (with an optimal τ of 0.8 and 0.6 for nucleus sampling and typical sampling, respectively). In fact, in this scenario, the self-similarity of samples generated with nucleus and typical sampling follows the self-similarity of the reference distribution closer than samples generated with ordinary sampling for most tested truncation strengths. This is not the case for the unbiased noise degradation, where the self-similarity increases with higher truncation strengths, increasing also the deviation from the reference statistics.

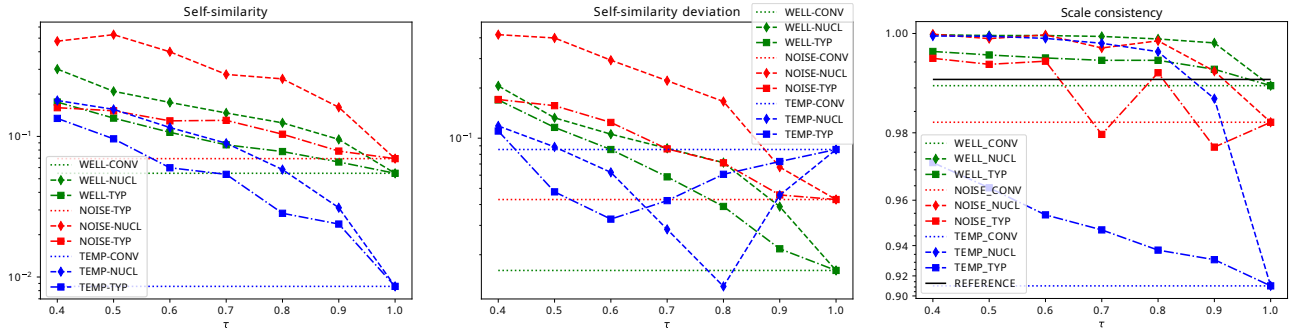
5.1.3 Tonal Consistency

We inspect the tonal consistency by calculating the scale consistency (see Equation (12)) and report the results in Figure 3c. For both nucleus and typical sampling, we find that samples generated with low values of τ lead to a higher degree of scale consistency. Furthermore, we find for any given τ that generations from typical sampling have lower scale consistency than samples generated with nucleus sampling. Especially when considering temperature degradation, the scale consistency of nucleus sampling is almost at the level of the reference distribution at $\tau = 0.9$, whereas typical sampling stays low even at high levels of τ. An important observation is that (with the exception of typical sampling in the temperature degradation scenario) there is an optimal τ for both truncation techniques that leads to a recovery of the dataset's scale consistency statistic in both degradation scenarios.

Similar to the findings in [6] for natural language, our objective evaluations in the high-temperature scenario indicate that the musical statistics of the samples generated with truncation techniques more closely match the statistics of samples from the reference distribution. This finding implies that truncation sampling techniques can be applied to music generative language models, similar to their application in natural language. This can help remove tokens with unreliable probability estimates that do not fit the musical context well. This approach may have implications for more complex datasets and limited resources, where obtaining a well-calibrated model can be challenging.

5.2 User Study

The user study was performed by 38 participants who, according to their self-assessment can be divided into 8 beginners, 18 intermediate and 12 musical experts. The presented melodies (except the *reference*) are generated as described in Section 4.4, with $\tau = 0.8$ for both, nucleus and typical sampling. Table 1 shows the user study results. As there is a high variance for all ratings, we performed for all attributes a Welch's t-test between all $m = 10$ tune types. Using a desired significance level of $\alpha = 0.05$, the corresponding Bonferroni correction to the



(a) (b) (c)

Figure 3: Structural and tonal consistency for different model degradations, sampling strategies and τ values. In (a) the self-similarity of sample sets generated with different sampling techniques is shown. Higher values indicate a higher degree of self-similarities. In (b) the deviation of the generated samples' self-similarities to the self-similarity and the data reference distribution is shown. A deviation of 0 indicates that the self-similarity of a sample set fits the reference distribution exactly. In (c) the scale consistency of different sample strategies and the reference dataset is shown.

Method	QULT	ST_STR	LT_STR	CPLX
REFERENCE	**3.7**±1.0	3.8±1.0	**3.7**±1.1	3.6±0.8
WELL_CONV	3.2±1.1	3.7±0.9	3.5±1.2	3.3±1.0
WELL_NUCL	3.6±1.1	**3.9**±1.1	**3.7**±1.1	2.8±1.0
WELL_TYP	3.4±1.2	3.6±0.9	**3.7**±1.0	3.3±1.0
NOISE_CONV	2.7±1.0	3.2±0.9	3.0±1.0	2.8±0.9
NOISE_NUCL	2.6±1.3	3.2±1.4	2.8±1.5	2.5±1.2
NOISE_TYP	2.7±1.1	3.2±1.1	3.1±1.2	2.4±1.0
TEMP_CONV	2.1±1.3	2.7±1.1	2.1±1.1	**3.7**±1.0
TEMP_NUCL	3.4±1.2	3.6±0.9	3.4±1.3	3.4±1.1
TEMP_TYP	2.2±1.1	2.7±0.9	2.4±1.0	3.3±0.8

Table 1: Results showing the mean-opinion scores of the user study $\pm$ the standard deviation. QULT denotes the overall quality estimation, ST_STR the perceived short-term structure, LT_STR the perceived long-term structure and CPLX the perceived complexity of the rated samples.

multiple comparisons problem gives a significance level of $\frac{\alpha}{\frac{1}{2}m(m-1)} = \frac{0.05}{45} = 0.001$. We can see in the first column that the human-composed reference tracks have the highest quality scores on average and that the perceived quality of the tunes tends to degrade for the noise- and temperature degradation cases as expected. The t-test shows that the users' preference for REFERENCE is significant compared to all samples of the under-calibrated models (with $p < 1 \times 10^{-4}$), except for TEMP_NUCL with $p = 0.37$. This shows that nucleus sampling can potentially improve the sample quality of low-confidence models, while typical sampling is not able to recover any degradations. Furthermore, we find that WELL_CONV, WELL_NUCL and WELL_TYP differ in QULT with $p = 0.07, 0.67$ and 0.37 respectively compared to REFERENCE. This provides some evidence that nucleus and typical sampling improves the sampling quality of well-calibrated models, but this effect is not significant. While nucleus sampling performs well in the temperature-degraded model, we observe some (non-significant) evidence of a lower complexity than conventional and typical sampling in the well-calibrated model (with $p = 0.023$ and $p = 0.044$, respectively). Typical sampling (with $\tau = 0.8$) does not cause significant differences from conventional sampling. As the p-value between NOISE_TYP and NOISE_CONV is also low (but not significant, with $p = 0.06$), there is some evidence that typical sampling slightly reduces the complexity of outputs from under-calibrated models. This could be explained by typical sampling pruning the higher and lower probability events, overall reducing the possible number of events to be sampled. The well-calibrated model performs well with all sampling techniques (no significant differences to REFERENCE), with only some non-significant evidence for lower complexity with nucleus sampling.

6. CONCLUSION

We investigated the effect of distribution truncation sampling techniques on the musical qualities of information content, self-similarity, scale consistency and complexity of samples generated under different degradation scenarios. Our objective evaluations show that a higher truncation strength leads to increased self-similarity and tonal consistency. This trend is more pronounced for samples generated with nucleus sampling compared to samples generated with typical sampling. For a well-calibrated model, we show that the increase in self-similarity and scale consistency leads to an increase in deviations of these metrics from the reference distribution. However, for under-calibrated models, we showed that the deviations from the original data statistics could often be reduced with the correct truncation strategy and carefully selected truncation levels (where a τ between 0.8 and 0.9 seems to be good trade-off value over all experiments). While nucleus sampling carries the risk to reduce complexity of the outputs, this trend could not be observed with typical sampling.

7. 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?"*).

8. REFERENCES

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *NIPS*, 2017, pp. 5998–6008.

[2] C. 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," *CoRR*, vol. abs/1809.04281, 2018.

[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] 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.

[5] B. Yu, P. Lu, R. Wang, W. Hu, X. Tan, W. Ye, S. Zhang, T. Qin, and T. Liu, "Museformer: Transformer with fine- and coarse-grained attention for music generation," in *NeurIPS*, 2022.

[6] A. Holtzman, J. Buys, L. Du, M. Forbes, and Y. Choi, "The curious case of neural text degeneration," in *ICLR*. OpenReview.net, 2020.

[7] C. Meister, T. Pimentel, G. Wiher, and R. Cotterell, "Typical decoding for natural language generation," *arXiv preprint arXiv:2202.00666*, 2022.

[8] M. R. Bjare and S. Lattner, "On the typicality of musical sequences," in *ISMIR (Late-breaking demo)*, 2022.

[9] B. L. Sturm, J. F. Santos, O. Ben-Tal, and I. Korshunova, "Music transcription modelling and composition using deep learning," in *Proc. Conf. Computer Simulation of Musical Creativity*, Huddersfield, UK, 2016.

[10] S. Dieleman, "Musings on typicality," 2020. [Online]. Available: https://benanne.github.io/2020/09/01/typicality.html

[11] R. Salakhutdinov, A. Mnih, and G. Hinton, "Restricted boltzmann machines for collaborative filtering," in *Proceedings of the 24th international conference on Machine learning*, 2007, pp. 791–798.

[12] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," 2019.

[13] W. Hsiao, J. Liu, Y. Yeh, and Y. Yang, "Compound word transformer: Learning to compose full-song music over dynamic directed hypergraphs," in *AAAI*. AAAI Press, 2021, pp. 178–186.

[14] B. Sturm and L. Casini, "Tradformer: A transformer model of traditional music," in *International Joint Conference on Artificial Intelligence*, 2022.

[15] F. Mo, X. Ji, H. Qian, and Y. Xu, "A user-customized automatic music composition system," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 640–645.

[16] L. B. Meyer, "Meaning in music and information theory," *The Journal of Aesthetics and Art Criticism*, vol. 15, no. 4, pp. 412–424, 1957. [Online]. Available: http://www.jstor.org/stable/427154

[17] M. Pearce, "The construction and evaluation of statistical models of melodic structure in music perception and composition," Ph.D. dissertation, Department of Computing, City University, London, UK, 2005.

[18] M. R. Bjare, S. Lattner, and G. Widmer, "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.

[19] N. Fradet, J.-P. Briot, F. Chhel, A. El Fallah Seghrouchni, and N. Gutowski, "MidiTok: A python package for MIDI file tokenization," in *ISMIR (Late-breaking demo)*, 2021.

[20] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *ICLR*, 2015.

[21] P. Shaw, J. Uszkoreit, and A. Vaswani, "Self-attention with relative position representations," in *NAACL-HLT (2)*. Association for Computational Linguistics, 2018, pp. 464–468.

[22] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR (Poster)*, 2015.

[23] O. Mogren, "C-rnn-gan: A continuous recurrent neural network with adversarial training," in *Constructive Machine Learning Workshop (CML) at NIPS 2016*, 2016, p. 1.

MEASURING THE EUROVISION SONG CONTEST: A LIVING DATASET FOR REAL-WORLD MIR

John Ashley Burgoyne
University of Amsterdam
j.a.burgoyne@uva.nl

Janne Spijkervet
ByteDance
janne.spijkervet@gmail.com

David John Baker
University of Amsterdam
d.j.baker@uva.nl

ABSTRACT

Every year, several dozen, primarily European, countries, send performers to compete on live television at the Eurovision Song Contest, with the goal of entertaining an international audience of more than 150 million viewers. Each participating country is able to evaluate every other country's performance via a combination of rankings from professional jurors and telephone votes from viewers. Between fan sites and the official Song Contest organisation, a complete historical record of musical performances and country-to-country contest scores is available, back to the very first edition in 1956, and for the most recent contests, there is also information about each individual juror's rankings. In this paper, we introduce MIRoVision, a set of scripts which collates the data from these sources into a single, easy-to-use dataset, and a discrete-choice model to convert the raw contest scores into a stable, interval-scale measure of the competitiveness of Eurovision Song Contest entries across the years. We use this model to simulate contest outcomes from previous editions and compare the results to the implied win probabilities from bookmakers at various online betting markets. We also assess how successful content-based MIR could be at predicting Eurovision outcomes, using state-of-the-art music foundation models. Given its annual recurrence, emphasis on new music and lesser-known artists, and sophisticated voting structure, the Eurovision Song Contest is an outstanding testing ground for MIR algorithms, and we hope that this paper will inspire the community to use the contest as a regular assessment of the strength of modern MIR.

1. INTRODUCTION

The Eurovision Song Contest (ESC) is an annual event wherein several, primarily European, countries compete against one another by performing original, live songs during an internationally televised event. The contest began in 1956 and is typically held in the country of the previous year's winner in the spring.

© J. A. Burgoyne, J. Spijkervet, and D. J. Baker. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** J. A. Burgoyne, J. Spijkervet, and D. J. Baker, "Measuring the Eurovision Song Contest: A Living Dataset for Real-World MIR", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

The content of the musical acts performed during the Eurovision Song Contest is always novel and notably diverse. Contestants are allowed to sing in whichever language they choose, often electing to sing in English to communicate the meaning of their song to a larger base, but some countries (notably France) have historically preferred to sing in their national language. According to the official Eurovision rules, all musical acts must perform an original song that is no more than three minutes in length, with the lead vocals performed live, and acts are limited to only six performers being on stage at any given moment during the performance [1].

Within these constraints, the musical acts of Eurovision are known for their ostentatious performances and camp aesthetics, which are often accompanied with visual spectacles from lightening to elaborate dance. As the contest is an international stage, the musical acts have also been a means in which countries are able to provide meta-political commentary on either national or global events [2, 3]. The contest has been noted as serving as an important platform for global LGBTQ+ visibility, which featured openly gay and transgender performers as early as the 1990s [4].

The winner of the contest is determined as a combination of both expert and panel voting, with no set criteria stated as to what should constitute a winning performance. A combination of the song's content, the visual performance, and the performer's ability to relate to the *zeitgeist* are all presumed to play an important role in determining the winner. Indeed, the Eurovision Song Contest can be and has been analysed from a variety of dimensions, summarised by Wolther as the media, the musical, the musical-economical, the political, the national-cultural, the national-economic, and the competitive [5].

We next detail the rules of the contest before introducing the MIRoVision data set, which contains a multi-faceted collection of historical data that could be used to predict the contest's winner and enable researchers to make deeper inquiries into the history and music of the contest.

1.1 Rules of Eurovision

In order to participate in the Eurovision Song Contest, participating countries work in coordination with the European Broadcasting Union. While each participating country – or more specifically the country's partnered national broadcaster – is allowed to decide for themselves which act to send to participate, the results of the Eurovision Song Contest are determined by voting over three events. These

three events referred to as the First Semi-Final, the Second Semi-Final, and the Grand Final. It is the Grand Final that typically receives the vast majority of the attention and viewership.

As described on the official Eurovision website [1] , all participating countries qualify for two semi-final shows in the week leading up to the Grand Final, which only a subset of the total countries will perform. France, Germany, Italy, Spain and the United Kingdom are automatically included in the Grand Final and are referred to as the 'Big Five'.

After a country has performed, each other country gives two sets of votes for the performance. The first set of votes comes from an expert panel of music industry professionals from within that country. Starting in 2016, the official Eurovision website has published the individual data of each juror from each participating country. The second set of votes comes from viewers from the of the performing country. The votes represent points that are added together and each country can use their set of points, $\{12, 10, 8, 7, 6, 5, 4, 3, 2, 1\}$ for one and only one country. No juror or television vote can be cast for one's own country. In the semi-finals, voting is limited to only countries participating in their respective show, whereas in the Grand Final, any country is allowed to vote. The Grand Final television show is also characterised by great fanfare surrounding each national jury's announcement of which country they chose to award '*douze points*'.

No explicit criteria are given as how any vote should be decided. Said another way, it should not be assumed that all participants attempt to vote for a measure of musical quality. Many factors have been discussed in academic literature on the topic, that suggest there are both geographic and political factors that can play into how countries decide to cast their votes [6–10].

2. MIROVISION DATASET

Data that comprises the MIRoVision dataset originates from three primary sources. The first is the official Eurovison website (https://eurovision.tv/), the second is the Eurovision World fan website (https://eurovisionworld.com), the third are audio features taken directly from the YouTube videos linked in the contestant metadata. The dataset contains five primary types of data: (1) contest meta-data; (2) contest results; (3) voting data; (4) audio features extracted from recorded performances of the musical acts and (5) betting office data. All data for each Eurovision Song Contest is available each year since the year 1956 until present day with the exception of 2020 when the contest was cancelled due to the global COVID-19 pandemic. As of 2016, the official Eurovision website has published data detailing how each of the five jurors from the expert panel have voted on all three nights of the contest. The current release of the data set contains the contestant metadata, contest ranking and voting data of 1719 entries. The dataset is hosted on a GitHub repository. [2]

In total, 56 countries are represented in the dataset, which includes countries that have been dissolved, renamed, or merged since the inception of the contest in 1956. Voting data for the contest is stored in three tables: (1) votes; (2) contestants; and (3) jurors.

The *votes* table contains data from the contest's beginning in 1956 and indicates how each country's aggregated jury and televoting points were distributed to each other participating country.

The *contestants* table contains all metadata regarding each song entry, such as the artist's name and song title, lyrics, composers and lyricists, the running order and the total points awarded by the jury and televoters in the Semi-Final and Final Rounds respectively. This table also includes links to YouTube videos of live performances from the televised Finals or Semi-Finals, as maintained by the Eurovision World team.

The *jurors* table contains data beginning from the year 2016 and indicates how the five anonymous jurors (designated with letter names A through E) voted for each other country and in which night of the contest. As noted above, countries are unable to vote for themselves, are only able to vote within the Semi-Final they are participating in, whereas all countries are able to vote in the Grand Final.

In addition to the voting tables, the *betting-offices* table provide tables of historical bookmakers' odds for the contest winners, as collected by Eurovision World. The Eurovision Song Contest is a popular target for online betting. Day-of-contest odds are available for 2016 and 2017, and daily odds up to six months prior to the contest are available from 2018 onward, for 10 to 20 betting offices.

3. A PREFERENCE MODEL FOR EUROVISION

The Eurovision Song Contest voting system is iconic, but because the number of contestants varies, it is not possible to use contest scores to make comparisons across years. Moreover, the contest scores do not operate on an interval level of measurement: even within a particular year, a difference of five or ten points may mean something quite different at the top end of the score range than it does at the bottom. With the rich data in the MIRoVision set, however, it is possible to fit statistical models with parameters that correspond monotonically to actual contest results but that *do* behave on an interval scale. Such an interval scale is not only interesting musicologically and sociologically, but also for machine-learning applications, as most common loss functions for training implicitly assume interval-scale outcomes. In short, we are looking for a true *measure* of competitiveness in the Eurovision Song Contest, and one that applies stably across years.

In order to achieve these desiderata, the contest results must be sufficient statistics for the model parameters of interest. If we make the stronger assumption that there be only a finite number of sufficient statistics beyond these, then by the Pitman–Koopman–Darmois theorem [11], the model must be a member of the exponential family. That leaves a surprisingly small class of plausible models.

The simplest model requires no sufficient statistics other

[1] https://eurovision.tv/about/how-it-works

[2] https://github.com/Spijkervet/eurovision-dataset

than the scores themselves. Under such a model, the probability of the set of scores from any particular country's jury or televoters

$$\Pr[\text{ranking}] \propto \exp(s_1\beta_1 + s_2\beta_2 + \cdots + s_N\beta_N)\,, \quad (1)$$

where the coefficients $s_n \in \{12, 10, 8, 7, 6, 5, 4, 3, 2, 1\}$ are the scores awarded from that jury or televoter group to contestant n and the β_n are the model's competitiveness parameters for contestant n. The normaliser $Z_0(\boldsymbol{\beta})$ for this distribution is the sum of these terms for any valid assignment of scores under the Eurovision system. After M juries and televote groups combine their scores independently to determine a winner, the combined probability

$$\Pr[\text{contest}] = \frac{\exp(s_1\beta_1 + s_2\beta_2 + \cdots + s_N\beta_N)}{Z_0(\boldsymbol{\beta})^M}\,, \quad (2)$$

where $s_1, s_2, \ldots, s_N$ now represent the *total* scores awarded to each contestant. The trouble with this model is that for a typical Eurovision show of 26 contestants, the normaliser contains ${}_{26}P_{10} \approx 19$ trillion terms. The model is thus infeasible in practice, despite its theoretical simplicity.

Most alternatives to this model lose their exponential-family properties. There is, however, an interesting alternative if we are willing to consider Eurovision contest scores from juries and televoters to be *ratings* instead of rankings. Specifically, assume that for each song, juries must award a scores in the set $\{12, 10, 8, 7, 6, 5, 4, 3, 2, 1, 0\}$, but that there is no restriction on how many times they can use each score. While the numerator of such a model remains the same as (1) and (2), its normaliser

$$Z(\boldsymbol{\beta}) = \prod_{n=1}^{N} \sum_{k\in\{12,10,8,7,6,5,4,3,2,1,0\}} \exp(k\beta_n)\,, \quad (3)$$

which can be computed easily. Although there are fundamental conceptual and mathematical differences between rankings and ratings [12], if we restrict the outcome space of rating model (3) to allow only outcomes that would also be valid in the ranking model (2), the models are equivalent [13]. Moreover, we can add an extra set of score-level parameters ξ_k to allow (3) to better approximate (2) without sacrificing equivalency on the restricted outcome space:

$$\Pr[\text{ratings}] = \frac{\exp\left(\sum_n s_n\beta_n\right) \cdot \exp\left(\sum_k \xi_k\right)}{\prod_n \sum_k \exp(k\beta_i + \xi_k)}\,, \quad (4)$$

where s_n are again the scores from a particular jury or televoter group and $k \in \{12, 10, 8, 7, 6, 5, 4, 3, 2, 1, 0\}$. This model is known in the psychometric literature as the *partial-credit model* [14] and is one of the standard mathematical tools used for assessing the reliability of rubrics, Likert scales, and educational test items with partial credit.

4. FITTING THE PREFERENCE MODEL

We fit the partial-credit model (4) to the MIRoVision data for all Song Contests since 1975, the year that the {12, 10, 8, 7, 6, 5, 4, 3, 2, 1, 0} scoring system was instituted. We considered every vote available as an individual observation: every country's jury, every country's televotes in years that those votes were counted separately from juries, and all votes from semi-final rounds when they occurred. We made the important but unavoidable assumption that the average competitiveness of a Eurovision entry has remained constant over time, as there are no cross-year comparisons that would make it possible to estimate the model otherwise.

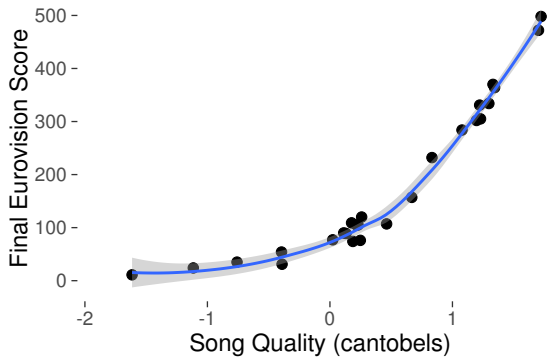


Figure 1. Correspondence between song competitiveness (in cantobels) and final Eurovision Song Contest scores in 2019. The pattern in this year is typical of all other years, with a relatively slow increase in points as competitiveness improves up to about 0.5 cantobels, followed by a rapid increase. Because of the semi-final rounds, the relationship between competitiveness and final score is not a strictly monotonic as in years without semi-finals, but it is still nearly monotonic.

We fit the joint probability model using the Bayesian probabilistic programming language Stan, with normal priors on average country competitiveness and song competitiveness and a multivariate normal prior on $\boldsymbol{\xi}$ for each contest. The complete model code is available in the supplemental material. For interpretive purposes, we fixed the mean of the song competitiveness parameters $\boldsymbol{\beta}$ to 0 and report them on a $10\log_{10}$ scale, analogous to the decibel. In honour of the singing at the contest, we deem this unit the *cantobel*. An increase of one cantobel in song competitiveness means that a song improves its chances of receiving one extra point from any given jury by $10^{\frac{1}{10}} \approx 1.26$. Like the decibel scale, an increase of 3 cantobels means that a song approximately doubles its chances of receiving one extra point.

Figure 1 illustrates the typical correspondence between competitiveness in cantobels and actual song contest results. After a slow increase, the slope rapidly increases for highly competitive entries. The Eurovision Song Contest scoring system compresses differences between relatively uncompetitive entries and dramatically exaggerates small differences at the top. While this surely contributes to the exciting television, cantobels are a better scale to use for scientific purposes.

Figures 2 and 3 reveal the heart of the model. The first shows the average song quality, as perceived by the Eurovision Song Contest juries and televoters, over the period from 1975 to 2022. Ukraine, Russia, Italy, and Sweden stand

Figure 2. Historical competitiveness of Eurovision Song Contest entries (in cantobels). Countries are coloured by their geographic region as defined in the United Nations M49 standard. Winners are boxed. The standard error of estimates is roughly 0.5 cantobel in early years and roughly 0.3 after the institution of semi-final rounds in 2004; as such, difference of approximately 1.0 cantobels are likely statistically significant. After a period when Northern and Western Europe exchanged victories, there was a period of Northern European dominance; recent years have been characterised by a good geographic diversity of winners.

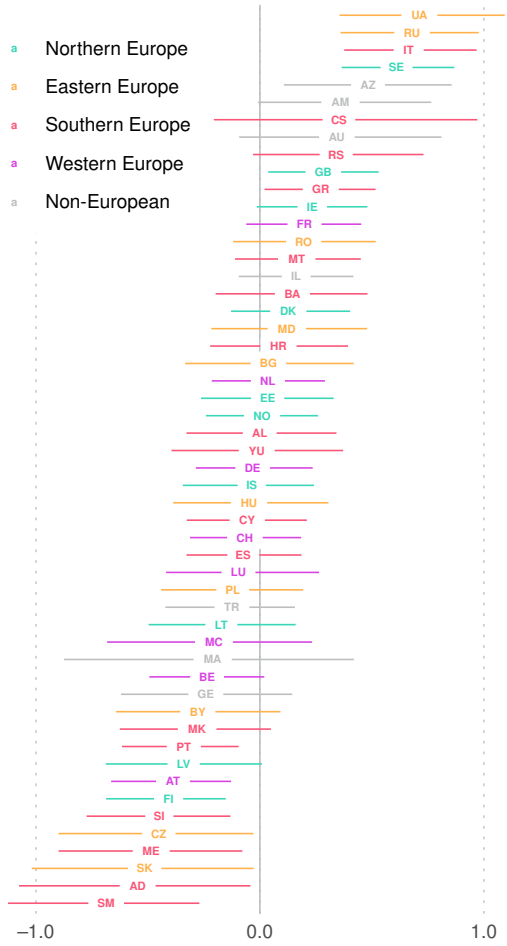


Figure 3. Median competitiveness of countries' Eurovision Song Contest entries, 1975–2022, in cantobels with 90% credible intervals. Countries are coloured by their geographic region as defined in the United Nations M49 standard. Ukraine, Russia, Italy, and Sweden stand out as having sent contestants of exceptional competitiveness, although Azerbaijan, the United Kingdom, and Greece's credible intervals are also strictly greater than zero.

out as having been particularly successful, even though they have suffered almost-wins instead of victories in many years. On average, songs from these countries have been a half cantobel above the average. But the first figure shows that there are dramatic swings from year to year underneath these averages. Even one the most convincing victories from one of the historically strongest countries – Måns Zelmerlöw's 'Heroes', Sweden's 2015 entry – was preceded and succeeded by much less appreciated acts.

4.1 Jury Model

Jury scores at the Eurovision Song Contest are determined by combining rankings from five independent jurors from each country, each of whom must make a complete ranking of contestants at a show, from best to worst. After averaging these ranks, they are converted to the better-known $\{12, 10, 8, 7, 6, 5, 4, 3, 2, 1, 0\}$ system that is reported on television. Since 2016, the European Broadcasting Union

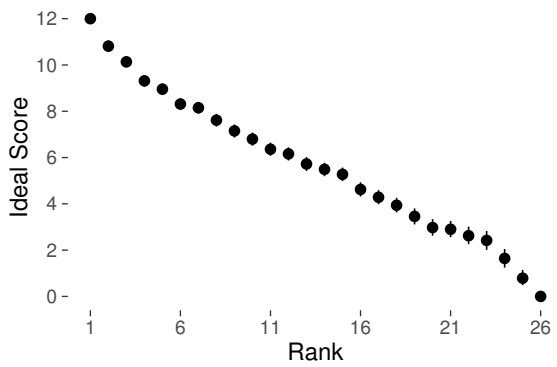


Figure 4. Ideal scores for averaging ranks within juries, according to a generalised partial-credit model, with 90% credible intervals. In recent years, the Eurovision Song Contest has used an exponential weighting scheme, but these results suggest that a linear scheme with a small bonus for the top-ranked entry would be sufficient.

has made not only the final scores but also these individual rankings public. They have also publicised that they continue to experiment with the proper way to average the ranks across jurors, currently using exponential decay. [3]

The theory of partial-credit models offers an alternative, more empirical solution. Rather than taking the scoring rule in (4) as fixed, the *generalised partial-credit model* considers an optimal scoring rule that would lead the model to make the best predictions. Concretely, that would mean considering alternatives to the $\{12, 10, 8, 7, 6, 5, 4, 3, 2, 1, 0\}$ rule for the main contest, and by extension, to the simpler $\{1, 2, \ldots, N\}$ rule for jury members making a full ranking.

The MIRoVision dataset includes these jury scores, and we fit a generalised partial-credit model to them analogous to the model we fit for the contest overall. The code is available in the supplemental material. Figure 4 shows the results. Like the European Broadcasting Union's current rule, we arbitrarily fix the maximum score to 12. It seems that rather than replacing the former linear scheme with the current exponential one may be a more effective simply to give a small fixed bonus to each juror's top-ranked entry. Such a solution would also solve the core issue motivating the exponential weighting, namely that it it undesirable for one juror to have unilateral power to spoil the chances of some other juror's favourite.

5. PREDICTING WINNERS

The Eurovision Song Contest is also notorious for attracting online and offline bets on the outcome. Since 2015, the EurovisionWorld web site has been collecting the odds posted at a large number of online betting offices, for each day leading up to the contest. These odds can be converted into implicit probabilities of winning, and there is often much discussion in the weeks leading up to the contest about which acts the bookmakers are favouring.

[3] `https://eurovision.tv/story/subtle-significant-ebu-changes-weight-individual-jury-rankings`

Year	Country	Actual	Bookmakers
2018	Israel	.87	.24
2018	Cyprus	.12	.37
2018	Germany	.01	.09
2019	Netherlands	.53	.51
2019	Italy	.45	.09
2019	Switzerland	.01	.09
2019	Russia	.01	.02
2021	Italy	.63	.26
2021	France	.35	.22
2021	Switzerland	.02	.05
2022	Ukraine	.98	.62
2022	Sweden	.01	.14
2022	United Kingdom	.01	.06
2022	Spain	.01	.06

Table 1. Probability of winning the Eurovision Song Contest, 2018–2022, given the partial-credit model and perfect information about jurors' and televoters' preferences, compared to bookmakers' implied win probabilities immediately prior to the contest final.

We can use our model fits to compare the bookmakers' predictions to the actual probabilities countries had to win given jurors' and televoters' preferences and the assumptions of the partial-credit model. To compute these probabilities, we reshuffled the draws from our Bayesian samples independently for each country and tallied how often these would have been the highest, taking advantage of the fact that competitiveness in cantobels is a sufficient statistics for actual contest outcomes. Table 1 presents the results. Both 2019 and 2021 were rather close contests, whereas 2018 and 2022 had clearer frontrunners. The bookmakers markedly mis-called 2018, but have been more accurate since. If one had been able to place stakes at the online betting offices with perfect knowledge of the jurors' and televoters' preferences, one would have quadrupled one's stake on average (before paying out the bookmakers' sometimes shockingly high margins on Eurovision odds).

6. CONTENT-BASED CONTEST PREDICTIONS

Perfect information is of course never available, but perhaps deep learning and content-based MIR offer something? Self-supervised music representation learning has advanced considerably in recent years. It has successfully been applied to many downstream tasks, including music tagging [16], genre classification, key detection and emotion recognition [17, 18]. These foundation models are generally pre-trained in an unsupervised, end-to-end fashion on raw audio samples. By defining an auxiliary loss objective on large quantities of music and using data perturbations, models are able to learn effective and robust representations.

To evaluate whether a pre-trained foundation model is able to predict preferences, we extracted embeddings on all song entries using the TUNe+ [19] and MERT [18] models. On every window of 2 seconds, an embedding vector of 512 feature dimensions is computed for the TUNe+ model. The MERT model returns 25 representation layers, and 1024 feature dimensions on 5-second windows. For every song entry between 1975 and 2022, a single embedding vector is calculated by taking the arithmetic mean along the time dimension for TUNe+ and along the representation layers for MERT respectively. This results in 1 261 embeddings in total. For every song entry, we took 4 000 draws from the fitted model for song competitiveness (in cantobels) and treated these as our targets Y; using 4 000 draws instead of a single point estimate more accurately averages over our uncertainty about song competitiveness, given the inherently limited number of rankings available for any single edition of the contest. We freeze the pre-trained TUNe+ and MERT models and perform a linear probe using the mean-squared error between $(\hat{y}, y)$. We use 5-fold cross-validation and sample all song entries from two years within each decade between 1975 and 2023 as our validation set.

Model	L1	L2
TUNe+ [19]	0.828 (0.039)	1.063 (0.052)
MERT [18]	0.820 (0.019)	1.025 (0.027)

Table 2. L1 (MAE) and L2 (RMSE) losses and their standard deviations after training two state-of-the-art audio embeddings to predict the competitiveness of Eurovision Song Contest entries from 1975–2022, in cantobels.

Our results in Table 2 show that we can achieve RMSE of 1.025 cantobels by way of training a linear layer on embeddings extracted from a pre-trained foundation model. These models are not specifically trained or designed for our downstream task of preference prediction, e.g., features extracted by the different layers in MERT vary in their downstream task performance, and we leave further improvements to future work. But to contextualise the result, the overall standard deviation of our Eurovision competitiveness ratings is 1.064 cantobels, which means that state-of-the-art MIR audio embeddings are able to predict 7.2% of the variance in Eurovision Song Contest competitiveness.

7. CONCLUSION

We present MIRoVision, a collection of data and tools for studying the Eurovision Song Contest and applying music information retrieval to several types of data generated from the contest. One of our key results is a model for converting the highly non-linear contest scores into a well-behaved interval-scale measurement we dub the *cantobel*. Cantobels facilitate understanding of fluctuations in the contest over time and more accurately represent both the competitiveness and the uncertainty surrounding the competitiveness of Eurovision Song Contest entries. They also behave better with the standard loss functions used in machine learning systems, and allow us to predict a small but meaningful portion of variance in contest outcomes. We hope this result is sufficiently tantalising to encourage the community to try their own models – the Eurovision Song Contest offers a fresh set of contestants every year – and to find their own creative uses for this rich musicological data source.

8. REFERENCES

[1] "How the Eurovision Song Contest works," Jul 2022. [Online]. Available: https://eurovision.tv/about/how-it-works

[2] C. Baker, "Wild dances and dying wolves: Simulation, essentialization, and national identity at the Eurovision Song Contest," *Popular Communication*, vol. 6, no. 3, pp. 173–189, 2008.

[3] J. K. O'Connor, *The Eurovision Song Contest: The Official History*. Carlton, 2010.

[4] C. Baker, "The gay olympics? the Eurovision Song Contest and the politics of LGBT/European belonging," *European Journal of International Relations*, vol. 23, no. 1, pp. 97–121, 2017.

[5] I. Wolther, "More than just music: The seven dimensions of the Eurovision Song Contest," *Popular Music*, vol. 31, no. 1, pp. 165–171, 2012.

[6] G. Yair, "'Unite Unite Europe': The political and cultural structures of europe as reflected in the Eurovision Song Contest," *Social Networks*, vol. 17, no. 2, pp. 147–161, 1995.

[7] G. Yair and D. Maman, "The persistent structure of hegemony in the Eurovision Song Contest," *Acta Sociologica*, vol. 39, no. 3, pp. 309–325, 1996.

[8] D. Fenn, O. Suleman, J. Efstathiou, and N. F. Johnson, "How does Europe make its mind up? Connections, cliques, and compatibility between countries in the Eurovision Song Contest," *Physica A*, vol. 360, pp. 576–598, 2006.

[9] V. Ginsburgh and A. G. Noury, "The Eurovision Song Contest: Is voting political or cultural?" *European Journal of Political Economy*, vol. 24, no. 1, pp. 41–52, 2008.

[10] M. Blangiardo and G. Baio, "Evidence of bias in the Eurovision Song Contest: Modelling the votes using Bayesian hierarchical models," *Journal of Applied Statistics*, vol. 41, no. 10, pp. 2312–2322, 2014.

[11] B. O. Koopman, "On distributions admitting a sufficient statistic," *Transactions of the American Mathematical Society*, vol. 19, pp. 399–409, 1936.

[12] S. J. Brams and P. C. Fishburn, "Voting procedures," in *Handbook of Social Choice and Welfare*, K. J. Arrow, A. Sen, and K. Suzumura, Eds. Elsevier, 2002, vol. 1, pp. 173–236.

[13] D. Andrich, "Understanding the response structure and process in the polytomous Rasch model," in *Handbook of Polytomous Item Response Theory Models*, M. L. Nering and R. Ostini, Eds. New York: Routledge, 2010, pp. 123–152.

[14] G. N. Masters, "A Rasch model for partial credit scoring," *Psychometrika*, vol. 47, no. 2, pp. 149–174, 1982.

[15] United Nations Statistics Division, "Standard country area codes for statistical use (M49)," 2021. [Online]. Available: https://unstats.un.org/unsd/methodology/m49/

[16] J. Spijkervet and J. A. Burgoyne, "Contrastive learning of musical representations," in *Proceedings of the 22nd Society for Music Information Retrieval Conference*, 2021.

[17] R. Castellon, C. Donahue, and P. Liang, "Codified audio language modeling learns useful representations for music information retrieval," *Proceedings of the 22nd Society for Music Information Retrieval Conference*, 2021.

[18] Y. Li, R. Yuan, G. Zhang, Y. Ma, C. Lin, X. Chen, A. Ragni, H. Yin, Z. Hu, H. He *et al.*, "Large-scale pretrained model for self-supervised music audio representation learning," Presentation at the Digital Music Research Network, 2022.

[19] M. A. Vélez Vásquez and J. A. Burgoyne, "Tailed U-net: Multi-scale music representation learning," in *Proceedings of the 23rd International Society for Music Information Retrieval Conference*, 2022.

EFFICIENT SUPERVISED TRAINING OF AUDIO TRANSFORMERS FOR MUSIC REPRESENTATION LEARNING

Pablo Alonso-Jiménez **Xavier Serra** **Dmitry Bogdanov**
Music Technology Group, Universitat Pompeu Fabra, Barcelona
pablo.alonso@upf.edu

ABSTRACT

In this work, we address music representation learning using convolution-free transformers. We build on top of existing spectrogram-based audio transformers such as AST and train our models on a supervised task using patchout training similar to PaSST. In contrast to previous works, we study how specific design decisions affect downstream music tagging tasks instead of focusing on the training task. We assess the impact of initializing the models with different pre-trained weights, using various input audio segment lengths, using learned representations from different blocks and tokens of the transformer for downstream tasks, and applying patchout at inference to speed up feature extraction. We find that 1) initializing the model from ImageNet or AudioSet weights and using longer input segments are beneficial both for the training and downstream tasks, 2) the best representations for the considered downstream tasks are located in the middle blocks of the transformer, and 3) using patchout at inference allows faster processing than our convolutional baselines while maintaining superior performance. The resulting models, MAEST, [1] are publicly available and obtain the best performance among open models in music tagging tasks.

1. INTRODUCTION

The goal of representation learning is to develop features that are suitable for a variety of tasks, rather than being specific to the training objective. In the context of audio, these features are sometimes referred to as embeddings, and they typically have a much lower dimensionality than the original signals, making them easier to store and process. When the embeddings are well-suited to a downstream task, it is often possible to achieve good performance using shallow models that require few resources to train and run. Additionally, using a single embedding model to feed several shallow classifiers or regressors is more efficient than having individual end-to-end models, and it simplifies addressing new related tasks with minimal additional effort. As a result, embedding models are valuable for a diverse range of applications, from quick prototyping without requiring detailed knowledge of audio processing to large-scale processing of industrial audio databases.

The universal success of transformers in text [1], vision [2], and audio [3] tasks motivate further research using this architecture for music representation learning. However, most state-of-the-art (SOTA) models are based on convolutional neural networks (CNNs) [4–7]. We hypothesize that transformers are not ruling this domain yet because they require large amounts of data and computational power to overcome their convolutional counterparts, while such resources are not always available. To address these challenges, we propose leveraging a large collection of 3.3 M tracks annotated with public-domain metadata from Discogs and using techniques to train transformers efficiently. Specifically, we focus on PaSST [8], a method that has demonstrated remarkable performance in the AudioSet [9] benchmark. This method uses patchout, a technique consisting of discarding parts of the input to regularize the training process, while also allows reducing the GPU memory and computations required for training. In this work, we investigate the effectiveness of this technique for music representation learning, considering the impact of specific design aspects.

We focus on the impact of using different combinations of tokens from different blocks of the transformer as embeddings, starting the training from different pre-trained weights from publicly available models, using different input segment lengths, and using patchout at inference time to speed up the embedding extraction. Our experiments show that the best performance is obtained by extracting embeddings from the middle of the transformer and initializing it with weights pre-trained on other audio tasks. Contrary to previous studies based on CNNs, our transformers benefit from long input segments both in training and different downstream scenarios. Finally, we show that, on certain patchout conditions, our transformers are able to double the inference speed of an EfficientNet-B0 baseline while producing embeddings that obtain better performance on downstream tasks. Moreover, this approach has the advantage of being entirely configurable at inference time, allowing the throughput/performance tradeoff to be adapted to the task at hand.

The remainder of this paper is structured as follows: In

[1] Music Audio Efficient Spectrogram Transformer. Code for training: https://github.com/palonso/MAEST. This model is part of Essentia models: https://essentia.upf.edu/models.html

© P. Alonso-Jiménez, X. Serra, and D. Bogdanov. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** P. Alonso-Jiménez, X. Serra, and D. Bogdanov, "Efficient Supervised Training of Audio Transformers for Music Representation Learning", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

Section 2 we present existing works related to this study. The experimental setup is presented in Section 3, and the proposed experiments and results are in Section 4. Finally, we conclude in Section 5.

2. BACKGROUND

In this section, we review the literature on music representation learning to motivate the selection of our training task and discuss existing audio and music transformers and justify our architecture and training approach. Finally, we introduce existing works on music representation learning with transformers.

2.1 Music representation learning

Some authors have pursued general-purpose representation models to address simultaneously speech, audio event, and music tasks, which led to the proposal of challenges such as HEAR [10] and benchmarks such as HARES [11]. However, for now, there is no evidence that a single training paradigm can yield excellent performance in all the audio domains at the same time. Alternatively, audio representations can be optimized to a single domain leveraging specific data, which tends to produce better performance. In this sense, music-specific representation models are typically evaluated in music description in terms of genre, mood, era, rhythmic properties or arousal and valence estimation, where the annotations are generally on the track level. Additionally, music representation models can be evaluated in more objective tasks such as tempo or key estimation, although, specific models using domain knowledge tend to be better suited for these tasks [12].

Music tagging is a multi-label classification task using a vocabulary that can combine multiple music notions (e.g., genre, moods, eras). Some of the most successful music representation learning approaches are based on music tagging [5, 13–15]. Other directions include training models on editorial metadata [4,6,16–20], multi-modal correspondence [21], co-listening statistics [4], contrastive supervised [7,22–24] and self-supervised [11,25–28] objectives, music generative models [29], playlist co-occurrences [20, 24], text [7, 30], or combinations of them [4, 19, 24, 29]. While self-supervised approaches have been narrowing the gap with their supervised counterparts, the SOTA models use music tagging [4,5], or supervised contrastive learning in a single-domain [6] or cross-domain [7] settings. Since the scope of this work is to assess the benefits of transformers, we fix our training task to music tagging for its simplicity, popularity, and empirically shown effectiveness.

2.2 Transformers in audio classification tasks

Transformers have become a popular choice for audio tasks due to their superior performance compared to their convolutional counterparts when sufficient data is available. Lately, AudioSet, with almost 2 M audio event excerpts, has become a popular benchmark led by transformer models. A popular approach consists of applying attention over small overlapping patches (e.g., 16 $\times$ 16) from the spectrogram using a classification objective. The sequence of spectrogram patches is linearly projected to a 1-D space where a trainable positional encoding signal is added. A trainable classification token is appended to the sequence of projections, and after a number of Transformer blocks it is used to solve the classification task using a linear classifier. This idea was first introduced in the image domain by ViT [2] and adapted to audio spectrograms in AST [3]. PaSST extends this approach by introducing *patchout*, a technique consisting of discarding random patches from the input spectrogram at training time (see Figure 1) [8]. This technique has two benefits. First, by discarding input patches, the training sequence length is significantly reduced, which increases the training speed. Second, it acts as a regularization technique that improves the robustness of the transformer. Additionally, patchout can be combined with other training methods. MaskSpec is a self-supervised pre-training method based on an encoder-decoder architecture where the decoder has to reconstruct the spectrogram from a partial spectrogram altered with patchout [31]. Beats is a transformer trained with a supervised objective and patchout where the labels come from a codebook of randomly initialized vectors that is iteratively optimized [32]. While these techniques prevent the transformers from depending on initializing from weights of pre-trained models, such systems are significantly more resource-demanding. Table 1 compares the mentioned audio transformers in terms of GPUs used for training, training duration, and mean Average Precision (mAP) on AudioSet. Remarkably, PaSST achieves an excellent trade-off between mAP and needed resources. Since we aim to use transformer models that can be trained with a computational budget equivalent to SOTA CNNs (i.e., using consumer-grade GPUs), we focus on the standard patchout training with a supervised objective.

Model	Init.	GPUs	Time	mAP
AST [3]	ViT	-	-	45.9
PaSST [8]	DeiT	2 RTX 2080ti	24 h	47.6
MaskSpec [31]	FS	64 Tesla V100	36 h	47.3
Beats [32]	FS	16	-	48.7

Table 1. Comparison transformers from the literature in terms of initialization weights, number of GPUs used for training, training time, and mAP obtained in AudioSet.

2.3 Music representation learning with transformers

Some works already combined music representation learning and pure-attention-based transformers. S3T combines MoCo's momentum-based self-supervised contrastive learning with the Swin Transformer [33] architecture to learn music representations for classification [28]. MuLan is an audio representation model trained with cross-domain contrastive learning that aligns the latent representations of associated audio and text pairs. The authors experiment both with a ResNet50 and an AST architecture, with the former obtaining better performance in downstream music tagging tasks [7].

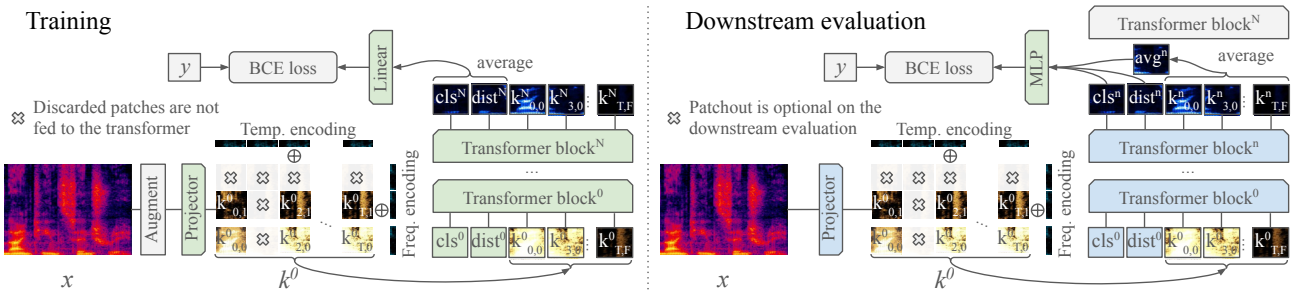


Figure 1. Illustration of our system at the training and downstream evaluation stages where x is the input spectrogram, k^0 is the sequence of tokens after the patchout, y is the target labels, and BCE is the binary cross-entropy loss. Trainable and frozen blocks are colored green and blue respectively.

The limited list of studies combining transformers and music representation learning motivates further research. We propose addressing this by using a simple supervised objective and patchout.

3. EXPERIMENTAL SETUP

We train our models using an in-house dataset with 3.3 M tracks mapped to the Discogs' public metadata dump. [2] The training task consists of a multi-label classification of the top 400 music styles from Discogs' taxonomy. We compare different training configurations in several downstream tasks by training Multi-Layer Perceptrons (MLP) on representations extracted from the transformers.

3.1 Dataset and pre-processing

Our dataset is derived from a pool of 4 M audio tracks mapped to the release information from the Discogs website's public dump. [3] All release metadata, which can include music style tags following a pre-defined taxonomy, is submitted by the community of platform users. *Master releases* group different versions of the same release such as special editions, or remasters. We obtain our training labels, y, at the master release level by first aggregating the style tags of all the associated releases and then discarding master releases with more than five style tags or without any style label among the 400 most frequent among our pool of tracks. We keep tracks longer than 20 seconds. Since the style annotations are done at the master release level, the resulting track annotations are expected to be noisy. We generate validation and testing subsets with approximately 40,000 tracks and a training set with 3.3 M tracks, ensuring that every artist appears on a single split. This pre-processing is similar to our previous work [6], and additional details and statistics about the resulting dataset can be found in the repository accompanying this publication. For now on, we refer to this internal dataset as *Discogs20*.

From every track, we sample 30 seconds from the center of the track and downmix it to a mono channel at 16 kHz. We extract 96-bands mel-spectrograms, x, using 32 ms windows and a hop size of 16 ms compressed with the expression $log_{10}(1 + 10000x)$ similar to previous works in music tagging [6, 34]. The resulting representations are stored as half-precision floats (16 bits) resulting in 1.3 TB of data. Given that our dataset is in the order of magnitude of AudioSet (1.8 M vs. 3.3 M) and presents similar label density (2.7 average labels in AudioSet and 2.1 in Discogs20), we adopt the sampling strategy used in previous works [8]. Every epoch, we take a balanced sample of 200,000 tracks without replacement using the inverse label frequencies as sample weight. We normalize the input to the mean and standard deviation of the training set.

3.2 Model and training

Our transformer, *MAEST*, has the same architecture as AST [3], ViT [2], or PassT [8], and features 12 blocks of self-attention plus a dense layer resulting in close to 87 million parameters. We use 16 $\times$ 16 patches, $x_{t,f}$, with a stride of 10 $\times$ 10. Similar to PaSST, we split the positional encoding into time/frequency encodings (te_t, fe_f) and apply patchout by randomly discarding entire rows and columns from the sliced spectrogram. The input sequence of tokens, k^0, is created as a linear projection of the patches plus the correspondent time/frequency encodings, $k^0_{t,f} = P(x_{t,f}) + te_t + fe_f$, where $P(\cdot)$ is a trainable linear layer. [4] k^1 to k^{12} represent the output tokens of the respective transformer blocks. Similar to DeiT [35] and PaSST, we extend k^0 with classification (cls^0) and distillation ($dist^0$) trainable tokens, which are initialized with the DeiT or PaSST pre-trained weights in the experiments involving these models. [5] We take the average of cls^{12} and dis^{12} tokens to feed a linear classifier targeting y.

We use the Adam Optimizer with a weight decay of $1e{-4}$ and train the model for 130 epochs. We warm up the model for 5 epochs and then keep the learning rate at $1e{-4}$ until epoch 50. Then the learning rate is linearly decreased to $1e{-7}$ during 50 additional epochs. We consider two sets of weights for inference: those from the last epoch and

[2] `https://www.discogs.com/data/`

[3] In Discogs, releases include albums, EPs, compilations, etc.

[4] Since the mel scale is not linear, we considered specialized projectors for each frequency patch. However, this did not improve the performance.

[5] We considered a teacher-student approach similar to DeiT by using a pre-trained MAEST-30 to generate pseudo-labels that were targeted by the $dist^{12}$ token in the training stage. We decided to omit the experiment details since it did not achieve a significant improvement.

Dataset	Size	Lab.	Dur.	Av.	Split
MTGJ-Genre	55,215	87	FT	2.44	split 0 [38]
MTGJ-Inst	25,135	40	FT	2.57	split 0 [38]
MTGJ-Moods	18,486	56	FT	1.77	split 0 [38]
MTGJ-T50	54,380	50	FT	3.07	split 0 [38]
MTT	25,860	50	29s	2.70	12-1-3 [39]
MSDs	241,889	50	30	1.72	usual [15]
MSDc	231,782	50	30	1.31	CALS [40]

Table 2. Automatic tagging datasets used in the downstream evaluation. The datasets are compared in terms of sample size, number of labels, audio duration (Full Tracks or excerpts of fixed duration), average labels per track, and the splits used in our evaluations.

those obtained by taking the mean of the model's weights every 5 epochs from epoch 50 using Stochastic Weight Averaging (SWA). We pre-compute the mel-spectrograms for efficiency, which limits the set of data augmentations we could apply. We use mixup [36] with $alpha = 0.3$ and SpecAugment [37] by masking up to 20 groups of 8 timestamps and up to 5 groups of 8 frequency bands. [6]

Initialization weights. Previous works showed the importance of initializing the transformer to weights pre-trained on ImageNet [3]. To gain further knowledge, we consider three initialization options: the DeiT B↑384 model pre-trained on ImageNet [35], the PaSST S S16 model pre-trained on mel-spectrograms from AudioSet, and random initialization.

Spectrogram segment length. We consider spectrogram segment lengths of 5 to 30 seconds resulting in the architectures MAEST-5s, MAEST-10s, MAEST-20s, and MAEST-30s. In all cases, we take existing PaSST frequency and temporal encodings and interpolate them to the target shape as an initialization. We use patchout discarding 3 frequency and 15 temporal patches for MAEST-5s and increase the temporal patchout proportionally for models with longer input sequences (e.g., 60 patches for MAEST-20s).

3.3 Evaluation

We evaluate our models in several music automatic tagging datasets covering various musical notions. We consider the popular MagnaTagATune (MTT) and the Million Song Dataset (MSD) with the commonly used training, validation, and testing splits used in [39] and [15] respectively. Additionally, we report the performance of our models in the CALS split, which is an artist-filtered version of the MSD ground truth [40]. Finally, we use the MTG-Jamendo Dataset, a dataset of Creative Commons music containing sub-taxonomies with the tags related to genre (MTGJ-Genre), moods and themes (MTGJ-Mood), and instrumentation (MTGJ-Inst), along with the top 50 tags (MTGJ-T50) in the dataset. We use the official split 0 for all the subsets similar to previous works [5, 30, 41]. Table 2 summarizes these datasets in terms of size, number of labels, audio duration, average number of labels per track, and used splits.

We evaluate our models by extracting internal representations from different blocks of the transformer and training MLP classifiers on top. Instead of averaging the cls^{12} and $dist^{12}$ tokens as done in the training stage, we consider three types of representations, cls^n, $dist^n$, and the average of the tokens representing the input spectrogram patches (avg^n) after n transformer blocks. Additionally, we evaluate the complementarity of these embeddings training MLP classifiers on stacks of the different tokens. To generate the dataset of embeddings, we average the embeddings extracted from half-overlapped segments across the entire audio available for the tracks in the downstream datasets. The same setup is used for the training, validation and testing stages.

The downstream model is an MLP with a single-hidden layer of 512 dimensions with a ReLU activation and dropout. In the experiments described in Sections 4.1, 4.2, 4.3, and 4.5, we use a batch size of 128, drop out of 0.5 and train the model for 30 epochs. In the downstream evaluation from Section 4.4, we perform a grid search over the following hyper-parameters for each task:

- **batch size**: {64, 128, 256}
- **epochs**: {30, 40, 50, 60, 70, 80}
- **drop out**: {0.5, 0.75}
- **maximum learning rate**: $\{1e{-}3, 1e{-}4, 5e{-}4, 1e{-}5\}$

The MLP is trained with the binary cross-entropy loss using the Adam optimizer with a weight decay of $1e{-}3$. The learning rate is exponentially raised to its maximum value during the first 10 epochs, kept constant for the number of epochs, and linearly reduced until reaching $1e{-}7$ at the end of training. After training, we report the performance on the testing set obtained using the weights from the epoch with the highest validation ROC-AUC.

4. EXPERIMENTS AND RESULTS

In this section, we present the conducted experiments and discuss the results.

4.1 Extracting embeddings from the transformer

We are interested in finding the optimal representations from the transformer to be used as embeddings. To do this, we extract representations cls^n, $dist^n$, and avg^n from different transformer blocks $n \in [5, 12]$. To measure the complementarity of these features, we train MLPs fed with stacks of combinations of these representations. In this experiment, we use MAEST-30s intialized with PaSST weights and the MTT dataset.

Figure 2 shows mAP scores obtained with different stacks of embeddings extracted from the different transformer blocks. In accordance with previous studies [29], we find that the embeddings with the best performance are found in the middle blocks of the transformer. This contrasts with the typical behavior of CNNs, where the best

[6] We trained MAEST using 4 Nvidia 2080 RTX Ti GPUs with 12GB of RAM. The training takes 31 hours for MAEST-5 and 48 hours for MAEST-30.

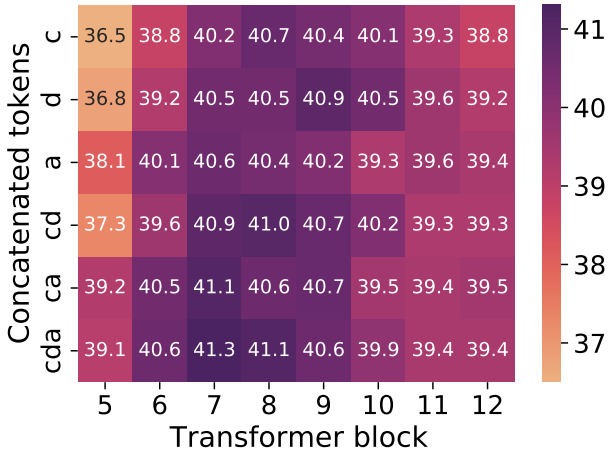


Figure 2. mAP scores obtained with our evaluation setup in the MTT dataset using embeddings extracted from different blocks and tokens transformer. We evaluate the cls (c), $dist$ (d), and avg (a) tokens and stacks of their combinations extracted from the transformer blocks 5 to 12.

features are normally towards the last layers of the model, especially, when the downstream task is well aligned with the training task. Also, concatenating the features benefits the performance. In the remaining experiments, we fix our embedding to the stack (cls^7, $dist^7$, avg^7).

4.2 Impact of the initial weights

Due to the lack of inductive biases present in architectures such as CNNs, transformers are heavily dependent on pre-training. Because of this, many audio transformers are initialized with weights transferred from image tasks [3, 8]. We evaluate the impact of initializing our models from the weights of DeiT [35] (image input), the best single PaSST model [8] (mel-spectrogram input), and random initialization. In this experiment, we use MAEST-10s and its version with SWA weights, MAEST-10s-swa. Although our main focus is to evaluate MAEST on public downstream datasets, we also report their performance on the training task to provide additional insights.

Table 3 shows the performance in both, the training (Discogs20), and a downstream (MTT) task. In both cases, the scores are higher when the training is started from pre-trained weights. Since the PaSST weights result in slightly higher performance, we use this initialization for the remaining of this work. Regarding the SWA, we observe a positive effect on the training task when the model is initialized with pre-trained weights. However, we do not observe improvements in the downstream task.

4.3 Effect of the input segment length

We train MAEST using input segment lengths ranging from 5 to 30 seconds. In our experiments, we keep the frequency patchout constant and proportionally increase the temporal patchout. For our models with segment lengths of 5, 10, 20, and 30 seconds we discard 15, 30, 60, and 90 temporal patches respectively.

Model	RW	DeiT	PaSST
Pre-training task: Discogs20			
MAEST-10s	20.5	22.7	22.8
MAEST-10s-swa	20.1	23.2	23.5
Downstream task: MTT			
MAEST-10s	38.7	40.4	41.1
MAEST-10s-swa	39.0	40.2	41.0

Table 3. mAP scores obtained in the training and downstream tasks using different initializations. We considered Random Weights, and pre-trained weights from DeiT and PaSST.

Table 4 shows the performance of the MAEST models with respect to their input spectrogram segment length in terms of mAP both in the training (Discogs20) and a downstream (MTT) evaluation. While music tagging CNNs tend to reach their peak of performance with receptive fields of 3 to 5 seconds [14], attention-based systems have shown the capability to take advantage of longer temporal contexts [40]. Our models are consistent with this trend, reaching their best performance when trained on segments of 30 seconds. Although even longer segments could be beneficial, we could not use them while keeping the same model size due to GPU memory limitations.

4.4 Performance in downstream tasks

Considering our previous findings, we extend the evaluation of MAEST to a number of downstream datasets. We evaluate MAEST-10s, MAEST-20s, MAEST-30s, and a baseline consisting of embeddings from the penultimate layer of an EfficientNet-B0 (EffNet-B0) architecture [43] trained in the same 400 music style tags from Discogs20 following previous work [6]. Additionally, we report the performance of SOTA models from the literature considering approaches fully trained in the downstream tasks and based on embeddings plus shallow classifiers.

Table 4 shows the results of the different models in terms of ROC-AUC and mAP. We observe that all the MAEST models outperform the baseline in all tasks, confirming the superiority of the proposed approach. Additionally, we achieve a new SOTA for the MTGJ-Genre, MTGJ-Inst, and MSDc datasets, although other models remain superior in the rest of the datasets. Specifically, MuLan [7] obtains higher mAP in MTT, probably because it is

Model	5s	10s	20s	30s
Pre-training task: Discogs20				
MAEST-T	21.1	22.8	24.8	26.1
MAEST-T-swa	21.3	23.5	25.8	27.0
Downstream task: MTT				
MAEST-T	40.8	41.1	41.2	41.7
MAEST-T-swa	40.9	41.0	41.2	41.5

Table 4. mAP scores obtained in the training and downstream tasks using different spectrogram segment lengths. T represents the spectrogram segment length.

	MTGJ-Genre		MTGJ-Inst		MTGJ-Mood		MTGJ-T50		MTAT		MSDs		MSDc	
	ROC	mAP	ROC	mAP	ROC	mAP	ROC	mAP	ROC	mAP	ROC	mAP	ROC	mAP
State of the art														
Fully-trained	-	-	-	-	77.8	15.6	83.2	29.8	90.69	38.44	92.2	38.9	89.7	34.8
	-	-	-	-	*[42]*	*[42]*	*[34]*	*[34]*	*[41]*	*[41]*	*[40]*	*[40]*	*[40]*	*[40]*
Embeddings	87.7	19.9	77.6	19.8	78.6	16.1	84.3	32.1	92.7	41.4	-	-	90.3	36.3
	[6]	*[6]*	*[6]*	*[6]*	*[5]* †	*[5]* †	*[5]* †	*[5]* †	*[7]* †	*[5]* †	-	-	*[5]* †	*[5]* †
Baseline														
EffNet-B0	87.7	19.9	77.6	19.8	75.6	13.6	83.1	29.7	90.2	37.4	90.4	32.8	88.9	32.8
Our models														
MAEST-10s	88.1	21.1	79.7	22.4	77.9	15.1	84.0	31.3	91.8	41.0	91.5	36.9	88.9	32.7
MAEST-20s	88.1	21.4	79.9	22.6	77.9	15.2	**84.1**	**31.5**	91.8	41.0	92.1	39.2	89.5	34.5
MAEST-30s	**88.2**	**21.6**	**80.0**	**22.9**	**78.1**	**15.4**	84.0	**31.5**	**92.0**	**41.9**	**92.4**	**40.7**	**89.8**	**35.4**

Table 5. ROC-AUC and mAP scores obtained in the downstream tasks. Our baseline consists of an EffNet-B0 architecture trained in Discogs20. Additionally, we report the SOTA results distinguishing models with all parameters trained in the downstream tasks (fully trained) and models evaluated with shallow classifiers. For every task, we mark in bold the best score obtained by a MAEST model and highlight in grey models achieving better performance than the best open alternative. † Models not publicly available.

trained on a much larger corpus of 40 M tracks. In MTGJ-Moods, MTGJ-T50, MTT, and MSDs, Musicset-Sup, a model trained on a curated dataset of 1.8 M expert annotations, remains superior [5]. In both cases, the advantage is likely due to the superiority of the training task. Notably, none of these models is public, which makes MAEST the best open music embedding extractor available.

4.5 Faster feature extraction with inference patchout

Inferring with transformers is typically more computationally expensive than with CNNs. To speed up our models, we consider using two types of patchout at inference time: Time-wise, we keep one out of T spectrogram patches. Frequency-wise, we discard specific rows of patches. We experiment with temporal patchout using $T \in [2, 3, 5, 10]$ and frequency patchout of 3 and 4 patches corresponding to the first and the two last blocks, and the two first and two last blocks respectively. The embeddings obtained under different patchout settings are compared in the training and a downstream task following our downstream evaluation approach on the MTT dataset.

Figure 3 shows the mAP scores on the training and downstream tasks under different patchout settings. In the downstream task, even under strong patchout settings, MAEST-30s overcomes the throughput of standard CNN architectures by two to three times while keeping higher mAP. On the training task, this technique is not so effective because the classifier is frozen and cannot adapt to the effects of patchout, and also it operates on tokens from the last block, which requires more computations.

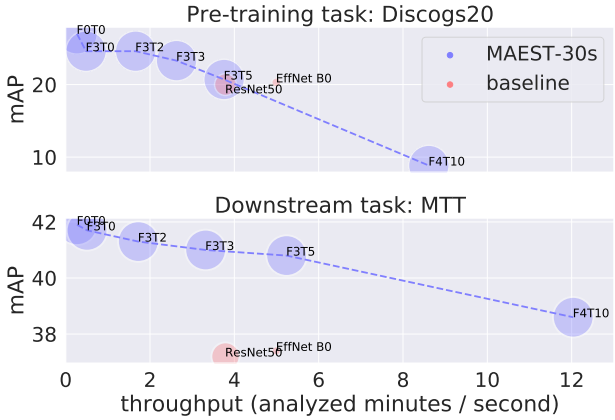


Figure 3. mAP scores against throughput for MAEST-30s under different amounts of frequency (F) and time (T) patchout. The radius is proportional to the parameter count and the inference is performed on the CPU.

5. CONCLUSION

In this work, we demonstrate the benefits of pure-attention-based transformers for music representation learning and study how different design decisions affect the downstream performance. Our experiments show that the best embeddings come from a stack of features from the middle blocks of the transformer, initializing from weights pre-trained in audio event recognition provides the best performance, and that longer input segments correlate with better results. We evaluate our models in six popular music tagging datasets, and experiment with patchout at inference time, finding that it allows speeding up significantly the transformer while producing embeddings with better performance/speed trade-offs than our convolutional baselines. Finally, we present MAEST, a family of transformers for music style tagging and embedding extraction, which are publicly available and achieve SOTA performance among currently available music representation models.

In future work, we will combine our architecture with additional training objectives combining supervised and self-supervised paradigms. Additionally, we will experiment with longer input segments and teacher-student setups suitable for noisy datasets such as ours.

6. ACKNOWLEDGEMENTS

This work has been supported by the Musical AI project - PID2019-111403GB-I00/AEI/10.13039/501100011033, funded by the Spanish Ministerio de Ciencia e Innovación and the Agencia Estatal de Investigación.

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*, 2017.

[2] 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 Intl. Conf. on Learning Representations (ICLR)*, 2021.

[3] Y. Gong, Y. Chung, and J. R. Glass, "AST: audio spectrogram transformer," in *22nd Annual Conf. of the Intn. Speech Communication Association (Interspeech)*, 2021.

[4] Q. Huang, A. Jansen, L. Zhang, D. P. Ellis, R. A. Saurous, and J. Anderson, "Large-scale weakly-supervised content embeddings for music recommendation and tagging," in *Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.

[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 *Intl. Society for Music Information Retrieval Conf. (ISMIR)*, 2022.

[6] P. Alonso-Jiménez, X. Serra, and B. Dmitry, "Music representation learning based on editorial metadata from Discogs," in *Intl. Society for Music Information Retrieval Conf. (ISMIR)*, 2022.

[7] 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 *Intl. Society for Music Information Retrieval Conf. (ISMIR)*, 2022.

[8] K. Koutini, J. Schlüter, H. Eghbal-zadeh, and G. Widmer, "Efficient training of audio transformers with patchout," in *23rd Annual Conf. of the Intl. Speech Communication Association (Interspeech)*, 2022.

[9] 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 *Intl. Conf. on acoustics, speech and signal processing (ICASSP)*, 2017.

[10] 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 *Conf. on Neural Information Processing Systems (NeurIPS)*, D. Kiela, M. Ciccone, and B. Caputo, Eds., 2021.

[11] L. Wang, P. Luc, Y. Wu, A. Recasens, L. Smaira, A. Brock, A. Jaegle, J.-B. Alayrac, S. Dieleman, J. Carreira *et al.*, "Towards learning universal audio representations," in *Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.

[12] H. Schreiber and M. Meinard, "A single-step approach to musical tempo estimation using a convolutional neural network," in *Intl. Society for Music Information Retrieval Conf. (ISMIR)*, 2018.

[13] A. van den Oord, S. Dieleman, and B. Schrauwen, "Transfer learning by supervised pre-training for audio-based music classification," in *Intl. Society for Music Information Retrieval Conf. (ISMIR)*, 2014.

[14] K. Choi, G. Fazekas, M. Sandler, and K. Cho, "Transfer learning for music classification and regression tasks," in *Intl. Society for Music Information Retrieval Conf. (ISMIR)*, 2017.

[15] J. Lee, J. Park, K. Kim, and J. Nam, "SampleCNN: End-to-end deep convolutional neural networks using very small filters for music classification," *Applied Sciences*, 2018.

[16] J. Park, J. Lee, J.-W. Ha, and J. Nam, "Representation learning of music using artist labels," in *Intl. Society for Music Information Retrieval Conf. (ISMIR)*, 2018.

[17] J. Kim, M. Won, X. Serra, and C. C. S. Liem, "Transfer learning of artist group factors to musical genre classification," *Intl. World Wide Web Conf.*, 2018.

[18] J. Lee, J. Park, and J. Nam, "Representation learning of music using artist, album, and track information," in *Intl. Conf. on Machine Learning (ICML), Machine Learning for Music Discovery Workshop*, 2019.

[19] 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*, 2020.

[20] 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 *Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2023.

[21] J. Cramer, H.-H. Wu, J. Salamon, and J. P. Bello, "Look, listen, and learn more: Design choices for deep audio embeddings," in *Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2019.

[22] X. Favory, K. Drossos, T. Virtanen, and X. Serra, "COALA: co-aligned autoencoders for learning semantically enriched audio representations," in *Workshop on Self-supervised Learning in Audio and Speech, Intl. Conf. on Machine Learning (ICML)*, 2020.

[23] ——, "Learning contextual tag embeddings for cross-modal alignment of audio and tags," in *Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2021.

[24] A. Ferraro, X. Favory, K. Drossos, Y. Kim, and D. Bogdanov, "Enriched music representations with multiple cross-modal contrastive learning," *Signal Processing Letters*, 2021.

[25] J. Spijkervet and J. A. Burgoyne, "Contrastive learning of musical representations," in *Intl. Society for Music Information Retrieval Conf. (ISMIR)*, 2021.

[26] D. Niizumi, D. Takeuchi, Y. Ohishi, N. Harada, and K. Kashino, "Byol for audio: Self-supervised learning for general-purpose audio representation," in *2021 Intl. Joint Conf. on Neural Networks (IJCNN)*, 2021.

[27] D. Yao, Z. Zhao, S. Zhang, J. Zhu, Y. Zhu, R. Zhang, and X. He, "Contrastive learning with positive-negative frame mask for music representation," in *Intl. World Wide Web Conf.*, 2022.

[28] H. Zhao, C. Zhang, B. Zhu, Z. Ma, and K. Zhang, "S3t: Self-supervised pre-training with swin transformer for music classification," in *Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.

[29] R. Castellon, C. Donahue, and P. Liang, "Codified audio language modeling learns useful representations for music information retrieval," in *Intl. Society for Music Information Retrieval Conf. (ISMIR)*, 2021.

[30] I. Manco, E. Benetos, E. Quinton, and G. Fazekas, "Learning music audio representations via weak language supervision," in *IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.

[31] D. Chong, H. Wang, P. Zhou, and Q. Zeng, "Masked spectrogram prediction for self-supervised audio pre-training," *arXiv preprint arXiv:2204.12768*, 2022.

[32] S. Chen, Y. Wu, C. Wang, S. Liu, D. Tompkins, Z. Chen, and F. Wei, "BEATs: Audio pre-training with acoustic tokenizers," *arXiv preprint arXiv:2212.09058*, 2022.

[33] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. of the Intl. Conf. on Computer Vision (ICCV)*, 2021.

[34] J. Pons and X. Serra, "musicnn: Pre-trained convolutional neural networks for music audio tagging," *Late-Braeking/Demo, Intl. Society for Music Information Retrieval Conf. (ISMIR)*, 2019.

[35] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *Intl. Conf. on Machine Learning (ICML)*, 2021.

[36] H. Zhang, M. Cissé, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *Intl. Conf. on Learning Representations (ICLR)*, 2018.

[37] 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 *Annual Conf. of the Intl. Speech Communication Association (Interspeech)*, 2019.

[38] D. Bogdanov, M. Won, P. Tovstogan, A. Porter, and X. Serra, "The MTG-Jamendo dataset for automatic music tagging," in *Intl. Conf. on Machine Learning (ICML)*, 2019.

[39] A. van den Oord, S. Dieleman, and B. Schrauwen, "Transfer learning by supervised pre-training for audio-based music classification," in *Conf. of the Intl. Society for Music Information Retrieval (ISMIR)*, 2014.

[40] M. Won, K. Choi, and X. Serra, "Semi-supervised music tagging transformer," in *Intl. Society for Music Information Retrieval Conf. (ISMIR)*, 2021.

[41] M. Won, A. Ferraro, D. Bogdanov, and X. Serra, "Evaluation of cnn-based automatic music tagging models," in *Sound and Music Computing Conf. (SMC)*, 2020.

[42] D. Knox, T. Greer, B. Ma, E. Kuo, K. Somandepalli, and S. Narayanan, "Mediaeval 2020 emotion and theme recognition in music task: Loss function approaches for multi-label music tagging," in *Proc. of the MediaEval 2020 Workshop*, 2020.

[43] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *Intl. Conf. on Machine Learning (ICML)*, 2019.

A CROSS-VERSION APPROACH TO AUDIO REPRESENTATION LEARNING FOR ORCHESTRAL MUSIC

Michael Krause[1] **Christof Weiß**[2] **Meinard Müller**[1]
[1] International Audio Laboratories Erlangen, Germany
[2] University of Würzburg, Germany
{michael.krause,meinard.mueller}@audiolabs-erlangen.de, christof.weiss@uni-wuerzburg.de

ABSTRACT

Deep learning systems have become popular for tackling a variety of music information retrieval tasks. However, these systems often require large amounts of labeled data for supervised training, which can be very costly to obtain. To alleviate this problem, recent papers on learning music audio representations employ alternative training strategies that utilize unannotated data. In this paper, we introduce a novel cross-version approach to audio representation learning that can be used with music datasets containing several versions (performances) of a musical work. Our method exploits the correspondences that exist between two versions of the same musical section. We evaluate our proposed cross-version approach qualitatively and quantitatively on complex orchestral music recordings and show that it can better capture aspects of instrumentation compared to techniques that do not use cross-version information.

1. INTRODUCTION

Deep learning (DL) has become a common tool for approaching diverse tasks in music information retrieval (MIR). These approaches usually follow a supervised learning scheme, where a neural network is trained on the annotations of some dataset. For many MIR tasks, however, such annotations are costly to obtain. Recent work has investigated alternatives that require little or no annotations and enable training on large, unannotated datasets.

For certain music genres, there are datasets that contain several versions (i. e., recorded performances) of a musical work. For example, the same classical symphony or concerto can be performed by different orchestras, and several commercial recordings are often available. On such datasets, automatic music synchronization techniques can be used to find alignments between different versions of a work, requiring minimal annotation effort [1, 2].

In this paper, we introduce a conceptually novel approach to audio representation learning that exploits cross-version datasets, thus requiring only alignments between versions and no further human annotations. Our approach aims at learning embeddings of audio excerpts such that musically corresponding excerpts in different versions are mapped to close points in the embedding space (Figure 1).

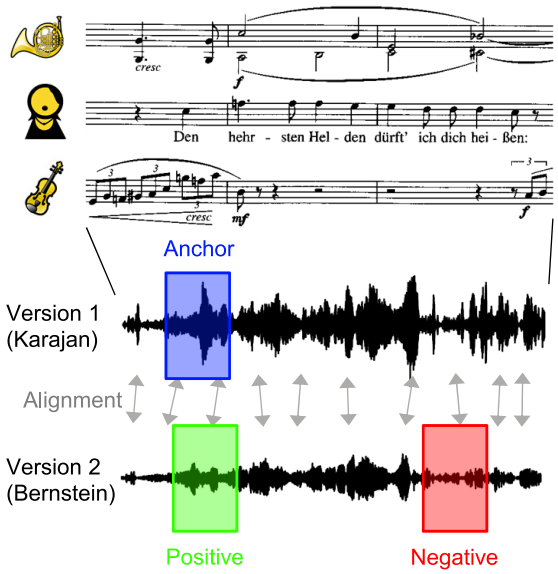


Figure 1: Visualization of our cross-version approach to representation learning for orchestral music. An anchor (blue) excerpt is selected from a music recording. The positive (green) and negative (red) excerpts are chosen from a different version of the same musical piece. For this, an alignment between versions is needed (gray arrows).

There are several musical aspects that stay roughly constant across most versions, e. g., pitches, harmonies or rhythm. For orchestral music, aspects of instrumentation (i. e., active instruments or instrument families) are another such property. Instrumentation represents a challenging MIR scenario given the complexity of instrument taxonomies and the difficulty of annotating instrument activity in orchestral music. In our experiments on a dataset of complex orchestral music, we show qualitatively and quantitatively that—by utilizing the correspondences between different versions of a musical section—our proposed representation learning technique is better at capturing aspects of instrumentation and instrument texture compared to approaches that do not exploit cross-version information.

© M. Krause, C. Weiß, and M. Müller. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).
Attribution: M. Krause, C. Weiß, and M. Müller, "A Cross-Version Approach to Audio Representation Learning for Orchestral Music", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.*, Milan, Italy, 2023.

The remainder of the paper is structured as follows: Section 2 covers related work on music audio representation learning, cross-version analysis, and instrumentation in orchestral music. In Section 3, we introduce our proposed approach. In Section 4, we describe our experimental setup, including datasets, our model architecture, and baselines. Section 5 contains qualitative and quantitative results and Section 6 concludes the paper with a discussion of possible future work.

2. RELATED WORK

Several recent contributions have explored so-called self-supervised strategies for learning representations from unannotated music recordings. Often, in these studies, excerpts from a music recording that are in close proximity are considered as positive pairs (i. e., should be mapped to similar representations) whereas excerpts that are further apart (or from other recordings) are negative pairs (i. e., should be mapped to dissimilar representations). This idea is also illustrated in Figure 2. McCallum [3] originally considered this with the aim of learning features for music structure analysis. Wang et al. [4] had a similar use case but used a supervised learning approach. Several authors employed such a strategy for learning more general purpose representations [5–10], often applying additional augmentations. Apart from using temporal proximity, other papers on music representation learning exploit audio-visual or audio-text correspondences [11, 12], use classical features as training targets [13], exploit metadata [14], or investigate music generation models [15].

The approach proposed in this paper is conceptually different since we utilize cross-version datasets, rather than relying on temporal proximity alone. Such datasets contain several recorded versions of a musical work, which may vary in aspects related to musical interpretation, recording conditions, and timbral characteristics of the instruments used. These datasets have been exploited for expressive performance rendering [16] or improved harmonic analysis [17]. Cross-version datasets also allow for investigating model biases and overfitting effects in MIR models through different dataset splits [18]. To our knowledge, the only other work utilizing cross-version information for embedding learning is by Zalkow et al. [19], whose aim was to compress chromagram excerpts for efficient music retrieval. In contrast, we propose to learn representations based on spectrogram-like input features and investigate them for capturing instrument texture.

In the wider machine learning literature, representations are often learned by masking a part of an input and predicting the masked content [20, 21]. Other strategies utilize multi-modal datasets, e. g., containing text–image [22] or audio–text pairs [23].

Orchestral music has been explored in the context of source separation [24] or melody extraction [25]. The authors in [26] considered instrument family recognition for classical, monotimbral recordings using a supervised learning approach. Other recent papers on instrument activity detection in music recordings [27–29] have also considered DL-based, supervised learning approaches, but not within orchestral scenarios.

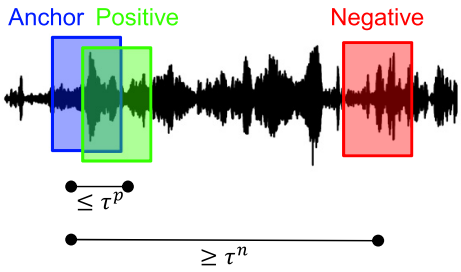


Figure 2: When forming triplets of audio excerpts, the anchor and positive/negative excerpts are chosen according to a maximum/minimum distance $\tau^{\mathrm{p}}/\tau^{\mathrm{n}}$.

3. CROSS-VERSION APPROACH TO AUDIO REPRESENTATION LEARNING

In this section, we formalize our proposed cross-version approach to representation learning. The key idea is to utilize correspondences between different versions (i. e., recorded performances played by different orchestras) of the same musical work. We aim to learn embeddings of audio excerpts such that the same musical section in different versions is represented by neighboring points in the embedding space and audio excerpts for unrelated musical sections are mapped to distant points in the embedding space. To this end, inspired by [19], we sample triplets of audio excerpts as in Figure 1, and apply a triplet loss for learning. Musical characteristics that stay roughly constant across different versions of an orchestral work include pitches and harmonies, as well as instrumentation. In later sections, we will analyze to what extent our approach captures pitches or aspects of instrumentation.

Single-Version Approach (SV)**.** We begin by formalizing a common approach to music representation learning that only utilizes temporal proximity inside a single version, see also Section 2 and Figure 2. Let $\mathcal{W}$ be a set of musical works and let V_w be the set of available versions for a work $w \in \mathcal{W}$. We first randomly select a work $w \in \mathcal{W}$ and some version of this work $v \in V_w$. Let T denote the length of v in seconds. We choose an anchor excerpt by uniformly sampling an anchor position $a \in [0, T]$ and extracting the excerpt $\mathbf{x}^{\mathrm{a}}$ of v that is centered around a. To obtain the positive and negative excerpts, we choose a position $p \in [0, T]$ for the positive excerpt $\mathbf{x}^{\mathrm{p}}$ of v such that $|a - p| \leq \tau^{\mathrm{p}}$. Thus, the positive excerpt is in temporal proximity of the anchor excerpt—up to a threshold of τ^{p} seconds—and is likely to correspond to a musically similar section. In the same way, we choose a position $n \in [0, T]$ for the negative excerpt $\mathbf{x}^{\mathrm{n}}$ of v such that $|a - n| \geq \tau^{\mathrm{n}}$. The negative excerpt is therefore a certain minimum distance of τ^{n} seconds away from the anchor position, likely corresponding to a musically dissimilar section. [1]

[1] Due to repetitions and other structural similarities, there may in fact be some musically related sections that are far apart temporally. In the majority of cases, however, the assumption underlying positive and negative sampling will hold [3].

Embedding Learning. We obtain embeddings by passing these excerpts through a neural network (described in Section 4.2), i. e.:

$$\mathbf{Y} = (\mathbf{y}^{\mathrm{a}}, \mathbf{y}^{\mathrm{p}}, \mathbf{y}^{\mathrm{n}}) = (f(\mathbf{x}^{\mathrm{a}}), f(\mathbf{x}^{\mathrm{p}}), f(\mathbf{x}^{\mathrm{n}}))\,, \quad (1)$$

where f is a neural network that embeds an audio excerpt $\mathbf{x}$ into an embedding vector $\mathbf{y}$. Using this triplet, we can apply a standard triplet loss [30] such as:

$$\mathcal{L}(\mathbf{Y}) = \max\left(0, \|\mathbf{y}^{\mathrm{a}} - \mathbf{y}^{\mathrm{p}}\|_2^2 - \|\mathbf{y}^{\mathrm{a}} - \mathbf{y}^{\mathrm{n}}\|_2^2 + \alpha\right), \quad (2)$$

where $\alpha \in \mathbb{R}_{\geq 0}$ describes the desired minimum margin between the distance of embeddings for anchor and positive versus the distance of embeddings for anchor and negative.

Cross-Version Approach (CV)**.** For our proposed cross-version approach, we sample triplets in a different fashion. Since we utilize multiple versions per work, we now require $|V_w| \geq 2$. To form a triplet of excerpts, we randomly select some version $v_1 \in V_w$ of a work $w \in \mathcal{W}$. We then sample an anchor position $a_1 \in [0, T_1]$, where T_1 is the length of v_1 in seconds, and extract the corresponding excerpt $\mathbf{x}^{\mathrm{a}}$ of v_1. To obtain the positive and negative excerpts, we randomly select another version $v_2 \in V_w \setminus \{v_1\}$ of w. As before, let T_2 denote the length of v_2 in seconds. We can find the position $a_2 \in [0, T_2]$ in v_2 corresponding to the same musical position as the anchor a_1 in v_1 using music alignment techniques. With this, we choose a position $p \in [0, T_2]$ for the positive excerpt $\mathbf{x}^{\mathrm{p}}$ of v_2 such that $|a_2 - p| \leq \tau^{\mathrm{p}}$. Thus, the positive excerpt corresponds to the same musical section as the anchor, up to some tolerance of τ^{p} seconds (in addition to alignment inaccuracies). Similarly, we sample $n \in [0, T_2]$ (with $|a_2 - n| \geq \tau^{\mathrm{n}}$) and extract $\mathbf{x}^{\mathrm{n}}$. Note that only $\mathbf{x}^{\mathrm{a}}$ is an excerpt of the first version v_1, whereas both $\mathbf{x}^{\mathrm{p}}$ and $\mathbf{x}^{\mathrm{n}}$ are excerpts of the second version v_2. As before, we construct a triplet $\mathbf{Y}$ using these excerpts and apply a standard triplet loss.

4. EXPERIMENTAL SETUP

4.1 Dataset and Splits

To show the potential of our representation learning technique, we construct a cross-version dataset of commercial symphonic and opera music recordings, illustrated in Table 1. Our dataset contains an act from an opera (the first act from Richard Wagner's "Die Walküre") as well as orchestral pieces by Beethoven, Dvorak and Tschaikowsky. Counting each movement as an individual work, the dataset contains eleven different works in total. We choose the first movement of the Beethoven Symphony, the fourth movement of the Dvorak Symphony and the third movement of the Tschaikowsky Concerto for testing. Since we do not have multiple opera acts that could be split into train and test, we choose an excerpt of the Wagner opera act (measures 697 to 955, corresponding to around twelve minutes), omit this excerpt during training, and use it for testing. We further ensure that the train and test set contain different versions. By splitting our dataset in this fashion, we aim to avoid overfitting to specific musical compositions or recording conditions (the latter is also referred to as "album effect" [31]).

Composer	Work	Versions Num.	Versions Avg. Duration
Wagner	Die Walküre, Act 1	8	1 h
Beethoven	Symph. 3, Mvmts. 1–4	6	45 min
Dvorak	Symph. 9, Mvmts. 1, 2, 4	6	40 min
Tschaikowsky	Violin Concerto, Mvmts. 1–3	6	35 min
Total duration			20 h

Table 1: Our cross-version dataset containing several commercial recordings of different orchestral and opera compositions.

For the cross-version approach CV, we obtain an alignment between versions of the same work using state-of-the-art music synchronization techniques involving chroma onset features and multi-scale alignment [2]. For some experiments, we also require pitch-class and instrument activity annotations for our dataset. To this end, we manually encoded a score representation of "Die Walküre" and obtained further scores from the Mutopia project. [2] Again, we use music synchronization techniques to align score to audio and create the annotations.

4.2 Model

We implement all representation learning approaches using a convolutional neural network that takes a harmonic CQT representation (HCQT, [32]) of an audio excerpt as input and outputs a corresponding embedding vector. The HCQT input consists of 201 frames (at a frame rate of 43 Hz, i. e., roughly 4.7 seconds), three bins per semitone from C1 to B7 (leading to 252 bins), and five harmonics (with frequency multiples of $[0.5, 1, 2, 3, 4]$).

The model architecture is adapted from [33] and receives an HCQT input patch, processes it through several convolution and max-pooling layers, and outputs a single ℓ_2-normalized vector (length 128) per input. We take this output as the embedding vector for the center frame of the input patch. In total, the architecture has roughly 1.5 million learnable parameters. We train our network for 200 epochs (with 16 000 triples randomly sampled per epoch) using the Adam optimizer with a learning rate of 0.002. In the interest of reproducibility, we release code and trained models for our approach. [3]

In line with previous studies on audio representation learning [5–7], we apply a number of augmentations to excerpts during training, including time scaling, pitch shifting, random masking, adding noise and applying random equalization. For all experiments, we set $\tau^{\mathrm{p}} = 0.2\,\mathrm{s}$. With this, the maximal distance between anchor and positive excerpt is in the same order of magnitude as the typical alignment inaccuracy between versions. We further set

[2] https://www.mutopiaproject.org/

[3] https://www.audiolabs-erlangen.de/resources/MIR/2023-ISMIR-CrossVersionLearning

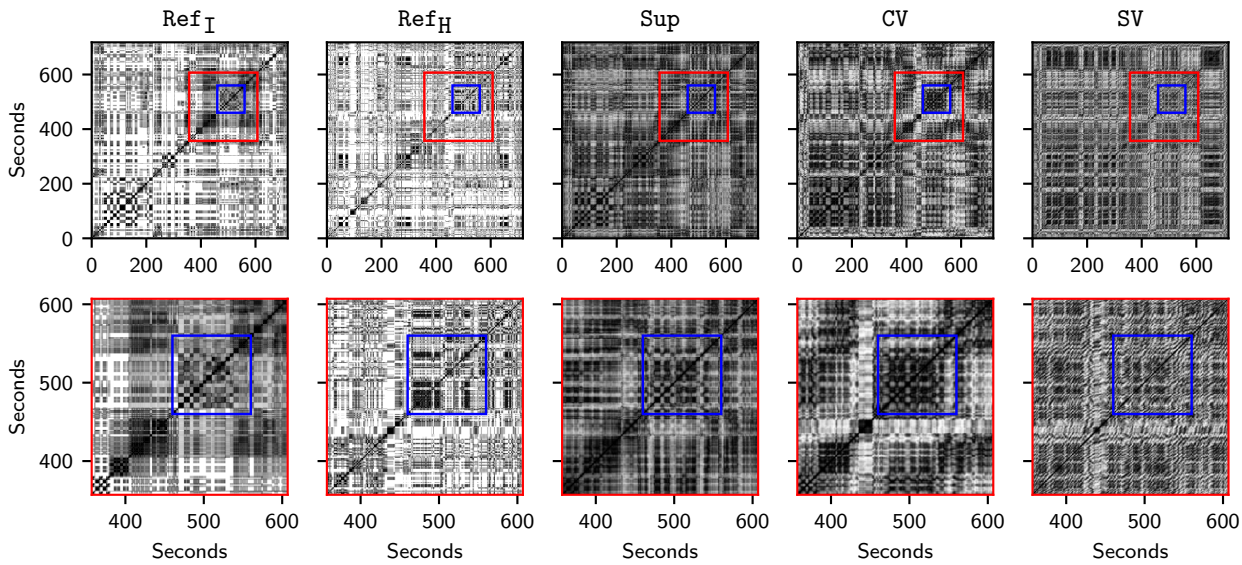


Figure 3: Self-similarity matrices constructed from instrument annotations ($\texttt{Ref}_\texttt{I}$) and pitch-class annotations ($\texttt{Ref}_\texttt{H}$), or obtained with a supervised learning system (Sup), the proposed cross-version approach (CV), and a baseline that does not incorporate cross-version information (SV). The lower row shows the sections highlighted in red from above.

$\tau^{\mathrm{n}} = 10.0\,\mathrm{s}$ and $\alpha = 1.0$. We found that results are stable for a broad range of settings of these parameters.

4.3 Baselines

To investigate the musical properties captured by the representation learning approaches CV and SV, we compare them to several optimistic baselines: First, we extract traditional music audio features. We use mel-frequency cepstral coefficients (MFCC), which are known to capture aspects of instrumentation [34], and Chroma features, which contain the dominant pitch-classes in the recording. Here, our goal is not to outperform MFCC or Chroma, but to compare them to our learned representations. If our learning approaches capture instrumentation, we expect them to behave similar to MFCCs. Likewise, in case they contain pitch-class information, we expect them to perform like Chroma features.

Second, we consider a supervised learning approach Sup where we train a model on instrument activity annotations and use its hidden representations as features. For this, we utilize the same model architecture as for CV and SV and only add a final dense layer with a number of outputs equal to the number of instruments to detect. Rather than using the triplet loss from Section 3, we train this approach by applying a sigmoid activation and binary cross-entropy loss. Note that in contrast to CV and SV, the Sup approach requires instrument activity annotations for the recordings in the training set.

5. RESULTS

5.1 Feature Analysis using Self-Similarity

In order to visualize and compare the representations learned by different techniques, we employ self-similarity matrices. Such matrices are commonly used for music structure analysis and allow for visualizing structures based on repetition and homogeneity in feature sequences [1]. Here, we use them to analyze our learned representations without the need for additional fine-tuning. This also allows us to directly compare approaches trained with a fixed instrument vocabulary (Sup) to others that are not informed about instruments. We provide an alternative evaluation in Section 5.4.

Given a sequence $X = (x_1, \ldots, x_N)$ of (learned) representations of N audio frames, we construct the corresponding self-similarity matrix $S \in \mathbb{R}^{N \times N}$ as follows. We first normalize all representations with respect to the ℓ_2-norm, yielding $\tilde{X} = (\tilde{x}_1, \ldots, \tilde{x}_N)$. We then compute $S(n,m) := \langle \tilde{x}_n, \tilde{x}_m \rangle$ for $n, m \in [1:N]$. Thus, S contains the cosine similarities between elements of X, and all its entries lie in the interval $[-1, 1]$. By definition, all entries on the diagonal of S are equal to 1. In addition, repeated subsequences appear as path-like structures and homogeneous segments appear as block-like structures, see also [1].

We compare the self-similarity matrices obtained from learned representations to matrices created using reference annotations. First, we represent an instrument activity annotation as a sequence of multi-hot binary vectors (indicating the presence of instruments in different frames). By normalizing and computing the dot product as before, we obtain a matrix corresponding to instrument texture, where blocks indicate segments with similar instrumentation. We will refer to this matrix using the shorthand $\texttt{Ref}_\texttt{I}$. For example, the start of the middle measure in Figure 1 would be encoded as a vector $(1,1,1)^\top$, i. e., all instruments are active, and the end of that measure would be encoded as $(1,1,0)^\top$, i. e., only horn and soprano are active. After normalization, the dot product of these vectors is 0.82,

indicating similar instrumentation. Analogously, we construct another matrix $\mathtt{Ref}_{\mathtt{H}}$ from a sequence of pitch-class annotations. This matrix captures regions with similar harmonies and pitches.

5.2 Qualitative Results

Figure 3 shows several self-similarity matrices obtained through reference annotations or by different representation learning approaches. The excerpt shown in the upper row is the test excerpt from "Die Walküre" (similar results are obtained on other inputs). The lower row shows magnified sections from above. Darker color indicates higher similarity.

In the $\mathtt{Ref}_{\mathtt{I}}$ matrix, arising from instrument annotations as explained in Section 5.1, one can observe many block and checkerboard-like structures. For example, from seconds 460 to 560, different combinations of woodwind instruments are playing together, creating block and checkerboard-like patterns (highlighted in blue). White areas indicate $S(n,m) = 0$, i. e., no common instruments are playing. The matrix $\mathtt{Ref}_{\mathtt{H}}$, on the other hand, indicates harmonic similarities which are mostly distinct from the instrument similarities in $\mathtt{Ref}_{\mathtt{I}}$.

For the Sup system, many of the patterns in $\mathtt{Ref}_{\mathtt{I}}$ are replicated, albeit with less detail. This is expected, since this system has been trained on the same kind of annotations that have been used to create $\mathtt{Ref}_{\mathtt{I}}$. Interestingly, many of the patterns present in the $\mathtt{Ref}_{\mathtt{I}}$ and Sup matrices also appear for the proposed approach CV, which has not been trained using instrument annotations. In particular, the checkerboard pattern starting at second 460 is captured by CV, as well as many block structures.

There are fewer similarities between CV and $\mathtt{Ref}_{\mathtt{H}}$, indicating that the CV representations are more likely to capture instrumentation rather than pitch-class content. This behavior is encouraged by our augmentation strategy, where we randomly pitch-shift the anchor, positive and negative excerpts.

The matrix obtained through the SV approach is blurry and, unlike the results for CV, fails to capture many of the checkerboard-like patterns present in $\mathtt{Ref}_{\mathtt{I}}$. The example suggests that exploiting cross-version information during training is important for capturing aspects of instrumentation in learned representations.

5.3 Quantitative Results

In order to quantify the correlation between our learned representations and instrument texture, we now apply a procedure for detecting the boundaries of block-like structures in self-similarity matrices. We then compare block boundaries estimated on $\mathtt{Ref}_{\mathtt{I}}$ with boundaries from all other matrices. Such procedures are often used in the context of music structure analysis [1, 35].

To detect block boundaries, we first correlate a self-similarity matrix with a checkerboard kernel along the main diagonal, as proposed in [35]. From this, we obtain a novelty curve. We then apply a peak picking procedure using local thresholding on this novelty curve, yielding sparse positions of detected block structures. We do this for all approaches and reference matrices. We finally compare—with a tolerance of up to three seconds—the detected boundaries for all approaches to those of $\mathtt{Ref}_{\mathtt{I}}$, yielding a boundary F-measure. By adjusting the size of the checkerboard kernel in this procedure, we can identify changes of instrument texture on short or larger time scales. For more details on the boundary detection, peak picking, and evaluation procedure, we refer to [1].

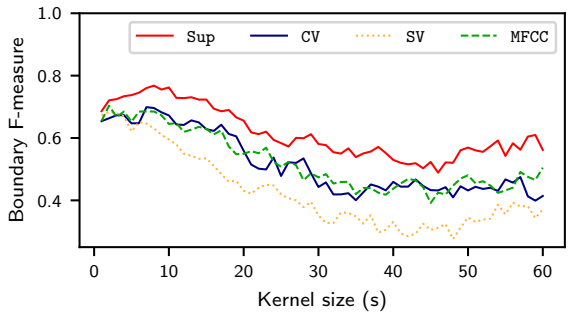


Figure 4: Results for different representation learning approaches when comparing estimated structure boundaries to boundaries from $\mathtt{Ref}_{\mathtt{I}}$.

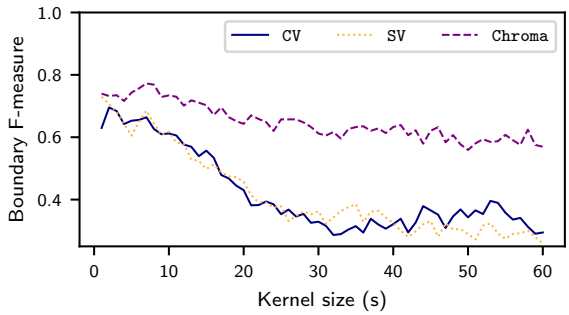


Figure 5: Results for comparing with $\mathtt{Ref}_{\mathtt{H}}$.

Figure 4 shows the results of our quantitative evaluation for different sizes of the checkerboard kernel. The F-measures given are averaged over all recordings in the test dataset. We observe that the supervised approach is best at capturing instrument texture (as encoded by $\mathtt{Ref}_{\mathtt{I}}$) compared to all others, with the highest F-measure of 0.77 for a kernel of eight seconds. CV and MFCC perform roughly on par. This is surprising, since CV is trained without any instrument annotations, while MFCC is known to capture instrumentation. Results for SV deteriorate with larger kernel sizes, dropping to as low as 0.28 F-measure for a kernel of 48 seconds. The proposed approach CV is better at capturing instrument texture than the alternative SV that does not utilize cross-version information.

To examine whether our representations capture information related with harmonies and pitches played, we perform the same evaluation procedure with boundaries from $\mathtt{Ref}_{\mathtt{H}}$ (see Figure 5). We obtain low F-measures for both CV and SV (dropping below 0.4 for kernel sizes above 20 seconds for both approaches). In particular, while we observe an advantage of CV over SV for capturing instrumentation, there is no such advantage with regard to

Scenario	AP	AUC	Micro Avg. F1	Micro Avg. S	Macro Avg. F1	Macro Avg. S
MFCC	0.777	0.780	0.600	0.890	0.450	0.847
SV	0.708	0.735	0.590	0.871	0.407	0.820
CV	0.753	0.795	0.657	0.872	0.514	0.835
Sup	0.838	0.881	0.772	0.894	0.714	0.874

Table 2: Results for different representation learning approaches when performing instrument classification.

Scenario	AP	AUC	Micro Avg. F1	Micro Avg. S	Macro Avg. F1	Macro Avg. S
Chroma	0.802	0.854	0.591	0.964	0.586	0.963
SV	0.427	0.568	0.001	1.000	0.001	1.000
CV	0.430	0.584	0.021	0.994	0.018	0.994
Sup	0.457	0.612	0.137	0.959	0.122	0.958

Table 3: Results for pitch-class classification using the learned representations.

pitch-classes. Additionally, standard Chroma features are clearly superior at capturing the structures in $\mathrm{Ref}_{\mathrm{H}}$. We conclude that the representations learned by our proposed approach CV indeed contain information about instrument texture rather than pitch-classes and harmonies.

5.4 Feature Analysis Using Classification

To gain further insights into the information captured by our learned representations, we also perform an indirect evaluation as typically done in representation learning. Previous studies often rely on training small classifiers on top of learned representations to investigate their usefulness for different downstream tasks [5, 15]. In this section, we complement our self-similarity-based analysis with such a classification-based evaluation strategy.

To this end, we pass individual representation vectors through a small network of dense layers with 128, 64, and 32 hidden units followed by leaky ReLU activations, respectively. The final layer produces outputs for every instrument annotated in our dataset, followed by a sigmoid activation. For each representation learning technique, we train and evaluate such a network using the dataset split as described in Section 4.1. Concretely, we minimize the mean binary cross-entropy loss over all instrument classes on the training set, using stochastic gradient descent with a learning rate of 10^{-4} for 10 epochs. We finally evaluate the classification results on the test set using standard metrics, including ranking-based average precision (AP), mean area under the ROC curves (AUC), F-measure (F1), and specificity (S). For F1 and S, we threshold the predicted probabilities at 0.5 and compute both micro and macro averages of the evaluation scores, where the macro average is not affected by imbalance among instrument classes.

The results of this experiment are shown in Table 2. We observe similar trends as in our self-similarity-based evaluation. As expected, the supervised baseline again yields best results. Our proposed cross-version approach CV clearly outperforms the traditional SV across all metrics (e. g., AP $= 0.753$ as opposed to 0.708 for SV). Furthermore, CV even improves upon the optimistic MFCC baseline in terms of AUC and F-measure (e. g., micro F1 $= 0.657$ instead of 0.600 for MFCC). Finally, SV performs worse than MFCC. Overall, the representations learned by our proposed approach CV are more effective for instrument classification compared to the standard SV approach that does not utilize cross-version information.

We repeat this experiment using pitch-classes as the classification targets instead of instruments. Table 3 shows the results of the modified experiment, which are inline with our conclusions from previous sections. Standard Chroma features strongly outperform all learned representations on this task. We conclude that our proposed approach captures instrumentation rather than pitches.

6. CONCLUSION

In this paper, we described a novel audio representation learning approach for cross-version music data and investigated its application to orchestral music. Our approach utilizes the correspondences between different versions of the same musical work. We showed qualitatively and quantitatively that the representations learned by our approach capture aspects of instrumentation. We outperform a standard training strategy that relies on temporal proximity alone.

Our approach can be applied to any kind of cross-version music dataset where alignments between versions can be obtained using standard music synchronization techniques. Future work may apply our approach to other musical scenarios and larger datasets, explore more complex feature extraction networks, investigate alternatives to our triplet loss formulation, or apply the learned representations in the context of different downstream tasks (such as structure analysis). One may also study the impact of design choices such as τ^{p} and τ^{n}, the pitch shifting augmentation, or the number of versions used for training.

Acknowledgments: This work was supported by the German Research Foundation (DFG MU 2686/7-2, MU 2686/11-2). 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. The authors gratefully acknowledge the compute resources and support provided by the Erlangen Regional Computing Center (RRZE).

7. REFERENCES

[1] M. Müller, *Fundamentals of Music Processing – Using Python and Jupyter Notebooks*, 2nd ed. Springer Verlag, 2021.

[2] 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.

[3] 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)*, Brighton, UK, 2019, pp. 346–350.

[4] J. Wang, J. B. L. Smith, W. T. Lu, and X. Song, "Supervised metric learning for music structure features," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Online, 2021, pp. 730–737.

[5] J. Spijkervet and J. A. Burgoyne, "Contrastive learning of musical representations," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Online, 2021, pp. 673–681.

[6] C. Thomé, S. Piwell, and O. Utterbäck, "Musical audio similarity with self-supervised convolutional neural networks," in *Demos and Late Breaking News of the International Society for Music Information Retrieval Conference (ISMIR)*, Online, 2021.

[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)*, Toronto, Canada, 2021, pp. 3875–3879.

[8] M. Buisson, B. McFee, S. Essid, and H. C. Crayencour, "Learning multi-level representations for hierarchical music structure analysis," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Bengaluru, India, 2022.

[9] M. A. V. Vásquez and J. A. Burgoyne, "Tailed U-Net: Multi-scale music representation learning," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Bengaluru, India, 2022.

[10] 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 International Society for Music Information Retrieval Conference (ISMIR)*, Bengaluru, India, 2022.

[11] B. Li and A. Kumar, "Query by video: Cross-modal music retrieval," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Delft, The Netherlands, 2019, pp. 604–611.

[12] I. Manco, E. Benetos, E. Quinton, and G. Fazekas, "Learning music audio representations via weak language supervision," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Singapore, 2022, pp. 456–460.

[13] H. Wu, C. Kao, Q. Tang, M. Sun, B. McFee, J. P. Bello, and C. Wang, "Multi-task self-supervised pre-training for music classification," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Toronto, Canada, 2021, pp. 556–560.

[14] P. Alonso-Jiménez, X. Serra, and D. Bogdanov, "Music representation learning based on editorial metadata from discogs," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Bengaluru, India, 2022.

[15] R. Castellon, C. Donahue, and P. Liang, "Codified audio language modeling learns useful representations for music information retrieval," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Online, 2021, pp. 88–96.

[16] H. Zhang, J. Tang, S. R. M. 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.

[17] S. Ewert, M. Müller, V. Konz, D. Müllensiefen, and G. A. Wiggins, "Towards cross-version harmonic analysis of music," *IEEE Transactions on Multimedia*, vol. 14, no. 3-2, pp. 770–782, 2012.

[18] H. Schreiber, C. Weiß, and M. Müller, "Local key estimation in classical music audio recordings: A cross-version study on Schubert's Winterreise," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Barcelona, Spain, 2020, pp. 501–505.

[19] F. Zalkow and M. Müller, "Learning low-dimensional embeddings of audio shingles for cross-version retrieval of classical music," *Applied Sciences*, vol. 10, no. 1, 2020.

[20] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. B. Girshick, "Masked autoencoders are scalable vision learners," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, LA, USA, 2022, pp. 15 979–15 988.

[21] W. Hsu, B. Bolte, Y. 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.

[22] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," in *Proceedings of the International Conference on Machine Learning (ICML)*, Virtual, 2021, pp. 8748–8763.

[23] A. Guzhov, F. Raue, J. Hees, and A. Dengel, "AudioCLIP: Extending clip to image, text and audio," in *Proceedings of the IEEE International Conference on*

Acoustics, Speech, and Signal Processing (ICASSP), Singapore, 2022, pp. 976–980.

[24] M. Miron, J. J. Carabias-Orti, J. J. Bosch, E. Gómez, and J. Janer, "Score-informed source separation for multichannel orchestral recordings," *Journal of Electrical and Computer Engineering*, vol. 2016, no. 8363507, 2016.

[25] Z. Tang and D. A. A. Black, "Melody extraction from polyphonic audio of Western opera: A method based on detection of the singer's formant," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Taipei, Taiwan, October 2014, pp. 161–166.

[26] M. Taenzer, J. Abeßer, S. I. Mimilakis, C. Weiß, H. Lukashevich, and M. Müller, "Investigating CNN-based instrument family recognition for Western classical music recordings," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Delft, The Netherlands, 2019, pp. 612–619.

[27] Y.-N. Hung and Y.-H. Yang, "Frame-level instrument recognition by timbre and pitch," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Paris, France, 2018, pp. 135–142.

[28] S. Gururani, C. Summers, and A. Lerch, "Instrument activity detection in polyphonic music using deep neural networks," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Paris, France, 2018, pp. 569–576.

[29] Y. Han, J. Kim, and K. Lee, "Deep convolutional neural networks for predominant instrument recognition in polyphonic music," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 1, pp. 208–221, 2017.

[30] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, 2015, pp. 815–823.

[31] A. Flexer, "A closer look on artist filters for musical genre classification," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Vienna, Austria, 2007, pp. 341–344.

[32] R. M. Bittner, B. McFee, J. Salamon, P. Li, and J. P. Bello, "Deep salience representations for F0 tracking in polyphonic music," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Suzhou, China, 2017, pp. 63–70.

[33] 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.

[34] H. Terasawa, M. Slaney, and J. Berger, "The thirteen colors of timbre," in *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, NY, USA, 2005, pp. 323–326.

[35] J. Foote, "Automatic audio segmentation using a measure of audio novelty," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, New York, NY, USA, 2000, pp. 452–455.

MUSIC SOURCE SEPARATION WITH MLP MIXING OF TIME, FREQUENCY, AND CHANNEL

Tomoyasu Nakano **Masataka Goto**
National Institute of Advanced Industrial Science and Technology (AIST), Japan
{t.nakano, m.goto}@aist.go.jp

ABSTRACT

This paper proposes a new music source separation (MSS) model based on an architecture with MLP-Mixer that leverages multilayer perceptrons (MLPs). Most of the recent MSS techniques are based on architectures with CNNs, RNNs, and attention-based transformers that take waveforms or complex spectrograms or both as inputs. For the growth of the research field, we believe it is important to study not only the current established methodologies but also diverse perspectives. Therefore, since the MLP-Mixer-based architecture has been reported to perform as well as or better than architectures with CNNs and transformers in the computer vision field despite the MLP's simple computation, we report a way to effectively apply such an architecture to MSS as a reusable insight. In this paper we propose a model called TFC-MLP, which is a variant of the MLP-Mixer architecture that preserves time-frequency positional relationships and mixes time, frequency, and channel dimensions separately, using complex spectrograms as input. The TFC-MLP was evaluated with source-to-distortion ratio (SDR) using the MUSDB18-HQ dataset. Experimental results showed that the proposed model can achieve competitive SDRs when compared with state-of-the-art MSS models.

1. INTRODUCTION

Music source separation (MSS) is the task of obtaining individual source signals — such as vocals, drums, and bass — from real music acoustic signals. This is an essential technique for various applications, including music information retrieval and music listening interfaces, where the characteristics of individual sound sources are analyzed and utilized. MSS has actually been used to add effects to individual source (instrument) sounds for music appreciation [1] and adjust their volume [1–4], to improve the cochlear implant user's musical experience by adjusting the volume of preferred instruments [5], to synthesize singing voices [6], to acquire feature expressions of singing voices [7], to identify singers [8], to achieve audio-to-lyrics alignment [9,10], to create music mashups [11], to separate sources for music education [12], and to estimate compatibility between vocals and accompaniment [13].

Currently, the mainstream approaches for MSS use deep neural networks [14, 15], and their performance is improving year by year. For their performance comparison to measure the improvement, MUSDB18 [16] had been used as the common standard data set for the four target sound sources (Vocals, Drums, Bass, and Other). Then MUSDB18-HQ [17], an extended frequency bandwidth version of MUSDB18, was released and has been used for recent evaluations.

As for the current state-of-the-art MSS models, the top-ranked models [18, 19] in the Music Demixing (MDX) Challenge 2021 [20] and the two models presented in two arXiv papers in 2022 [21, 22] have an average SDR (source-to-distortion ratio) over 7 dB for the four sources when using MUSDB18-HQ as training, validation, and test data. These models are explained in the next section.

Such deep MSS models can be classified in terms of the type of input and output used for separation and the type of architecture. The input and output of the models are selected from waveforms, amplitude spectrograms, complex spectrograms, phase spectrograms, etc. The architecture of the models is mainly selected from ResNet, DenseNet, U-Net, and Transformer and is used with layers of Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs). Simpler architectures based on multilayer perceptrons (MLPs), however, have not been used in state-of-the-art MSS models.

Meanwhile, in the field of computer vision, high-performance architectures based on MLPs have recently been proposed and reported to perform as well as or better than architectures using CNNs or transformers [23, 24]. In addition, those simpler MLP-centric architectures have also been applied in audio-related research, with cases of singing voice synthesis [25] and speech enhancement [26]. However, such MLP-centric architectures have not been applied in the MSS domain, though fully connected layers and MLPs have been used for linear transformation such as embedding and expansion. Since we believe that new perspectives, in addition to the development of established methodologies, are important for the advancement of the research field, this paper investigates how MLP-based architectures can be effectively leveraged for MSS.

As such a modern MLP-centric high-performance architecture, Tolstikhin *et al.* proposed the *MLP-Mixer* [23] that applies MLPs to a $T_p \times C_p$ matrix to estimate the

© T. Nakano and M. Goto. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** T. Nakano and M. Goto, "Music source separation with MLP mixing of time, frequency, and channel", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

Table 1. SDRs in MUSDB18-HQ for the state-of-the-art models and our proposed TFC-MLP. The "Avg." column means the average of the SDR results for the four sources. The "Per-source" column means that the per-source adjustment/tuning has been implemented. The "Extra" column indicates the number of songs added as extra training data, and † means that only mixed sounds were added. Models with "*" are evaluated in MUSDB18 [16], and SDRs for models with "**" are recalculated in MUSDB18-HQ using the pretrained model. A **bold** font indicates the maximum value at each source.

Model			Test SDR in dB				
Name	Per-source	Extra	Avg.	Vocals	Drums	Bass	Other
KUIELab-MDX-Net (w/o Demucs) [18]*	✓		7.28	8.91	7.07	7.33	5.81
TFC-MLP (ours)			7.30	8.91	7.18	6.96	6.14
KUIELab-MDX-Net [18]**	✓		7.48	8.97	7.20	7.83	5.90
Hybrid Transformer Demucs [22]			7.52	7.93	7.94	8.48	5.72
Hybrid Demucs [19]			7.64	8.35	8.12	8.43	5.65
Band-Split RNN [21]	✓		8.24	10.01	9.01	7.22	6.70
TFC-MLP (ours)		120	7.78	9.68	7.75	7.23	6.46
Hybrid Demucs [19]		800	8.34	8.75	9.31	9.13	6.18
Hybrid Transformer Demucs [22]		150	8.49	8.56	9.51	9.76	6.13
Hybrid Transformer Demucs [22]		800	8.80	8.93	10.05	9.78	6.42
Band-Split RNN [21]	✓	1750†	8.97	**10.47**	10.15	8.16	**7.08**
Hybrid Transformer Demucs [22]	✓	800	9.00	9.20	10.08	10.39	6.32
Sparse HT Demucs [22]	✓	800	**9.27**	9.37	**10.83**	**10.47**	6.41

class of an image. The matrix is obtained by dividing the image into patches and embedding each patch into a T_p-dimensional vector, which is called a token, and the number of tokens is the channel dimension C_p. Here, a *token-mixing MLP*, a full connection within an individual token, and a *channel-mixing MLP*, a full connection between tokens (*i.e.,* channel direction), are applied alternately. Given sufficient training data, MLP-Mixer was shown to perform as well as or better than CNNs or transformers.

By extending this MLP-Mixer, for the image reconstruction task, Mansour *et al.* proposed the *Image-to-Image Mixer* [24] that performs better when trained with fewer images than the original MLP-Mixer. The Image-to-Image Mixer transforms images as a 3D tensor ($W \times H \times C$) instead of the 2D matrix ($T_p \times C_p$) and preserves the relative positions of patches to induce a bias towards natural images. In other words, the token-mixing MLP and the channel-mixing MLP are split into three processes of *width-mixing MLP*, *height-mixing MLP* and *channel-mixing MLP*. This also keeps the total number of trainable parameters low, since the size of each dimension of the 2D matrix (*i.e.,* T_p and C_p) obtained by transformation from the 3D tensor is relatively large.

In investigating such MLP-based architectures in the MSS domain, we decided to use a complex spectrogram, which is a reasonable representation for MSS, as the input. Since the size of its complex time-frequency representation is larger than the size of typical images in the computer vision domain, we apply the memory-efficient Image-to-Image Mixer to MSS and report its experimental results.

2. RELATED WORK

As described in Section 1, the state-of-the-art models [18, 19, 21, 22] in the MSS study show that the average SDR score for the four source separations exceeds 7 dB in the evaluation using MUSDB18-HQ. The SDRs for each of the four sources based on these models are shown in Table 1.

KUIELab-MDX-Net was proposed by Kim *et al.* [18]. It is an architecture that combines an extended version of TFC-TDF-U-Net [27], which separates in the time-frequency domain (*i.e.,* complex spectrogram), and Demucs [28], which separates in the time domain (*i.e.,* waveform). Each source signal i separated by those subnetworks is mixed using source-dependent weights w_i. Specifically, w_i was set to 0.5, 0.5, 0.7, and 0.9 for bass, drums, other, and vocals in MDX Challenge 2021 [20] [1]. In KUIELab-MDX-Net, to improve performance, a mechanism called *Mixer* was used to remix the music acoustic signal with the separated signal by using the 1x1 convolution, and different FFT frame sizes were used for each sound source by applying a frequency cut-off trick [20].

TFC-TDF-U-Net used in the KUIELab-MDX-Net is a variant of U-Net architecture that combines the time-frequency convolutions (TFC) block with the time-distributed fully-connected networks (TDF) block. Here, TFC is a dense block of 2D CNNs, and TDF is a block that extracts nonlocal features along the frequency axis, such as correlations between harmonics, by fully connecting the entire frequency range of a single frame of the spectrogram. TDF was inspired by the Frequency Transformation Block (FTB) proposed by Yin *et al.* [29] and was introduced specifically to help separate singing voices [27]. As the other component of KUIELab-MDX-Net, Demucs [28] is a U-Net encoder/decoder structure with waveforms as input and BiLSTM applied to the innermost layer between the encoder and decoder to provide long-range context.

Hybrid Demucs was proposed by Défossez *et al.* [19]. It is a bi-U-Net encoder-decoder model that combines 1D convolution in the time domain and along the frequency

[1] `https://github.com/kuielab/mdx-net/blob/Leaderboard_A/README_SUBMISSION.md`

axis in the complex time-frequency domain. BiLSTM and local attention were used in the innermost layer, and residual branches, group normalization [30] and GELU were introduced. In addition, better generalization and stability were achieved by penalizing the largest singular value in each layer [31], and overall performance was improved by bagging multiple models.

Band-Split RNN was proposed by Luo *et al.* [21]. It is a state-of-the-art spectrogram-based model that uses complex spectrograms as input and output. By splitting the complex spectrogram input into subbands specifically designed for each sound source, the intrinsic properties and patterns of each source signal are utilized. For a 3D tensor representing the time dimension, frequency dimension, and subband dimension, similar to the Dual-path RNN [32], a sequence-level RNN is first applied across the time dimension, then a band-level RNN is applied across the band dimension. In fact, for Vocals, results showed that the modified utterance-level SDR can be improved by more than 2 dB by setting the bandwidth appropriately. In addition, semi-supervised fine tuning was performed by using pseudo-targets separated from the mixtures using a pretraining model in order to make effective use of songs with only mixtures (1750 songs). The Band-Split RNN currently achieves a state-of-the-art SDR of 8.24 dB in the evaluation using only MUSDB18-HQ as training data.

Hybrid Transformer Demucs was proposed by Rouard *et al.* [22]. The innermost layer, called the bottleneck, of the Hybrid Demucs [19] is replaced by a cross-domain transformer encoder (*i.e.,* time domain and time-frequency domain) that uses self-attention within each domain and cross-attention across domains. Compared to the Hybrid Demucs, the performance is lower when trained with MUSDB18-HQ only, but the SDR was 0.46 dB better when 800 additional training songs were used. Sparse HT Demucs [22], which extended the receptive field using a sparse attention kernel, achieved a state-of-the-art result of 9.27 dB SDR by fine-tuning for each source.

As described above, the MLP-centric architecture has never been applied in state-of-the-art MSS models where the average SDR exceeds 7 dB on MUSDB18-HQ.

3. TFC-MLP

This paper proposes *Time-Frequency-Channel-MLP (TFC-MLP)*, which is a model that leverages the Image-to-Image Mixer architecture [24] to separate music sources using a complex spectrogram as input. This is realized by replacing the height, width, and color (RGB) in the image with the time, frequency, and channel in the complex spectrogram, respectively. In other words, TFC-MLP has a structure that alternates mixing in the time, frequency, and channel dimensions. In this way, we expect to be able to take into account the nonlocal structure. Especially with respect to the frequency dimension, we expect to extract nonlocal relationships along the frequency axis, such as harmonic structures, by connecting the entire frequency range of the spectrogram, as in the FTB [29] and the TDF [27].

The process of the TFC-MLP model is shown in Figure 1. The complex spectrogram $\mathbf{X}_{\text{STFT}} \in \mathbb{C}^{F_0 \times T_0 \times 2}$ of a 2-channel (stereo) mixture signal is first converted to a 4-channel 3D tensor $\mathbf{X}_{\text{CaC}} \in \mathbb{R}^{F_0 \times T_0 \times 4}$ with the real and imaginary parts represented as channels, a.k.a. complex-as-channels (CaC) [27]. Here T_0 is a fixed length. Then, as with the MLP-Mixer and Image-to-Image Mixer as well as the Vision Transformer (ViT) [33], the patch embedding is first performed using C linear weights of size $P_T \times P_F \times 4$ (Figure 2). This reduces the frequency dimension F_0 to F_0/P_F and the time dimension T_0 to T_0/P_T, which reduces the matrix size and memory consumption in the following frequency-mixing MLP and time-mixing MLP. To compensate for the information loss due to decreasing the resolution in the time-frequency plane and the loss of continuity in the time-frequency direction, we increase the dimension in the channel direction.

The embedded tensor then passes through N MLP-Mixer layers. As shown in Figure 3, each MLP-Mixer layer mixes the tensor with the frequency dimension, then the time dimension, and finally the channel dimension. Such mixing is passed through an MLP consisting of a linear layer, a GELU nonlinearity, and another linear layer, and output without changing the tensor size. The dimension of the tensor at the input/output layer of the MLP is kept constant, and the dimension at the hidden layer is adjusted by multiplying it by a factor f depending on the input dimension. Skip connections and channel layer normalization have also been added to help with the optimization. Here, following the implementation of the Image-to-Image Mixer [24], skip connections are placed before and after the two mixing steps of frequency and time ("Type A"). In addition to that, a version with skip connection and layer normalization added before time-mixing MLP was also implemented ("Type B"). After mixing the time, frequency, and channel dimensions, patch expansion is used to restore the number of time and frequency dimensions for inverse transformation into waveforms and also to match the channel dimension to the number of separated sources.

4. EVALUATION

The proposed TFC-MLP was evaluated using the MUSDB18-HQ dataset [17]. 86 songs were used as training data, 14 songs as validation data, and 50 songs as test data. The music acoustic signals were stereo with a sampling frequency of 44.1 kHz, and four sound sources – "Vocals," "Drums," "Bass," and "Other" – were used for separation. Separation performance was evaluated by calculating SDR using the *museval* Python package [2]. As in most previous studies ([19, 21], etc.), the SDR of each source was calculated by taking the median values over all 1-second segments of each song to obtain the SDR of the track and then taking the median of all tracks.

4.1 Experimental setting

The proposed model was optimized using Adam [34] for the L1 loss between its separated signals and the ground-

[2] https://github.com/sigsep/sigsep-mus-eval

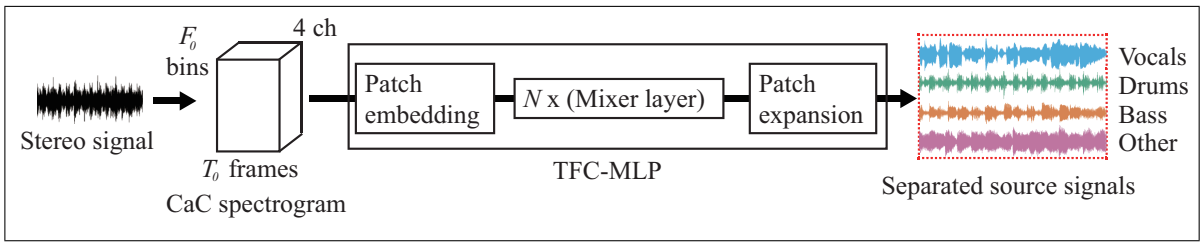


Figure 1. Overview of the TFC-MLP model.

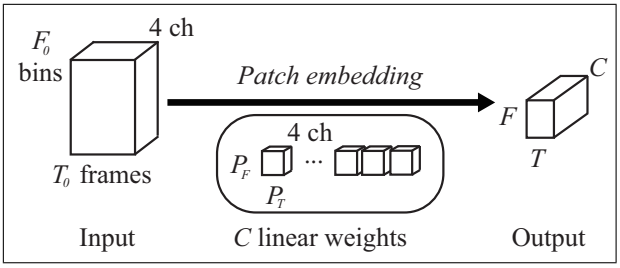


Figure 2. Patch embedding. After dividing the CaC tensor of size $F_0 \times T_0 \times 4$ into patches of size $P_F \times P_T \times 4$, each patch was linearly transformed into $1 \times 1 \times C$ to obtain a 3D tensor of size $F_0/P_F \times T_0/P_T \times C$ (*i.e.,* $F \times T \times C$). This can be implemented as a nonoverlapping CNN.

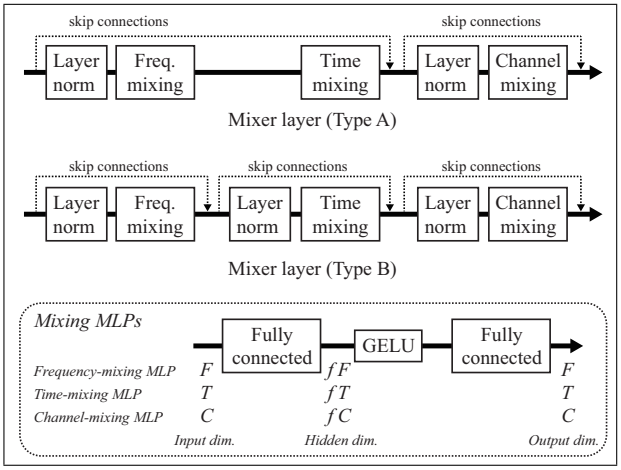


Figure 3. The Mixer layer contains one frequency-mixing MLP, one time-mixing MLP, and one channel-mixing MLP. Before each MLP, a transposition is performed to apply the MLP to the frequency dimension F, the time dimension T, and the channel dimension C. Transposition is also performed before every layer normalization to normalize along the channel dimension C.

truth source signals. The Adam parameters were set as no weight decay, learning rate 0.0003, $\beta_1 = 0.9$, and $\beta_2 = 0.999$. The waveform for calculating the CaC spectrogram to be input to the model was normalized so that the mean amplitude of the music acoustic signal was 0 and the standard deviation was 1. The training was distributed across multiple GPUs, with a batch size of 4 on each GPU. In training, we used data augmentation techniques [19], including pitch shift and tempo stretch, randomly swapping channels, random sign flip, random scaling of amplitude, and remixing of stems within one batch.

4.2 Base hyperparameters

Due to the many hyperparameters required for the building of the proposed TFC-MLP model, hyperparameters related to the short-time Fourier transform (STFT) were first determined through preliminary experiments. To obtain $\mathbf{X}_{\text{CaC}}$, the STFT frame size (*i.e.,* FFT size) was chosen as 4096, which had the highest SDR when compared among 512, 1024, 2048, and 4096. As Kim *et al.* [18] also stated, the performance tended to increase with larger FFT size. Therefore the frequency dimension F_0 is 2048. Related to the FFT size, the STFT hop size was investigated among 128, 256, 512, and 1024, and 1024 was selected. Also related to the FFT size and STFT hop size, the number of time frames T_0 in the complex spectrogram was selected as 128 from 32, 64, 128, 256, and 512.

The number of time frames T_0 of 128 based on an FFT size of 4096 and an STFT hop size of 1024 is input to the model, thus the waveform size required for this is equal to $T_w = 1024 \times 128 - 1$ (about 3 seconds). Therefore, it is necessary to train the model while cutting out an approximately 3-second segment of the waveform, and the shift size of 16384 (about 0.37 seconds) was used. Luo *et al.* [21] found that the shorter the shift, the better the modified utterance-level SDR, and they used 0.5 seconds. The above 16384 is short enough compared to 0.5 seconds and can be considered sufficient in this study.

As hyperparameters regarding the model, the dimensions C of the patch embedding were investigated by us. Specifically, 128 and 256 were investigated and 256 was selected. As hyperparameters related to the Mixer layer, 8 and 16 were investigated as the number of layers N and 16 was selected, and the parameter f for adjusting the number of dimensions of the hidden layer of each MLP was selected as 4 from 1, 2, and 4. Dropout in the MLP [25] and skip connections before and after the Mixer layer were also investigated, but they have not yielded better results.

4.3 Experiment

Using the set of hyperparameters determined in Section 4.2 as a basis, we report the SDR scores obtained under the following various conditions for the other hyperparameters.

- Investigation of the results of complex spectrogram loss [21].

Table 2. SDRs for different hyperparameters of the proposed model. The "Seed" column indicates random seeds. In the "Loss" column, "W" indicates that only waveform loss was used, and "W+S" indicates that complex spectral loss was added. The "Epoch" column indicates the number of epochs with the smallest validation loss and the specified number of epochs, where "*" means that validation loss was not considered. Note that "†" means the cases where the number of epochs with the smallest validation loss did not change when training beyond 200 epochs (*i.e.,* the same model was used for the evaluation). A **bold** font indicates the maximum value at each source, both without and with extra training songs.

TFC-MLP						Test SDR in dB				
Type	Seed	(P_F, P_T)	Loss	Epoch	Extra	Avg.	Vocals	Drums	Bass	Other
A	seed 1	$(4, 4)$	W	134† / 200		**7.30**	8.91	**7.18**	6.96	6.14
A	seed 2	$(4, 4)$	W	166† / 200		7.26	8.84	7.07	6.89	**6.22**
A	seed 3	$(4, 4)$	W	109 / 200		6.97	8.32	6.98	6.57	5.99
A	seed 3	$(4, 4)$	W	283 / 300		6.93	8.49	6.80	6.55	5.87
B	seed 1	$(4, 4)$	W	190 / 200		7.17	**8.92**	6.95	6.83	5.96
B	seed 2	$(4, 4)$	W	191 / 200		7.02	8.59	6.78	6.68	6.01
B	seed 3	$(4, 4)$	W	157 / 200		6.95	8.56	6.58	6.73	5.91
A	seed 1	$(2, 2)$	W	189 / 200		7.13	8.58	7.02	**6.99**	5.91
A	seed 1	$(1, 1)$	W	175 / 200		6.38	7.40	6.33	6.49	5.30
A	seed 1	$(4, 4)$	W+S	197 / 200		6.83	8.76	6.60	6.14	5.83
A	seed 1	$(4, 4)$	W+S	253 / 300		6.72	8.39	6.69	6.02	5.79
A	seed 1	$(4, 4)$	W	142 / 200	120	7.71	9.42	7.66	**7.37**	6.39
A	seed 1	$(4, 4)$	W	200* / 200	120	**7.78**	**9.68**	**7.75**	7.23	**6.46**

Table 3. The number of model parameters and the average Real Time Factor (RTF) value. The Hybrid Transformer Demucs is denoted as "HT Demucs". Column "GPU" shows the RTF with a single GPU, and column "CPU" shows the RTF under the condition without a GPU.

Model	GPU	CPU	Params.
Hybrid Demucs	0.14	1.87	83.9 M
HT Demucs	0.17	2.55	26.9 M
TFC-MLP	0.51	12.28	43.2 M

- Investigation of patch size (P_F, P_T) between $(4, 4)$, $(2, 2)$, and $(1, 1)$, using $(4, 4)$ as the basis, halving the FFT size and time dimension T for $(2, 2)$, and halving it further for $(1, 1)$. For $(1, 1)$, the STFT hop size was set to 512 since the FFT size is 1024.
- The number of epochs for training was set to 200 or 300, and models with the smallest validation loss within each epoch-condition were evaluated.
- The results of adding 120 full-length songs (sung in English) to the training data were evaluated.
- For the basic parameter condition, we trained three times with different random seeds.

In the test phase, the model with the smallest validation loss was used for evaluation in each training condition. The signal separated by the proposed model was divided into segments of fixed length T_w with shift width $T_w/4$, which were weighted overlap-added to obtain the final signal. In addition, the *shift trick* [28] was performed 10 times.

4.4 Results and discussions

In addition to the evaluation of SDRs using the MUSDB18-HQ test set, the number of parameters and the Real Time Factor (RTF) will also be discussed, as file size and the time required for separation may be important in some situations depending on how the MSS model is used.

4.4.1 SDRs

The experimental results are shown in Table 2. The highest SDR score, up to 7.30 dB, was obtained for the Type A model using the patch size of $(4, 4)$ and waveform L1 loss in addition to the base hyperparameters. This was not significantly higher than the state-of-the-art values shown in Table 1, but close performance was achieved.

Focusing on the results for each source with respect to our TFC-MLP, as shown in Table 1, the SDRs for Vocals and Other were 8.91 dB and 6.14 dB, respectively, which were higher than the 8.35 dB and 5.65 dB SDRs for the Hybrid Demucs and the 7.93 dB and 5.72 dB SDRs for the Hybrid Transformer Demucs. This model's SDR score for Other also exceeded the one obtained using KUIELab-MDX-Net, 5.90. Here, in comparison to the KUIELab-MDX-Net results without waveform information (*i.e.,* excluding processing by Demucs), it is possible that similar or better performance was obtained for Drums and Vocals, although an exact comparison cannot be made because the test data is different (*i.e.,* MUSDB18 was used). In comparison to the Band-Split RNN results, TFC-MLP could not yield a higher SDR for all sound sources. However, the current TFC-MLP does not include a source-specific framework, so addressing this issue is a future challenge.

As for the patch size, the FFT size and other conditions were different due to memory capacity. Therefore, although exact comparisons are not possible, the best results were obtained for $(4, 4)$ in the current results. However, additional study is needed for $(2, 2)$, since it gave results similar to those given by $(4, 4)$. As for complex spectrogram

loss, its SDR was slightly lower than that of all conditions using only waveform loss. Not only comparisons using the real and imaginary parts of the complex numbers, but also losses based on amplitude and phase could be considered.

We also showed that the performance of the models was further improved by using additional training data. As shown in Table 1, compared to the current world's best model Sparse HT Demucs with 800 songs added as the training data, we obtained competitive results with an SDR of 9.68 dB for the Vocals and 6.46 dB for the Others.

4.4.2 RTF and the number of parameters

As comparison, models were trained for each of the Hybrid Demucs and Hybrid Transformer Demucs. Based on the published source codes, a model parameter setting "80a68df8" [3] was used for Hybrid Demucs, and the default parameter setting was used for Hybrid Transformer Demucs. The same implementation for audio synthesis was used for all TFC-MLP, Hybrid Demucs, and Hybrid Transformer Demucs. We used the 50 songs in the MUSDB18-HQ test set to obtain the average of their RTFs.

Table 3 shows the results. The RTF values of TFC-MLP were slower than the other two models. TFC-MLP had an RTF lower than 1.0 (faster than real time) when a GPU was used, but the computation time was long without GPU. This could be due to the large size of the time-frequency spectrogram. As for the number of model parameters, TFC-MLP had more parameters than Hybrid Transformer Demucs, but fewer parameters than Hybrid Demucs.

4.4.3 Comparison with the state-of-the-art models

The proposed TFC-MLP model has some similarities to the state-of-the-art MSS models, which potentially have led to the competitive performance achieved.

- The frequency-mixing MLP is similar to the full connection of frequency dimensions in TDF [27] and the band-level RNN applied across band dimensions in Band-Split RNN [21].
- The time-mixing MLP is similar to the sequence-level RNNs applied across time dimensions in Band-Split RNN [21].
- Increasing the number of the channel dimensions through patch embedding and increasing the number of hidden layer dimensions in MLP are techniques that are usually used regardless of MSS. For MSS, the improvement is potentially related to the increase in channel dimensionality in the encoder part, such as Hybrid Transformer Demucs [22].
- The extra training data improved performance in the state-of-the-art models, and we confirmed the performance improvement with extra training data in TFC-MLP as well.

On the other hand, the following are included in the existing state-of-the-art MSS models but not currently included in our TFC-MLP. They have the potential to improve performance when applied in the future.

- As presented by Défossez *et al.* [19] and Kim *et al.* [18], a hybrid approach that also considers waveforms could improve performance.
- As Kim *et al.* [18], Luo *et al.* [21], and Rouard *et al.* [22] have shown, the introduction of source-specific techniques, such as band splitting, could improve separation performance.
- Deep learning techniques such as model selection methods (*e.g.,* exponential moving average), training stabilization (*e.g.,* singular value decomposition and sparsification), and the introduction of a learning rate scheduler could further improve performance.

Finally, to the best of our knowledge, there are no studies that mix the channel dimension with the time and frequency dimensions as in TFC-MLP. Such a mixer layer used in the TFC-MLP architecture has the advantage of reducing the overall memory usage compared to the original MLP-Mixer, just as the Image-to-Image Mixer reduced the memory usage. This reusable insight of mixing the channel dimension separately could be useful for other studies that have dealt with the time and frequency dimensions so far, but could be extended to the channel dimension.

4.4.4 Future directions

As future work, we plan to improve the performance of TFC-MLP by incorporating the ideas discussed above and further exploring more optimal hyperparameters. For example, increasing the FFT window size by utilizing frequency cut-off trick [20] is expected to improve the performance. Automatic optimization of hyperparameters could also be incorporated.

Future work will also include the visualization of the inside of TFC-MLP for analysis. We could visualize the linear weights used in the patch embedding by converting them back to complex numbers and then calculating their amplitudes. The visualized results could allow us to analyze what local patterns the model is focusing on. However, when we tried it, it was difficult to understand the behavior of the mixer layer due to the mixing of real and imaginary parts during the patch embedding. We are therefore interested in using the amplitude and phase (group delay) instead of the real and imaginary parts so that we can analyze the model in a more comprehensive way.

5. CONCLUSION

This paper has described a new MSS architecture called TFC-MLP that uses complex spectrograms as input. Our contributions are summarized as follows:

(1) We proposed a simpler MLP-centric MSS architecture that achieves competitive performance compared to state-of-the-art models.
(2) We reported on some hyperparameter searches that will be useful for other researchers exploring this type of architecture.
(3) We discussed the similarities and differences between the state-of-the-art models and TFC-MLP, and suggested directions for future research.

[3] `https://github.com/facebookresearch/demucs/blob/main/docs/training.md`

6. ACKNOWLEDGMENTS

This work was supported in part by JST CREST Grant Number JPMJCR20D4 and JSPS KAKENHI Grant Number JP21H04917, Japan.

7. REFERENCES

[1] J. Woodruff, B. Pardo, and R. Dannenberg, "Remixing stereo music with score-informed source separation," in *Proc. the 7th International Conference on Music Information Retrieval (ISMIR 2006)*, 2006, pp. 314–319.

[2] O. Gillet and G. Richard, "Extraction and remixing of drum tracks from polyphonic music signals," in *Proc. the 2005 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA 2005)*, 2005, pp. 315–318.

[3] K. Yoshii, M. Goto, and H. G. Okuno, "INTER:D: A drum sound equalizer for controlling volume and timbre of drums," in *Proc. the 2nd European Workshop on the Integration of Knowledge, Semantic and Digital Media Technologies (EWIMT 2005)*, 2005, pp. 205–212.

[4] K. Itoyama, M. Goto, K. Komatani, T. Ogata, and H. G. Okuno, "Instrument equalizer for query-by-example retrieval: Improving sound source separation based on integrated harmonic and inharmonic models," in *Proc. the 9th International Conference of Music Information Retrieval (ISMIR 2008)*, 2008, pp. 133–138.

[5] J. Pons, J. Janer, T. Rode, and W. Nogueira, "Remixing music using source separation algorithms to improve the musical experience of cochlear implant users," *The Journal of the Acoustical Society of America*, vol. 140, no. 6, pp. 4338–4349, 2016.

[6] Y. Ren, X. Tan, T. Qin, J. Luan, Z. Zhao, and T.-Y. Liu, "DeepSinger: Singing voice synthesis with data mined from the web," in *Proc. the 2020 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2020)*, 2020, pp. 1979–1989.

[7] H. Yakura, K. Watanabe, and M. Goto, "Self-supervised contrastive learning for singing voices," *IEEE/ACM Trans. on Audio Speech and Language Processing*, vol. 30, pp. 1614–1623, 2022.

[8] B. Sharma, R. K. Das, and H. Li, "On the importance of audio-source separation for singer identification in polyphonic music," in *Proc. the 20th Annual Conference of the International Speech Communication Association (Interspeech 2019)*, 2019, pp. 2020–2024.

[9] 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.

[10] L. Ou, X. Gu, and Y. Wang, "Transfer learning of wav2vec 2.0 for automatic lyric transcription," in *Proc. the 23rd International Society for Music Information Retrieval Conference (ISMIR 2022)*, 2022, pp. 187–195.

[11] J. Huang, J.-C. Wang, J. B. L. Smith, X. Song, and Y. Wang, "Modeling the compatibility of stem tracks to generate music mashups," in *Proc. the Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI-21)*, 2021, pp. 187–195.

[12] E. Cano, G. Schuller, and C. Dittmar, "Pitch-informed solo and accompaniment separation towards its use in music education applications," *EURASIP Journal on Advances in Signal Processing*, vol. 2014, no. 23, pp. 1–19, 2014.

[13] T. Nakatsuka, K. Watanabe, Y. Koyama, M. Hamasaki, M. Goto, and S. Morishima, "Vocal-accompaniment compatibility estimation using self-supervised and joint-embedding techniques," *IEEE Access*, vol. 9, pp. 101 994–102 003, 2021.

[14] 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. on Audio Speech and Language Processing*, vol. 26, no. 8, pp. 1307–1335, 2018.

[15] C. Gupta, H. Li, and M. Goto, "Deep learning approaches in topics of singing information processing," *IEEE/ACM Trans. on Audio Speech and Language Processing*, vol. 30, pp. 2422–2451, 2022.

[16] Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimilakis, and R. Bittner, "The MUSDB18 corpus for music separation," https://doi.org/10.5281/zenodo.1117372.

[17] ——, "MUSDB18-HQ - an uncompressed version of MUSDB18," https://doi.org/10.5281/zenodo.3338373.

[18] M. Kim, W. Choi, J. Chung, D. Lee, and S. Jung, "KUIELab-MDX-Net: A two-stream neural network for music demixing," in *Proc. Music Demixing Workshop 2021 (MDX 2021)*, 2021, pp. 1–7.

[19] A. Défossez, "Hybrid spectrogram and waveform source separation," in *Proc. Music Demixing Workshop 2021 (MDX 2021)*, 2021, pp. 1–11.

[20] 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," *Front. Sig. Proc.*, 2022.

[21] Y. Luo and J. Yu, "Music source separation with band-split rnn," *CoRR, arXiv:2209.15174*, pp. 1–10, 2022.

[22] S. Rouard, F. Massa, and A. Défossez, "Hybrid transformers for music source separation," *CoRR, arXiv:2211.08553*, pp. 1–5, 2022.

[23] I. O. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, A. Steiner, D. Keysers, J. Uszkoreit, M. Lucic, and A. Dosovitskiy, "MLP-Mixer: An all-MLP architecture for vision," in *Proc. the 35th Conference on Neural Information Processing Systems (NeurIPS 2021)*, 2021, pp. 24 261–24 272.

[24] Y. Mansour, K. Lin, and R. Heckel, "Image-to-Image MLP-Mixer for image reconstruction," *CoRR, arXiv:2202.02018*, pp. 1–15, 2022.

[25] J. Tae, H. Kim, and Y. Lee, "MLP Singer: Towards rapid parallel korean singing voice synthesis," in *Proc. the 2021 IEEE International Workshop on Machine Learning for Signal Processing (MLSP 2021)*, 2021, pp. 1–6.

[26] H. Song, M. Kim, and J. W. Shin, "Speech enhancement using mlp-based architecture with convolutional token mixing module and squeeze-and-excitation network," *IEEE Access*, vol. 10, pp. 119 283–119 289, 2022.

[27] W. Choi, M. Kim, J. Chung, D. Lee, and S. Jung, "Investigating U-Nets with various intermediate blocks for spectrogram-based singing voice separation," in *Proc. the 21st International Society for Music Information Retrieval Conference (ISMIR 2020)*, 2020, pp. 192–198.

[28] A. Défossez, N. Usunier, L. Bottou, and F. R. Bach, "Music source separation in the waveform domain," *CoRR, arXiv:1911.13254*, pp. 1–16, 2021.

[29] D. Yin, C. Luo, Z. Xiong, and W. Zeng, "PHASEN: A phase-and-harmonics-aware speech enhancement network," in *Proc. the The Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-20)*, 2020, pp. 9458–9465.

[30] Y. Wu and K. He, "Group normalization," in *Proc. the 15th European Conference on Computer Vision (ECCV 2018)*, 2018, pp. 3–19.

[31] Y. Yoshida and T. Miyato, "Spectral norm regularization for improving the generalizability of deep learning," *CoRR, arXiv:1705.10941*, pp. 1–12, 2017.

[32] Y. Luo, Z. Chen, and T. Yoshioka, "Dual-path RNN: efficient long sequence modeling for time-domain single-channel speech separation," in *Proc. the 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2020)*, 2020, pp. 46—-50.

[33] 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 *Proc. the Ninth International Conference on Learning Representations (ICLR 2021)*, 2021.

[34] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. the 3rd International Conference on Learning Representations (ICLR 2015)*, 2015, pp. 1–15.

SYMBOLIC MUSIC REPRESENTATIONS FOR CLASSIFICATION TASKS: A SYSTEMATIC EVALUATION

Huan Zhang[1] **Emmanouil Karystinaios**[2] **Simon Dixon**[1]
Gerhard Widmer[2] **Carlos Eduardo Cancino-Chacón**[2]
[1] Queen Mary University of London, United Kingdom
[2] Johannes Kepler University, Austra

ABSTRACT

Music Information Retrieval (MIR) has seen a recent surge in deep learning-based approaches, which often involve encoding symbolic music (i.e., music represented in terms of discrete note events) in an image-like or language-like fashion. However, symbolic music is neither an image nor a sentence, and research in the symbolic domain lacks a comprehensive overview of the different available representations. In this paper, we investigate matrix (piano roll), sequence, and graph representations and their corresponding neural architectures, in combination with symbolic scores and performances on three piece-level classification tasks. We also introduce a novel graph representation for symbolic performances and explore the capability of graph representations in global classification tasks. Our systematic evaluation shows advantages and limitations of each input representation. Our results suggest that the graph representation, as the newest and least explored among the three approaches, exhibits promising performance, while being more light-weight in training.

1. INTRODUCTION

The deep learning boom has profoundly impacted MIR, including research involving symbolic music representations (MIDI, scores, etc.). A large body of recent literature focuses on adapting existing architectures from computer vision and natural language processing to the field of symbolic MIR. These approaches often treat music data as an image (piano roll), as a sequence of language tokens, or, more recently, as a graph. However, a piece of music is neither an image nor a sentence or graph, therefore, a critical question still remains open concerning the choice of input representations for symbolic music.

A source of complexity in symbolic music arises from the different modalities of data such as scores and performances. A score contains information about music notation and often includes rich hierarchically structured information such as metrical structure and voicing. Symbolic music performances, on the other hand, such as those recorded on a MIDI-capable instrument, consist of a stream of controller events. Extracting a hierarchical structure from such a stream is not a trivial task [1–3]. Furthermore, such performance data omit some of the rich information that a score provides, such as pitch spelling and articulation markings, but instead, it can include information about expression, timing, local tempo, and performance dynamics.

Recent research has produced relatively large datasets containing scores and performances at the symbolic level, including efforts to align these [4–6]. Motivated by these developments, we present an attempt to shed light on questions revolving around the input representation of symbolic music for deep-learning-based MIR. We formulate an empirical framework where we test multiple input representations, models, and piece-level classification tasks.

In terms of input representations, we investigate piano rolls, tokenized sequences, and graphs. We evaluate multiple models based on these representations on three different tasks: composer classification, performer classification, and (playing) difficulty assessment. Furthermore, having datasets containing both performances and their corresponding scores such as ATEPP and ASAP [4, 5], allows us to apply each combination of representation and task to either score or performance. Our goal is to contribute an experimental overview of different symbolic music representations. The contributions of this work are threefold:

1. We investigate the performance and complexity of matrix, sequence and graph input representations, and their corresponding neural architectures (respectively Convolutional Neural Networks, Transformers, and Graph Neural Networks).
2. We compare the impact that the different information contained in symbolic scores and performances has on different piece-level classification tasks.
3. We introduce a new graph representation for symbolic performances, and explore the capability of graph representations in classification tasks.

2. RELATED WORK

The complexity of representing music data has been discussed in the literature [7–9]. Wiggins et al. [10] analyzed

© H. Zhang, E. Karystinatos, S. Dixon, G. Widmer, C.E. Cancino-Chacón. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** H. Zhang, E. Karystinatos, S. Dixon, G. Widmer, 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.,* Milan, Italy, 2023.

the trade-offs of music representation systems with respect to expressive completeness and structural generality. In the age of deep learning, such considerations are still relevant regarding the variety of machine-readable representations such as piano rolls, MIDI-like sequences, NoteTuples, and Musical Spaces [11, 12]. In this section, we focus on three symbolic representations (matrix, sequence, and graphs) and discuss their respective strengths and limitations.

Music as a Matrix: Similar to audio spectrograms, a pitch-time representation that is typically used as input to a CNN, the piano roll representation of music naturally emerges as the symbolic equivalent. Piano rolls have been widely applied in tasks such as automatic music transcription [13, 14], classification of piece-level attributes such as difficulty and composer [15–18], as well as generation of music accompaniment or performed dynamics [19, 20].

A piano roll is a bare-bones representation of symbolic music data, and, therefore, information such as key signatures, articulation annotations, metrical structure, different instrument parts, and voicing structure are not encoded in the representation [11, 21].

Music as a Sequence: Modeling symbolic music as sequences has a longstanding tradition in MIR. The multiple viewpoint system is a sequence representation that has been widely used for music analysis, generation, and classification [22–25], as well as the basis for cognitively plausible models of expectation [26, 27]. In this system, musical elements are represented by viewpoints [28], which are abstract functions mapping musical events to abstract derived features like pitch, interval, and melodic contour.

With the advances of deep learning-based language models, sequential representation of music as *language tokens* has recently received a lot of attention in sequence-to-sequence generative tasks from automatic orchestration [29] to description-based medley generation [30]. Similar to a stream of MIDI messages, various tokenization schemes encode music features such as pitch, onset time, duration, and velocity sequentially. Besides generation, large-scale pre-training using music sequences has been applied to downstream music understanding tasks [31, 32].

However, tokenized music sequence representations create difficulty for models to learn the dependency of long contexts. Length reduction methods such as Byte Pair Encoding (BPE) [29, 33] aim to address the length overflow problem by replacing the occurrence of frequent subsequences with new tokens.

Music as a Graph: A musical score can also be seen as a graph where notes form the vertices and relations between notes define the edges. Jeong and al. [34] introduced a graph modeling of a musical score for generating expressive performances. Recently, Karystinaios and Widmer [35] presented a new modeling of the score graph based on three different note relations and a Graph Convolutional Network for cadence detection in classical music. A score graph can be homogeneous or heterogeneous, i.e. having one or several types of edges and/or vertices, respectively [36]. We will investigate both heterogeneous and homogeneous score graphs based on the representation used in [35].

Graph Neural Networks have gained popularity in recent years, however, graph learning inherently presents some limitations, such as over-smoothing in deep graph networks [37] and restrictions of Message Passing, where information in graph neural networks flows only between edge relations predetermined by the representation (in contrast to a Transformer architecture where everything is interconnected [38]).

3. METHODOLOGY

In this section, we describe the methodology followed, the corpora used, and the experiments conducted to investigate in-depth the different symbolic representations.

3.1 Representation Design

We briefly introduce a formal definition of each representation type, i.e. matrix, sequence, and graph. An example of the three representations is shown in Figure 1.

3.1.1 Matrix

We define as a matrix representation of music a 2-dimensional array $\mathbf{M} \in \mathbb{N}^{H \times W}$ that depicts musical notes on the time axis, commonly referred to as a piano roll. The vertical axis consists of 128 possible values attributed to the MIDI pitch of note events, where we add three more optional fields for the *una corda*, *sostenuto*, and sustain pedals only applied on the MIDI performances.

In this work, we experimented with multiple channels as used in Onsets and Frames [39]. The onset channel is a binarized roll with activations at onset timestamps, while the frame channel encodes the duration of the note and the velocity of the MIDI event. For scores, the velocity values are substituted by the voice index, i.e. the integer number assigned to a note to indicate the index among the number of independent voices. [1]

3.1.2 Sequence

A symbolic music sequence $\mathbf{S} \in \mathbb{N}^{1 \times N}$ is defined by a series of discrete tokens that represent attributes of notes. Vocabularies such as $V_{\texttt{pitch}}, V_{\texttt{TimeShift}}, V_{\texttt{Vel}}$ assign semantic meanings to tokens, and different tokenization schemes translate into different grammars of sequence construction. In this work, we test three popular tokenization schemes: *MIDILike* [40, 41], *REMI* [42], and *CompoundWord* [43] and use the implementation of the MidiTok library [44].

As there is no existing tokenizer for processing scores, we implemented custom MusicXML tokenizers following MidiTok's framework, in the style of *REMI* as well as *CompoundWord*. The major difference is the timing of bars and event positions, as well as the addition of score-specific tokens such as $V_{\texttt{KeySig}}, V_{\texttt{Voice}}$. [2]

Byte Pair Encoding (BPE) is a tokenizer add-on technique that has recently been applied to music sequence learning [33]. It consists of a data compression technique

[1] This voice information is commonly available in formats such as MusicXML, **Kern, and MEI.

[2] Full documentation is provided with our open-source tokenizer in the project repository.

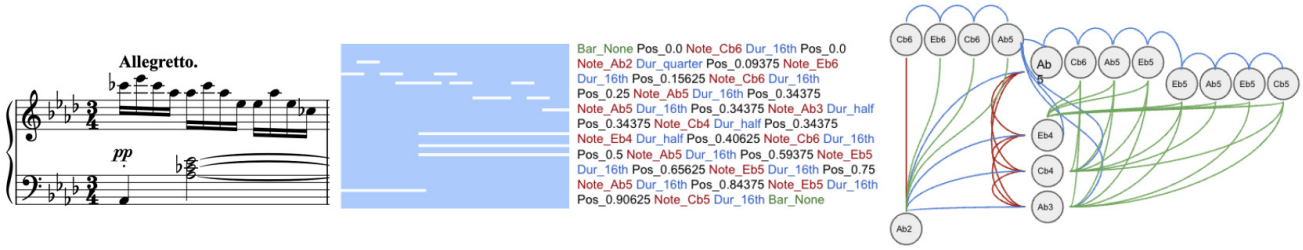


Figure 1. Excerpt of Schubert's *Impromptu Op. 90 No.4* and its input visualizations (from left to right): generic matrix, sequence (REMI-like) and graph.

that replaces the most common token subsequences in a corpus with newly created tokens. BPE increases the vocabulary size and shortens the sequence length. We follow the best results from [33] and adopt a BPE with 4 times the original vocabulary size. On average, this reduced our sequence length between $55 - 65\%$ in both datasets.

3.1.3 Graph

A homogeneous score graph G is defined by a tuple (V, E) of vertices and edges. V is the set of notes in a musical score and $E \subseteq V \times V$. Given a score with N notes, we extract a matrix of k-dimensional note-wise features $X \in \mathbb{R}^{N \times k}$ based on features contained in the score or performance. A heterogenous score graph $G = (V, E, \mathcal{R})$ also includes a set of relation types $\mathcal{R}$ such that for every edge $e \in E$, e is of type $r \in \mathcal{R}$ if a condition defined by r holds. In our work, we consider the following relations between two notes u, v which define the edges $e \in E$:

- u and v have the same onset, i.e. $on(v) = on(u)$, then $r =$ onset;
- The offset of u is the onset of v, i.e. $off(u) = on(v)$, then $r =$ consecutive;
- The onset of u lies between the onset of v and the offset of v, i.e. $on(v) < on(u) \wedge on(u) < off(v)$, then $r =$ overlap.

The above relations only hold in the case of score graphs. To adapt this to performance graphs, we use a window tolerance t_{tol}, such that if two notes $(u, v) \in E$ and:

- $|on(v) - on(u)| < t_{\text{tol}}$, then $r =$ onset;
- $|off(u) - on(v)| < t_{\text{tol}}$, then $r =$ consecutive;
- $on(v) < on(u) \wedge on(u) < off(v)$, then $r =$ overlap.

In our configurations, for all graphs created from performance MIDI, we set $t_{\text{tol}} = 30\,\text{ms}$, a perceptual threshold of expressive timing [45]. In addition to the above relations, we consider the possibility of adding an inversely directed edge for the overlap and the consecutive edge types, and we name the inclusion of such edges *inverse edges*. For a homogeneous graph G_{hom} and heterogeneous graph G_{het}, $e \in G_{\text{hom}} \implies e \in G_{\text{het}}$.

The node features X are divided into two categories, the basic and the advanced features. The basic features are implicitly contained in any score or performance note such as one-hot encoding of pitch class and octave of the note's pitch, and duration information. The advanced features

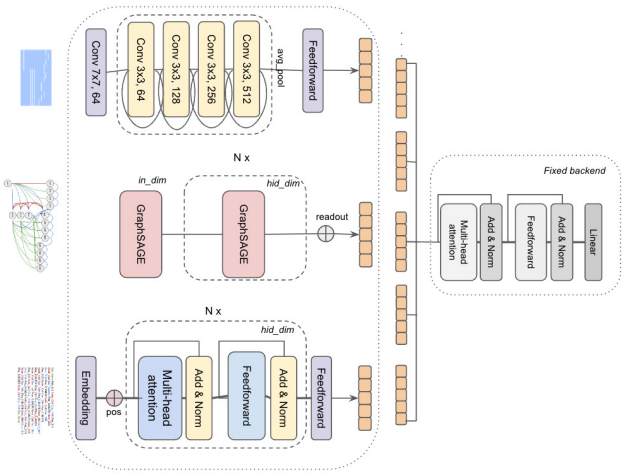


Figure 2. Left: front end for three representations, matrix, graph, and sequence, from top to bottom. Right: fixed back end with attention modules.

contains articulation, dynamics, and notation information from the *Partitura* python package [46]. The detailed computation of these features can be found in original partitura paper [47] and the basis mixer [48].

3.1.4 Information Levels

Given the differences in information captured by symbolic scores and performances (Sec. 1), we run experiments with separate levels of used information. For the base comparison experiments, we input the basic level of information that is present in both modalities: pitch, duration and onset. The advanced level of information for performance includes dynamics (MIDI velocity) and pedals, while for score includes the voice index (Sec. 3.1) as well as score markings such as articulation and dynamics. The results and comparison of each level of information, also with respect to different tasks, will be discussed in Section 4.3.

3.2 Modelling Pipelines

In this work, we evaluate the input representations under the same training pipeline of different piece-level classification tasks, as discussed in Section 3.3. We split our training architecture into two parts, a front end that projects a window of musical context into a 64-dimensional embedding, and a back end that aggregates the embedding for final prediction. The front end is representation-specific while the back end

		ASAP-performance		ASAP-score		ATEPP-performance		ATEPP-score	
		ACC	*F1*	*ACC*	*F1*	*ACC*	*F1*	*ACC*	*F1*
Matrix									
Resl	Chnl								
400	On+Fm	0.59±0.04	0.18±0.02	0.59±0.03	0.18±0.01	0.24±0.05	0.20±0.04	**0.25±0.02**	0.16±0.03
600	On+Fm	0.62±0.06	0.21±0.03	**0.61±0.07**	**0.19±0.02**	0.28±0.01	**0.22±0.03**	0.24±0.02	0.16±0.04
800	Fm	0.62±0.04	**0.21±0.02**	0.58±0.06	0.18±0.03	0.22±0.03	0.17±0.01	0.22±0.02	**0.18±0.03**
800	On+Fm	**0.63±0.04**	0.20±0.01	0.57±0.04	0.18±0.03	**0.28±0.02**	0.22±0.01	0.22±0.04	0.14±0.02
Sequence									
Tokn	BPE								
MidiLike	×	**0.53±0.05**	**0.16±0.02**	N/A	N/A	0.18±0.04	0.10±0.02	N/A	N/A
REMI	×	0.51±0.04	0.15±0.02	0.43±0.04	**0.14±0.01**	**0.23±0.04**	0.10±0.02	0.23±0.04	**0.13±0.02**
CP	×	0.48±0.02	0.09±0.05	**0.45±0.05**	0.10±0.01	0.11±0.02	0.09±0.01	0.17±0.06	0.11±0.04
MidiLike	4	0.52±0.04	0.15±0.02	N/A	N/A	0.17±0.03	0.12±0.01	N/A	N/A
REMI	4	0.51±0.02	0.15±0.01	0.43±0.03	0.13±0.01	0.21±0.01	**0.13±0.03**	**0.23±0.03**	0.13±0.01
Graph									
Bi-dir	Multi-rel								
×	×	0.56±0.01	0.17±0.02	0.51±0.05	0.16±0.02	0.22±0.02	0.10±0.03	0.23±0.03	0.21±0.05
×	✓	0.58±0.03	0.19±0.01	**0.54±0.05**	**0.17±0.02**	**0.27±0.03**	0.13±0.02	**0.29±0.10**	0.18±0.06
✓	✓	**0.62±0.02**	**0.21±0.01**	0.50±0.04	0.17±0.01	0.23±0.04	**0.16±0.03**	0.27±0.06	**0.22±0.03**

Table 1. Composer classification results for all representations, on all target subsets of our datasets on the composer classification task using only basic level features. For each subset of data, we present the accuracy score and the macro F1 score with 8-fold cross-validation. See Section 4.1 for explanation of the parameters.

rests fixed. For a fair comparison, we ensure that the same amount of musical context is given for different front ends to learn. For MIDI performances we fix a window of $60\,\mathrm{s}$, and for symbolic scores, we choose a window of 120 beats given that $120\,\mathrm{bpm}$ is a common tempo for music.

For the front end, we employ a commonly used architecture for each respective representation domain:

Matrix: Convolutional neural network based on ResNet [49] blocks with channel numbers adapted to our input.

Sequence: Transformer-encoder [50] front end with positional encoding. Each layer includes multi-head attention with 16 heads followed by an Add & Norm layer. For the combined tokens *CPWord* we add separate embedding layers for each token category in the front end.

Graph: Our graph convolution network (GCN) is built by stacking GraphSAGE blocks [51] followed by a global mean pooling layer. We experiment with both heterogeneous and homogeneous GraphSAGE. Note that a heterogeneous network has r times more parameters, where r is the number of distinct edge relation types.

For the fixed back end, we used a multi-head attention block with linear projection heads to the desired number of classes, as shown in Figure 2. To minimize the impact of model capacity on our comparative discussion, we carried out an ablation study to understand the size of the architecture proportional to each kind of representation (Sec. 4.2).

3.3 Tasks and Datasets

In this work, we focus on three tasks: composer classification, performer classification, and difficulty assessment. Each one of these tasks is a piece-level task since a label is attributed per piece. The composer classification consists of predicting the composer of the piece. The performer classification involves the prediction of the performer among a list of predefined performers included in the data source. Finally, difficulty assessment involves the prediction of a number between 1-9, with 1 being easy and 9 being hard. The difficulty labels were assembled from Henle Music. [3]

To evaluate the aforementioned tasks, we use two large-scale collections of Western classical piano music that contain corresponding symbolic scores (MusicXML files) and performances (MIDI files), ASAP (1067 performances, 245 scores) and ATEPP (11742 performances, 415 scores). Both datasets contain individual files per movement.

For the composer classification task, we exclude the least populated composer classes for balance in experiments, resulting in 10 classes for the ASAP dataset and 9 classes for the ATEPP dataset. The performer classification task uses MIDI performances of ATEPP with 20 classes. For difficulty, given that both ASAP and ATEPP datasets focus on concert repertoire, the actual classes used range from difficulty 4-9. [4] For all experiments, we use an eight-fold cross-validation evaluation where 85% of our data is used for training and 15% for testing in each fold.

3.4 Training

We performed hyperparameter optimization sweeps to determine the optimal learning rate and model hyperparameters. Our convergence criteria include early stopping at the 60 epoch breakpoint with the patience parameter set at 0.005 on the validation accuracy. All our experiments are trained on a single A5000 GPU, and the best models, training logs,

[3] Henle Music difficulty labels, `https://www.henle.de/en/about-us/levels-of-difficulty-piano/`

[4] The full distribution of the classes for each task is shown in the supplementary material.

and the code is available in the repository. [5]

4. EXPERIMENTS AND RESULTS

To evaluate the different representations we performed three experiments. Our first experiment focuses on a detailed comparison of the predictive accuracy of the three representations/architectures applied to the composer classification task, since it is the most well-understood task among the three. The second experiment studies the impact of model capacity (number of trainable parameters) per representation. Our last experiment investigates the effect of different levels of input features (see Section 3.1.4) on the three tasks.

4.1 Representations for Composer Classification

Our first experiment is a comparative analysis of the three representations on our two datasets, in the domains of both MIDI performance and MusicXML score with basic level features. For each representation group we test different configurations, i.e. for matrix we experiment with the channel (Chnl) and timestep resolution (Resl), for sequence we change the tokenization scheme (Tokn) and apply BPE, and for graph we investigate the effect of homogeneous or heterogeneous graphs (Multi-rel) and the addition of inverse edges (Bi-dir) (see Sec. 3.1). In Table 1, we present for each data subset the accuracy score and the macro F1 score and their respective standard deviations under 8-fold cross-validation (see Sec. 3.3).

In terms of observations per representation, the matrix representation results indicate no significant differences under different experimental configurations. For sequence representations, the *MIDILike* and *REMI* tokenization schemes yield comparable performance. However, our experiments suggest that *CPWord* is a more challenging representation to learn in the same setting. Concerning the BPE technique, no significant difference is observed between results with 4 times the original vocabulary and the non-BPE version.

Our graph-based models exhibit similar performance regardless of the configuration of the graph edges. In particular, the effect of reverse edges is not significant, and homogeneous graph convolution already achieves similar results to heterogeneous graph convolutional models, which indicates that implicit structural information contained in the heterogeneous approach is not strictly necessary for piece-level classification tasks.

Overall, we observe that three representations show small performance differences in given experiments, with the matrix-CNN approach having the overall best metric across the experiment groups and sequence have the worst.

Finally, we would like to discuss the *album effect*, which concerns the tendency of classification models to learn non-intended features, such as acoustic features in pieces of the same album [52]. In our case, this effect concerns different performances of the same piece that may give away cues for classification. Training with the entire corpus of performance MIDI, which involves different interpretations of the same piece, yields an average accuracy of 90% (see supplementary material), which is 30% higher for the `ASAP-perf` group. To address this issue, we fix the splits to only contain unseen pieces in the test set, which reduced the accuracy score gap between performance and score. This issue has often been overlooked in literature [53, 54] and a commonly-used dataset split is not piece-specific [16]. Given the recent development of large score-performance datasets, we wish to establish a scientifically correct evaluation split taking into consideration the *piece effect*.

[5] `https://github.com/anusfoil/SymRep`

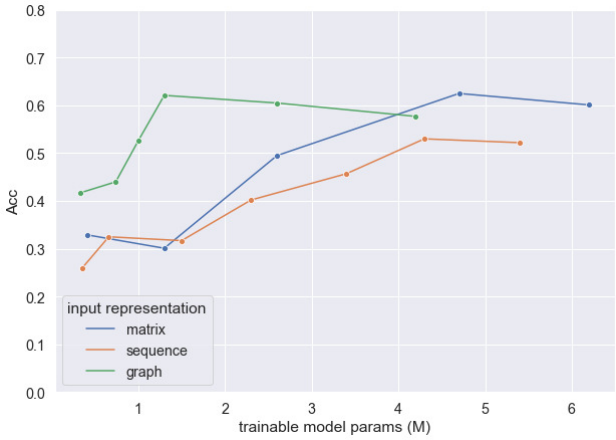


Figure 3. Model capacity vs. macro F1 score for each representation approaches on the `ASAP-composer` task.

4.2 Complexity

In our second experiment we investigate the impact of model capacity for each representation on the composer classification task using the ASAP dataset. We experiment with different hidden dimensions h and the number of layers N on each architecture corresponding to each of the three representations (Sec 3.2), and show our results in Figure 3. Overall, we observe that the GCN achieves its best performance using 1.3M parameters, while architectures for matrix and sequence achieve a similar accuracy at around three times the number of parameters.

Another observation concerns the use of large models for piece-level classification tasks on symbolic data. Large convolution models such as ResNet-18/34/50 [16] are substantially over-parametrized, as our results suggest we can achieve similar results using a reduced version of ResNet-8, using less than half the parameters of the smallest used ResNet architecture. Similar observations can be made for transformers, where scaling the model beyond 4.3M parameters does not further improve the performance. Our most efficient transformer encoder consists of 4 layers of attention modules with a hidden dimension of 256, significantly less than transformers used in previous related work [33].

Finally, we note one aspect of our results after scaling our graph network. While *oversmoothing* [37] (features of graph vertices converging to the same value) is a well-known challenge to train deep GCN, our best performing model is a relatively deep and narrow network consisting of 5 layers with a hidden dimension of 64. One possible interpretation is that convergence of node features does not

	Composer		Performer	Difficulty	
	perf	score	perf (ATEPP)	perf	score
Matrix					
basic feats	**0.625**	0.572	**0.364**	0.403	**0.420**
advanced feats	0.618	**0.577**	0.342	**0.411**	0.415
Sequence					
basic feats	**0.530**	**0.447**	0.287	**0.438**	**0.368**
advanced feats	0.513	0.393	**0.292**	0.426	0.349
Graph					
basic feats	**0.607**	0.545	0.305	**0.373**	0.361
advanced feats	0.598	**0.697**	**0.323**	0.356	**0.405**

Table 2. Accuracy of three identification tasks on the ASAP dataset, with basic or higher-level features.

complicate training in the graph-level classification context.

4.3 Comparison of Feature Levels and Tasks

As discussed in Section 3.1.4, we are also interested in understanding the impact of different levels of features on the three classification tasks. With this motivation, we performed our third set of experiments, where we adopted the best configuration of models explored in experiment 1 (see Section 4.1). We report the accuracy results in Table 2.

Our results indicate that MIDI performances and MusicXML scores have similar capabilities for distinguishing composers and difficulty. Furthermore, matrix and sequence approaches exhibit better results when learning with performances compared to scores. For the difficulty classification task, in particular, all three representations achieved approximately 40% accuracy on the 6 difficulty levels. Performer classification is more challenging since the difference lies in the timing nuances and dynamic changes instead of the pitch information, which are more prominent in our input representations. In the 20-way classification, our approaches generally achieved around 30% accuracy.

Our observations suggest that the addition of advanced features has a variable impact on the representations. Interestingly, the addition of advanced features does not improve the training from sequence representations in most experiments, which can possibly be explained by the increase in vocabulary size and relative sparsity of such information. Graph structures benefit from the addition of voice edges, especially in the representation of scores, where the performance boosts for both composer and difficulty classification. Notably, the `graph-score` with advanced features configuration achieved the best result in score-based composer classification, when jointly compared with Table 1.

4.4 Transformer vs. GNN: Are We Learning the Same Set of Musical Edges?

A transformer can be seen as a special case of Graph Neural Networks [38]. Assuming a fully connected graph where vertices are tokens in a sequence, we can draw parallels between a GCN and learned attention in a transformer block.

Therefore, we examine attention weights between `NoteOn` tokens in an effort to understand how our graph representation of the score relates to the sequence-based representation. For all pairs of `NoteOn` tokens from music sequences, we output their attention values and compute the correlation with the aggregated adjacency matrix (with all musical edges constructed in Sec. 3.1). Across the test set of ASAP composer classification on scores, there is a weak positive correlation, with Pearson's value of 0.212.

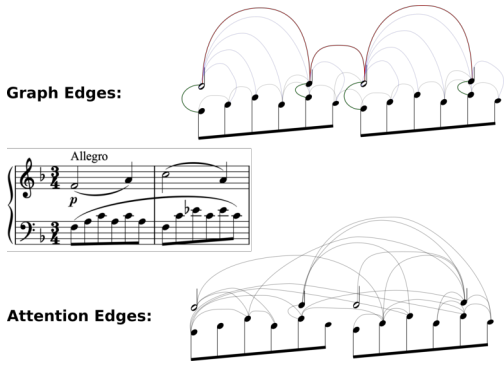


Figure 4. Visualization of graph edges (all edge types aggregated) and the attention among `NoteOn` tokens for the first measures of *Mozart Piano Sonata No.12, 1st mvt*.

In Figure 4, we visualize two measures of music with its constructed graph edges, and the attention across `NoteOn` tokens. We can observe some structural similarities, especially the overlap pattern in both measures, but overall the learned attention spans are much more global while graph edges connect nodes within a local range.

5. DISCUSSION AND FUTURE WORK

In this paper, we presented a series of systematic experiments to investigate the impact of symbolic representations for three piece-level tasks. In terms of simple *classification performance*, we found that for a given task, different representations showed small performance differences, but no clear pattern of superiority emerged. The matrix results were marginally better on average, and usually more robust to hyper-parameter changes. More advanced features were beneficial only for certain tasks and representations.

The *graph representation*, as the newest and least explored among the three approaches, exhibits promising performance, while being more light-weight (in terms of required model complexity – cf. Fig. 3). We observe that homogeneous graphs produce comparable results to heterogeneous graphs for our piece-level classification tasks, and deep GCNs perform better despite over-smoothing. As graphs are arguably a more natural representation for structured artifacts such as musical scores, we believe that they should merit more detailed studies in the future.

Our model complexity experiments demonstrated that commonly used architectures in the literature are larger than necessary for our tasks, as the same results can be achieved with smaller architectures (Section 4.2). Furthermore, we discussed the *album effect* in score-performance datasets, where multiple interpretations of the same composition may cause information leakage. Our results indicate the profound impact of the album effect, and we introduce new evaluation splits to guard against this effect.

6. ACKNOWLEDGEMENTS

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], also by the European Research Council (ERC) under the EU's Horizon 2020 research and innovation programme, grant agreement No. 101019375 (*Whither Music?*).

7. REFERENCES

[1] L. Liu, Q. Kong, V. Morfi, and E. Benetos, "Performance MIDI-to-score conversion by neural beat tracking," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2022.

[2] 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

[3] D. Temperley, *The Cognition of Basic Musical Structures*. MIT Press, 2004.

[4] 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)*, 2022.

[5] 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 International Society for Music Information Retrieval (in press)*, 2023.

[6] F. Foscarin, E. Karystinaios, S. D. Peter, C. Cancino-Chacón, M. Grachten, and G. Widmer, "The match file format: Encoding alignments between scores and performances," in *Proceedings of the Music Encoding Conference (MEC)*, 2022.

[7] I. Xenakis, *Formalized Music: Thoughts and Mathematics in Composition*, 1992.

[8] M. Harris, A. Smaill, and G. Wiggins, "Representing Music Symbolically," in *IX Colloquio di Informatica Musicale (Venice)*, 1991. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.43.473

[9] M. Babbit, "The use of computers in musicological research," *Perspectives of New Music*, vol. 3, no. 2, pp. 74–83, 1965.

[10] G. Wiggins, E. Miranda, A. Smaill, and M. Harris, "A Framework for the Evaluation of Music Representation Systems," *Computer Music Journal*, vol. 17, no. 3, pp. 31–42, 1993. [Online]. Available: https://about.jstor.org/terms

[11] C. Walder, "Modelling symbolic music: Beyond the piano roll," in *Journal of Machine Learning Research*, vol. 63, 2016, pp. 174–189.

[12] M. Prang, "Representation learning for symbolic music," Ph.D. dissertation, IRCAM, 2021. [Online]. Available: https://hal.archives-ouvertes.fr/tel-03329980

[13] E. Benetos, A. Klapuri, and S. Dixon, "Score-informed transcription for automatic piano tutoring," in *European Signal Processing Conference (EUSIPCO)*, 2012. [Online]. Available: http://c4dm.eecs.qmul.ac.uk/rdr/

[14] 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.

[15] 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. 10 183–10 196, 2022.

[16] S. Kim, H. Lee, S. Park, J. Lee, and K. Choi, "Deep Composer Classification Using Symbolic Representation," in *International Society for Music Information Retrieval (ISMIR) Late Breaking Demo (LBD)*, 2020. [Online]. Available: http://arxiv.org/abs/2010.00823

[17] 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. [Online]. Available: https://www.semanticscholar.org/paper/Composer-Recognition-Based-on-2D-Filtered-Velarde-Weyde/2ee8df37e3f5363c573b2aeed2243034ea638f71

[18] F. Foscarin, K. Hoedt, V. Praher, A. Flexer, and G. Widmer, "Concept-Based Techniques for "Musicologist-friendly" Explanations in a Deep Music Classifier," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2022. [Online]. Available: http://arxiv.org/abs/2208.12485

[19] 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 32nd AAAI Conference on Artificial Intelligence*, 2018. [Online]. Available: https://salu133445.github.io/musegan/

[20] S. van Herwaarden, M. Grachten, W. de Haas, and W. Bas de Haas, "Predicting expressive dynamics in piano performances using neural networks," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2014.

[21] J.-P. Briot, G. Hadjeres, and F.-D. Pachet, *Deep Learning Techniques for Music Generation – A Survey*, 2017. [Online]. Available: http://arxiv.org/abs/1709.01620

[22] 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. [Online]. Available: https://doi.org/10.1080/09298219508570672

[23] D. Conklin, "Multiple viewpoint systems for music classification," *Journal of New Music Research*, vol. 42, no. 1, pp. 19–26, 2013. [Online]. Available: https://doi.org/10.1080/09298215.2013.776611

[24] R. P. Whorley and D. Conklin, "Music generation from statistical models of harmony," *Journal of New Music Research*, vol. 45, no. 2, pp. 160–183, 2016. [Online]. Available: https://doi.org/10.1080/09298215.2016.1173708

[25] D. Conklin, "Chord sequence generation with semiotic patterns," *Journal of Mathematics and Music*, vol. 10, no. 2, pp. 92–106, 2016. [Online]. Available: https://doi.org/10.1080/17459737.2016.1188172

[26] M. T. Pearce, "Statistical learning and probabilistic prediction in music cognition: Mechanisms of stylistic enculturation," *Annals of the New York Academy of Sciences*, vol. 1423, no. 1, pp. 378–395, 2018. [Online]. Available: https://nyaspubs.onlinelibrary.wiley.com/doi/abs/10.1111/nyas.13654

[27] M. Pearce, "The Construction and Evaluation of Statistical Models of Melodic Structure in Music Perception and Composition," Ph.D. dissertation, City University of London, UK, 2005.

[28] 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.

[29] 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. [Online]. Available: http://arxiv.org/abs/2205.05448

[30] 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. [Online]. Available: http://arxiv.org/abs/2201.10936

[31] M. Keller, G. Loiseau, and L. Bigo, "What Musical Knowledge Does Self-Attention Learn?" in *Proceedings of the 2nd Workshop on NLP for Music and Spoken Audio (NLP4MusA)*, 2021, pp. 6–10. [Online]. Available: https://aclanthology.org/2021.nlp4musa-1.2

[32] 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.

[33] N. Fradet, J.-P. Briot, F. Chhel, A. E. F. Seghrouchni, and N. Gutowski, "Byte Pair Encoding for Symbolic Music," 2023. [Online]. Available: http://arxiv.org/abs/2301.11975

[34] 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.

[35] 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.

[36] C. Shi, Y. Li, J. Zhang, Y. Sun, and S. Y. Philip, "A survey of heterogeneous information network analysis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 1, pp. 17–37, 2016.

[37] G. Li, M. Muller, A. Thabet, and B. Ghanem, "DeepGCNs: Can GCNs go as deep as CNNs?" in *Proceedings of the IEEE International Conference on Computer Vision*, 2019. [Online]. Available: https://sites.google.com/view/deep-gcns

[38] P. Veličković, "Everything is Connected: Graph Neural Networks," *Artificial Intelligence (AI) Methodology in Structural Biology*, 2023. [Online]. Available: http://arxiv.org/abs/2301.08210

[39] 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 (ISMIR)*, 2018, pp. 50–57.

[40] 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, 2018.

[41] 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," in *Proceedings of the International Conference of Learning Representations (ICLR)*, 2019. [Online]. Available: http://arxiv.org/abs/1809.04281

[42] 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.

[43] 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 35th AAAI Conference on Artificial Intelligence*, 2021.

[44] N. Fradet, J.-P. Briot, F. Chhel, A. El Fallah Seghrouchni, and N. Gutowski, "Miditok: a Python Package for Midi File Tokenization," in *International*

Society for Music Information Retrieval (ISMIR) Late Breaking Demo (LBD), 2021.

[45] W. Goebl, "Melody lead in piano performance: Expressive device or artifact?" *The Journal of the Acoustical Society of America*, vol. 110, p. 641, 2001. [Online]. Available: https://asa.scitation.org/doi/10.1121/1.1376133

[46] C. Cancino-Chacón, S. D. Peter, E. Karystinaios, F. Foscarin, M. Grachten, and G. Widmer, "Partitura: A Python package for symbolic music processing," in *Proceedings of the Music Encoding Conference (MEC)*, 2022.

[47] C. Cancino-Chacón, S. D. Peter, E. Karystinaios, F. Foscarin, M. Grachten, and G. Widmer, "Partitura: A Python Package for Symbolic Music Processing," pp. 1–9, 2022. [Online]. Available: http://arxiv.org/abs/2206.01071

[48] C. E. Cancino-Chacón, M. Grachten, W. Goebl, and G. Widmer, "Computational Models of Expressive Music Performance: A Comprehensive and Critical Review," *Frontiers in Digital Humanities*, vol. 5, no. October, pp. 1–23, 2018.

[49] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[50] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017.

[51] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive Representation Learning on Large Graphs," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017.

[52] A. Flexer, "A closer look on artist filters for musical genre classification," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2007.

[53] G. Micchi, "A neural network for composer classification," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR) Late-Breading Demo (LBD)*, 2018.

[54] Q. Kong, K. Choi, and Y. Wang, "Large-Scale MIDI-Based Composer Classification," in *arXiv*, 2020.

8. APPENDIX

8.1 Album effect

As mentioned in the main paper Section 4, the *Album Effect* remains a non-trivial issue in similar classification tasks. Here we present in Table 4 the same content as the original table (Table 1) from the paper which contains results from the experiment that is trained on the entire performance corpus with overlapping interpretations. Training under this non-piece-specific split, we achieved comparable accuracy (93%) with the literature [16].

8.2 Complexity

8.2.1 Memory

Given that the same amount of music context is input into the models, we are interested in understanding the memory efficiency of the representations. We used the native `numpy` and `cuda` functions to monitor the memory of data and memory changes during training.

In terms the representation of a single piece of data, sequence is the most compact one while matrix takes $70\times$ more space, given that a lot of redundant pixels are taken in the 2D representation. The size of graph varies depending on the number of nodes and edges, but overall it is in between that of the matrix and sequence.

However, during training we can observe that the sequence is the least memory-efficient representation during training, and it takes $30\times$ compares to the memory usage of matrix and graphs. Given the quadratic complexity of transformer-like architectures, the training memory needed is one of the major limitation of sequence compared to the other representations.

	KB / seg	KB / piece	Training step (MB)
Mtr	819.2	5129.6 ± 3332.7	185.9 ± 105.9
Seq	12.8	77.8 ± 56.7	5548.9 ± 1736.2
Gph	100.5 ± 57.3	610.9 ± 300.0	125.2 ± 103.4

Table 3. Size estimation of each representation with basic level features from ASAP-perf data. We include the average size per segment (60s), average size per piece (as piece have different length), as well as the average allocated memory increase during each training step with a batch size of 1.

8.2.2 Convergence epochs

During training, we also observed a difference in the time it takes the models to convergence, given the 60 epochs convergence criteria defined in Sec 3.4. We first performed learning rate search using pytorch lightning's learning rate finder. Under the suggested learning rate, among different `ASAP-perf` experiment of composer classification, the matrix have on average 143.0±24.7 epochs to converge, the sequence and the graph have 132.0±31.1 and 262.0±55.7 epochs. During training, the graph models have relatively slower learning progress.

8.3 Dataset class distributions

We present our dataset class distribution for each task in the Table 8.4.

8.4 Silence and voice edges

Besides the onset, consecutive and overlap edges in Sec 3.1, we also add optional silence edges (edges that bridge over silence) to ensure a connected graph. A silence edge $E_{silence}$ is added between a node that's not connected by any consecutive edge and the time-wise closest node before it. The silence edge doesn't carry much music semantic meaning, and its main purpose is to prevent the disjoint subgraphs formed by distinct music sections, in which stops information flow in training.

In the advanced representation of score graph, we input the voicing information as voice edges. Given that we can't guarantee the consistency of voice annotation in MusicXML scores (as they are mostly labeled for visual purposes like beaming), we limit the voice edge connection within a measure: If two notes are labelled with the same voice, then they are connected by a voice edge E_{voice}.

		ASAP-performance		ASAP-score		ATEPP-performance		ATEPP-score	
		ACC	*F1*	*ACC*	*F1*	*ACC*	*F1*	*ACC*	*F1*
Matrix									
Resl	Chnl								
400	On+Fm	0.926±0.02	0.796±0.06	0.598±0.03	0.177±0.01	0.905±0.04	0.796±0.03	**0.246±0.02**	0.156±0.03
600	On+Fm	**0.931±0.01**	0.800±0.07	**0.613±0.07**	**0.186±0.02**	**0.930±0.05**	0.818±0.03	0.238±0.02	0.156±0.04
800	Fm	0.925±0.02	0.723±0.11	0.583±0.06	0.182±0.03	0.891±0.02	0.737±0.02	0.221±0.02	**0.181±0.03**
800	On+Fm	0.926±0.02	**0.812±0.05**	0.572±0.04	0.185±0.03	0.932±0.03	**0.832±0.01**	0.225±0.04	0.138±0.02
Sequence									
Tokn	BPE								
MidiLike	×	0.860±0.03	0.674±0.11	N/A	N/A	**0.926±0.01**	**0.769±0.01**	N/A	N/A
REMI	×	0.783±0.04	0.521±0.05	0.431±0.04	**0.138±0.01**	0.910±0.01	0.729±0.02	0.229±0.04	**0.129±0.02**
CP	×	0.679±0.08	0.331±0.06	**0.447±0.05**	0.099±0.01	0.864±0.02	0.556±0.01	0.171±0.06	0.107±0.04
MidiLike	4	**0.905±0.02**	**0.727±0.06**	N/A	N/A	0.895±0.01	0.691±0.01	N/A	N/A
REMI	4	0.862±0.01	0.692±0.07	0.432±0.03	0.132±0.01	0.826±0.04	0.529±0.03	**0.234±0.03**	0.125±0.01
Graph									
Bi-dir	Multi-rel								
×	×	0.768±0.03	0.500±0.08	0.509±0.05	0.163±0.02	0.788±0.03	0.501±0.06	0.226±0.03	0.205±0.05
×	✓	**0.861±0.03**	**0.763±0.03**	**0.545±0.05**	**0.174±0.02**	**0.928±0.01**	**0.781±0.03**	**0.289±0.10**	0.176±0.06
✓	✓	0.833±0.03	0.703±0.11	0.500±0.04	0.173±0.01	0.897±0.01	0.767±0.02	0.271±0.06	**0.217±0.03**

Table 4. Base experiment composer classification results with the entire performance MIDI corpus and no piece-specific split.

ASAP composer		**ATEPP composer**		**ATEPP performer**		**ASAP difficulty**	
Beethoven	195	Beethoven	3033	Richter	1581	9	164
Bach	163	Chopin	1739	Ashkenazy	1188	8	176
Chopin	162	Mozart	653	Arrau	833	7	132
Liszt	67	Schubert	264	Brendel	743	6	150
Schubert	55	Debussy	254	Kempff	609	5	56
Schumann	26	Schumann	243	Barenboim	603	4	23
Haydn	23	Bach	231	Schiff	595		
Mozart	10	Ravel	169	Horowitz	576		
Scriabin	9	Liszt	122	Gulda	459		
Ravel	9			Gieseking	362		
				Gould	326		
				Gilels	322		
				Perahia	288		
				Pollini	256		
				Argerich	240		
				Schnabel	240		
				François	234		
				Uchida	210		
				Casadesus	164		
				Lugansky	125		

Table 5. Dataset class distribution for the tasks. The performer task is in regards to the distribution of the performed MIDI, and the other three columns are in regards to the MusicXML score.

THE MUSIC META ONTOLOGY: A FLEXIBLE SEMANTIC MODEL FOR THE INTEROPERABILITY OF MUSIC METADATA

Jacopo de Berardinis[1] **Valentina Anita Carriero** [2] **Albert Meroño-Penuela**[1]
Andrea Poltronieri[2] **Valentina Presutti**[2]
[1] King's College London, UK
[2] University of Bologna, Italy
jacopo.deberardinis@kcl.ac.uk, andrea.poltronieri2@unibo.it

ABSTRACT

The semantic description of music metadata is a key requirement for the creation of music datasets that can be aligned, integrated, and accessed for information retrieval and knowledge discovery. It is nonetheless an open challenge due to the complexity of musical concepts arising from different genres, styles, and periods – standing to benefit from a lingua franca to accommodate various stakeholders (musicologists, librarians, data engineers, etc.). To initiate this transition, we introduce the Music Meta ontology, a rich and flexible semantic model to describe music metadata related to artists, compositions, performances, recordings, and links. We follow eXtreme Design methodologies and best practices for data engineering, to reflect the perspectives and the requirements of various stakeholders into the design of the model, while leveraging ontology design patterns and accounting for provenance at different levels (claims, links). After presenting the main features of Music Meta, we provide a first evaluation of the model, alignments to other schema (Music Ontology, DOREMUS, Wikidata), and support for data transformation.

1. INTRODUCTION

A music analyst, a computational musicologist, a music librarian, and a data engineer are working on a joint project. They need to contribute data from various musical sources, ranging from music libraries, annotated corpora and tune books, to audiovisual archives, radio broadcasts, and music catalogues. All data is eventually merged/aggregated as interconnected corpora, and linked to online music databases (e.g. MusicBrainz, Discogs) and knowledge bases (e.g. Wikidata). This creates opportunities to link cultural heritage artefacts to music industry data (streaming services, music professionals, etc.) and viceversa.

This plot subsumes a recurring challenge for musical heritage projects [1]. Besides the individual requirements of each stakeholder – possibly rooted in different music genres, periods and datasets, a fundamental requirement is the interoperability of music metadata.

Music metadata (alias bibliographic, or documentary music data) is used to consistently identify and describe musical works, their artists, recordings, and performances. For music industry, it allows for efficient management and distribution of music, which facilitate search and recommendation [2]. When metadata is accurate, it ensures that artists receive proper credit and compensation [3]. For musical heritage, metadata allows for the preservation and dissemination of musical works and traditions, but also aid in the research and study of music history and culture [4]. When integrating both views, metadata can help to promote diversity and inclusivity in the music industry by highlighting lesser-known genres and artists, while integrating information and artefacts of cultural interest [5].

Hence, a model that can consistently describe metadata is highly desirable – as it enables linking entities and concepts from various datasets (e.g. a composer is linked to a tune that has no authors in another collection). Semantic Web technologies can help achieve interoperability, as they facilitate data access and integration, resource discovery, semantic reasoning and knowledge extraction [6]. In the Resource Description Framework [7], data is described as `<subject-predicate-object>` triples using ontologies, and released as Knowledge Graphs (KGs).

To achieve interoperability, one possibility akin to [8] is to let stakeholders design their own domain-specific ontologies, then use alignment algorithms to find connections between them (e.g. `MusicalWork` and `Composition` referring to the same concept). However, this approach comes with three major drawbacks: (i) ontology alignment is error-prone, hence links would still require manual inspection; (ii) even when alignment is sound, the semantics of classes and relationships may vastly differ across domains, which in turn, may create inconsistent alignments; (iii) it does not address the problem in the long-term.

1.1 Challenges and requirements for interoperability

Another possibility is to reuse current ontologies for music metadata, such as the Music Ontology (MO) [9] and the DOREMUS ontology [10]. However, modelling music metadata across different genres and historical periods, to

© J. de Berardinis, V.A. Carriero, A. Meroño-Peñuela, A. Poltronieri, and V. Presutti. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** J. de Berardinis, V.A. Carriero, A. Meroño-Peñuela, A. Poltronieri, and V. Presutti, "The Music Meta Ontology: a flexible semantic model for the interoperability of music metadata", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

accommodate various use cases over heterogeneous data sources poses a number of challenges. First of all, it requires a perspective that harmonises all requirements from different stakeholders – to design a model that can be tailored to different data sources rather than to a single type of dataset. We categorise the main challenges and requirements for metadata interoperability as follows.

1.1.1 Domain specificity hampers interoperability

When looking at current ontologies, MO leans towards modelling discographic data with a focus on contemporary music, whereas DOREMUS is inherently rooted in classical music. These ontologies have been demonstrated to model metadata from MusicBrainz and BBC Music [11], and from classical music libraries and radio broadcasts for concerts programming [12], respectively. Their specificity makes them appealing when downstream applications show considerable overlap in terms of requirements and data. Examples include the reuse of MO in the WASABI project [13], to support the semantic annotation of audio music (emotions, lyrics, structures), but also for music recommendation [14] and listening [15]; and the adoption of DOREMUS by *Philarmonie de Paris*, *Biblioteque National de France*, and *Radio France*.

Nevertheless, when drifting from discographic data and classical music, or attempting to reuse both models, addressing e.g. cultural heritage requirements while fostering interoperability becomes difficult. Indeed, a model reflecting the view and the interpretations ascribable to a musical genre, stakeholder, or dataset type may be difficult to reuse and extend to other domains. For instance, a music artefact may originate from oral transmission or be the result of a creative process that does not necessarily entail a formal composition process. The latter is common in songwriting, but also in folk music whenever a set of tunes (collected from different manuscripts) allows for the identification of a tune family [16]. Similarly, when expressing relationships between musical artefacts (alias derivations), it is important not to impose any modelling bias that may constrain possible interpretations (e.g. an arrangement having proper musical identity vs simply providing a different instrumentation). This is commonly referred to as "dominance of concept" [12], whose definition should be left to users depending on their data and domain expertise.

Rather than attempting to achieve consensus on musical concepts and jargon, accounting for the interoperability calls for an abstraction layer for music metadata ("*zoom-out*") that can then be specialised, extended, and adapted to address domain-specific requirements ("*zoom-in*").

1.1.2 Expressivity is needed at different levels

Another requirement for interoperability and reuse across various data sources is providing expressivity at different degrees, i.e. the possibility to conveniently describe music metadata at the right level of detail. For example, one data source may have granular/detailed information that requires high semantic expressivity (a composition process spread over different time, places, and involving more artists); whereas others may have basic (only the name of an artist is known) or even incomplete and uncertain information (a composition tentatively attributed to an artist).

Here, the WikiProject Music [1] has been successful in providing expressivity to represent music metadata from different sources. As an extreme case of ontological flexibility, the schema underlying Wikidata – an open-ended, multi-domain KG built collaboratively like Wikipedia – is not specified in a previously agreed ontology, and the high expressivity overly adds complexity to the model. This is due to Wikidata's scope being the most general.

1.1.3 Provenance is fundamental for data integration

Accounting for provenance is a central requirement for both cultural heritage and music industry. This becomes fundamental when integrating Knowledge Graphs from different datasets and stakeholders – as every single bit of data (each triple) should be attributable to a dataset/KG. Furthermore, integrating provenance is also needed within the context of a single dataset, at least for claims and links.

Claims-Interpretations. Cultural heritage applications often require representing debatable statements or claims [17, 18]. These are usually the result of an interpretation process based on factual or documentary evidence (a dataset, a manuscript, etc.), and following a methodology and/or theory. Examples include personal information (e.g. the year/place of birth of a composer), and authorship claims (e.g. a composition being attributed to an artist).

Links and identifiers. These includes links to artists' official websites, fan pages, discussion forums, music reviews, record shops; as well as identifiers from music databases (e.g. MusicBrainz, Discogs, AllMusic), streaming platforms (e.g. Deezer, Spotify), and authoritative sources (e.g. ISNI, ISWC, ISRC). As most links and identifiers are crowdsourced or automatically inferred by entity linking algorithms, modelling provenance here promotes traceability and accountability of data sources.

Notably, Wikidata addresses both these requirements, as every triple is considered a statement per se, for which so-called *references* can be appended and ranked. References may include information on the source, whether a computational method was used, and a date of retrieval.

1.2 Our contribution

We leverage the expertise and complementary views of various music stakeholders (musicologists, data engineers, music analysts, and heritage archivists) to contribute:

- The *Music Meta* ontology, a rich flexible model to describe Western music metadata and its provenance at different levels of granularity.
- An example-driven validation of the model, focused on the data elicited from four different stakeholders.
- Code support to create Music Meta KGs without expert knowledge of the model, with automatic alignments to the MO, DOREMUS, and Wikidata.

[1] `https://en.wikipedia.org/wiki/Wikipedia:WikiProject_Music`

2. RELATED WORK

Besides metadata, the use of Semantic Web technologies in the music domain has contributed several ontologies, covering a variety of musical aspects and spanning both symbolic and audio music.

Among them, the Music Theory Ontology [19] describe theoretical concepts of compositions, whereas the Music Score [20] and Music Notation [21] propose granular ontologies to represent elements of music scores. OMAC expresses features of musical entities but also musicological claims [22], while the OMRAS project [23] contributed ontologies to describe music chords as well as concepts related to tonality and temperament.

In the audio domain, ontologies describe music production [24], audio features [25], effects [26]; an also model listening habits/taste [27], music-induced emotions [28], music structure [29, 30], and musical similarities [31].

These ontologies have specific focus, and many were developed as stand-alone projects, with little or no alignment [32]. Instead, some ontologies focus on achieving interoperability between notations, taxonomies, and formats. These include the Internet of Musical Things [33], where heterogeneous musical objects are envisioned to coexist; the Music Annotation Pattern [34] which allows to model music annotations in the JAMS format [35]; and the Hamse ontology [36] describing musical features for musicological research. Similarly, [37] models abstract annotations of musical works, rather than concrete encodings.

Interoperability at the level of musical content level resulted in successful MIR applications, such as the MIDI Linked Data Cloud [38] – integrating MIDI music to learn embeddings over the resulting KG [39]; and ChoCo [40] – a chord corpus integrating 18 chord datasets and enabling novel workflows for computational creativity [41].

3. THE MUSIC META ONTOLOGY

To derive requirements from various music stakeholders, we leverage the domain expertise and views in Polifonia – a European H2020 project aiming to connect "music, people, places and events" from the 16th century. The interdisciplinarity of Polifonia, involving data engineers, anthropologists, ethnomusicologists, historians of music, linguists, musical heritage archivists, cataloguers, and creative professionals – makes it an ideal testbed for this work.

Music Meta is part of the Polifonia Ontology Network [42], from which we reuse the CORE module. This is done to consistently reuse general-purpose elements of design (e.g. Person, Time, Place) and ontology design patterns. The reuse of this module also ensures alignment with other foundational models (FOAF, Dublin Core, etc.).

The ontology (prefixed as `mm`) is available at the following URI: `https://w3id.org/polifonia/ontology/music-meta/`, and is released as open source project under the `CC-BY 4.0` on GitHub [2].

[2] `https://github.com/polifonia-project/music-meta-ontology`

3.1 Methodology

The development of Music Meta is driven by eXtreme Design (XD) [43], an agile ontology engineering methodology that makes extensive use of ontology design patterns (ODPs) – small ontologies that work as reusable templates for recurrent modelling problems. An ODP is intuitive and compact, clearly and formally defined, tackles a specific (sub)set of requirements, and is designed for a modular reuse, enabling a pragmatic cognitive analysis [44].

In XD, a story-based approach guides the collection of requirements. A story is a framework for *customers* to describe their needs, and is composed of 4 sections: (i) the persona, a description of a typical user; (ii) the overarching goal they need to address; (iii) the scenario, describing how the goal will be address; (iv) the competency questions (CQs) translating needs into formal requirements. Ontology modelling starts iteratively from the CQs, and is based on the reuse of ODPs and existent templates.

3.1.1 From FRBR to Information Objects/Realisations

At the core of Music Meta lies the use of the Information-Realisation (IR) ODP [45]. An *information object* is a non-physical social object carrying information that can have one or multiple materialisations (*information realisations*). Each realisation is a particular physical object, or event, realising the the *information object*, or involving the latter as a participant. Both information object and realisation are intended as information entities (IE), i.e. (social) objects created and/or used to communicate, reason, and specify new entities. This allows to distinguish between a piece of information (e.g. the *content* of a composition) from how it is materialised (e.g. as a performance).

On the other hand, both the Music Ontology [9] and DOREMUS [12] are built on top of different flavours of FRBR [46] (FRBRer and FRBRoo, respectively). FRBR is a conceptual model describing bibliographic resources at four levels: *Work*, *Expression*, *Manifestation*, and*Item*. In contrast, the two levels of the IR pattern map to *Expression* and *Item*, since *Work* and *Manifestation* are said to provide non-informative conceptualisations [45]. Moreover, [47] argues that FRBR's Works – intended as "entities that pre-exist expressions", cannot represent improvisations or traditional music, as they do not derive from a formal composition process leading to a realisation. FRBR's Work is often ambiguously intended as an entity retrospectively created for grouping multiple expressions for cataloguing needs. As for the Manifestation level, while its representation is straightforward in the bibliographic domain (e.g. the printed version of a book), its correspondence in the music domain is not fully intuitive, as it may relate to either a recording, a score, a compact disc, or all the above – thereby introducing complexity and ambiguity.

Nevertheless, being aligned to two levels of FRBR, the IR ODP makes our model leaner and flexible, while still achieving interoperability with FRBR-based (music) ontologies. In fact, IE patterns are meant to boost the semantic integration of contents, tools, platforms, resources that are silo-ed or non-interoperable [45].

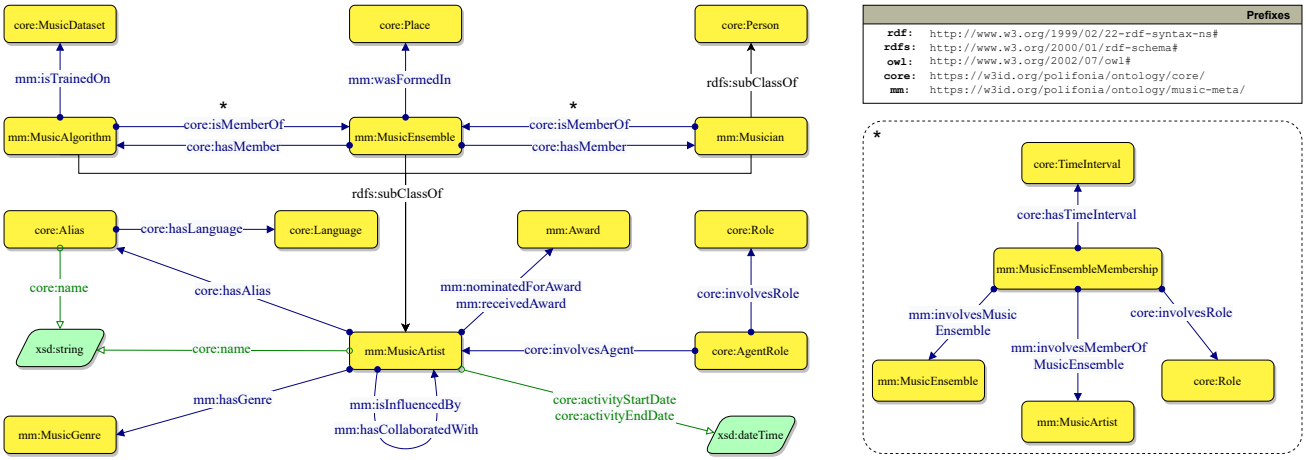


Figure 1. Describing music artists as musicians, music ensembles, and algorithms using the Graffoo notation (*yellow* boxes are classes, *blue/green* arrows are object/datatype properties, *purple* circles are individuals, *green* polygons are datatypes).

3.2 Main elements of design

From Polifonia's CQs [3], we identified those related to metadata, and aimed for a model capable to address the requirements in Section 1.1. Music Meta follows a hierarchical design (where each level extends the former to add expressiveness) and is complemented by data transformation rules to conveniently translate one level into another.

To enable data integration from existing knowledge bases and datasets, we align Music Meta to other ontologies: the Music Ontology, DOREMUS, and Wikidata, after having identified common/similar classes and properties.

3.2.1 Music artists

To represent music creatives the class `mm:MusicArtist` generalises over musicians (`mm:Musician`), ensembles (`mm:MusicEnsemble`), and computational methods (`mm:MusicAlgorithm`), as illustrated in Figure 1. Musicians are seen as a specialisation of persons who can optionally be associated to a medium of performance (e.g. voice, guitar), and be part of a music ensemble (e.g. `MusicGroup`, `Orchestra`, `Choir`). Depending on the data available, the latter can be expressed either through a membership relationship (`core:isMemberOf`), a specialisation of the former, such as `mm:isSingerOf`, or through a `mm:MusicEnsembleMembership` when the period of participation of the musician is available.

All music artists can be associated to (one or more) `mm:MusicGenre`(s), express influences or collaborations, and share a period of activity. Here, the start date refers to the foundation for music ensembles, whereas the end date is used for discontinued projects for algorithms.

3.2.2 Music inception

The focal point of Music Meta is the `mm:MusicEntity` class (Figures 2 and 3). This class represents an Information Object, which is defined as the sum of all the elements that make up a piece of music. A Music Entity is composed of several components, including lyrics (generalised through `mm:Text` to also account for `mm:Libretto`), the entailed musical content (`mm:AbstractScore`) and its instrumentation (`mm:Instrumentation`).

A `mm:AbstractScore` provides an abstraction to describe the musical properties of an entity, such as the form of a piece (`mm:FormType`), its constituents parts (e.g. `mm:Movement` or `mm:Section`), and its key (`mm:Key`). Datatype properties also describe the tempo of the composition (`mm:tempo`) and its order (`mm:orderNumber`). A `mm:Instrumentation` can instead be formalised in a `mm:Score`, which can be either digital or paper. Through the score, the instrumentation describes one or more `mm:MediumOfPerformance`, each of which has a cardinality (e.g. 3 violins).

It is also possible to describe relationships between different Music Entities, defined by parthood (`mm:hasPart`) and derivation (`mm:isDerivedFrom`). Derivations are used at the user's discretion, based on the dominance of concept [12] (whose criteria attribute proper identity to a musical entity) and can be of different types: revision, transposition, cover, reconstruction, reduction, etc. This makes it possible to describe different types of compositions, rearrangements and modifications of an original piece, as well as influences and more complex types of derivations. For example, the production of a cover song (e.g. in a different musical genre) may keep the lyrics and introduce a new composition and instrumentation, hence resulting in a new `mm:MusicEntity`. In addition, Music Entities can be organised in `mm:Collection`, according to a `mm:CollectionConcept` that binds them together.

In sum, the model provides flexibility across periods and genres as the proposed classes allow generalisations to be made about the text, the musical composition and its arrangement. (c.f. Section 1.1.1). Through the specialisation of classes, depending on the target domain/application, specificity can easily be achieved (c.f. Section 1.1.2). For example, a tune family can be seen as a `mm:Collection` encompassing several tunes (as music entities) based on specific criteria (e.g. similarity, provenance).

[3] `https://github.com/polifonia-project/stories`

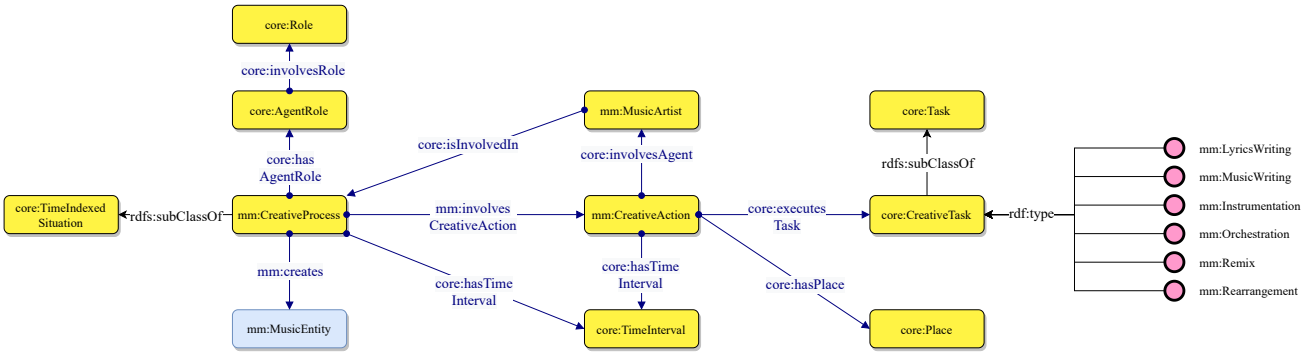


Figure 2. Abstracting music inception as an product of a creative process, involving music artists in activities (music writing, instrumentation, etc.), defined in time and space and according to different roles.

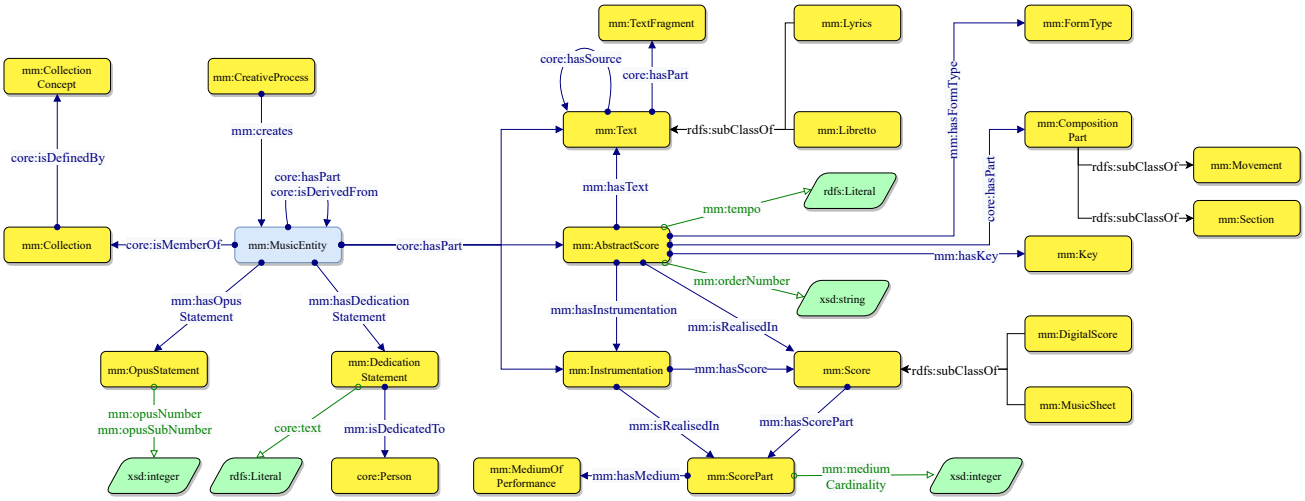


Figure 3. Describing a music entity and the elements it contains: Text, AbstractScore and Instrumentation.

3.2.3 From performance to recording and broadcast

The realisation of a `mm:MusicEntity` is exemplified by `mm:MusicalPerformance`, which can be either live (`mm:LivePerformance`) or in a studio (`mm:StudioPerformance`). As illustrated in Figure 4, the place and time interval of a performance are described by `core:Place` and `core:TimeInterval` – involving one or more music artists (optionally, with a specific role). A performance may also create a new `mm:MusicEntity` if, e.g., the execution differs significantly from the original version.

A Music Entity can also be recorded by means of a `mm:RecordingProcess`, which is a subclass of a `mm:CreativeProcess`. This makes it possible to describe information about both the production (e.g., producers) and the technical aspects of it (e.g., sound engineer, equipment used). The recording process produces a `mm:Recording`, which is contained in a `mm:Release`.

Information about the broadcasting of a recording is modelled through the `mm:BroadcastingSituation` class (an instance of the Situation ODP [48]), which describes when and where the song was broadcast, and by which broadcaster (`mm:Broadcaster`).

3.2.4 Publishing and licensing information

The `mm:PublicationSituation` class describes information about the publication of a release, which is common to the publication of a `mm:Score` (c.f. Figure 4). For both a release and a score, it describes when and where they were published, and by a `mm:Publisher`.

Licence information is described by the `mm:License` class, which applies to records, releases and scores.

3.2.5 Modelling links and integrating provenance

We propose a pattern based on *RDF** [49] to describe the provenance at different levels (Figure 5). The use of RDF* is particularly useful for this purpose, as it allows to embed provenance information to every triple in the dataset. This simplifies and streamlines the model, eliminating the need for n-ary relations or reification for each triple.

The proposed pattern is straightforward and comprises the class `core:Reference`, which describes the source of the reference (using the class `core:Source`) and the method used to obtain the annotation (using the class `core:SourceMethod`). Additionally, the datatype properties `core:confidence` and `core:retrievedOn` describe the confidence of the annotation and the date it was produced, respectively.

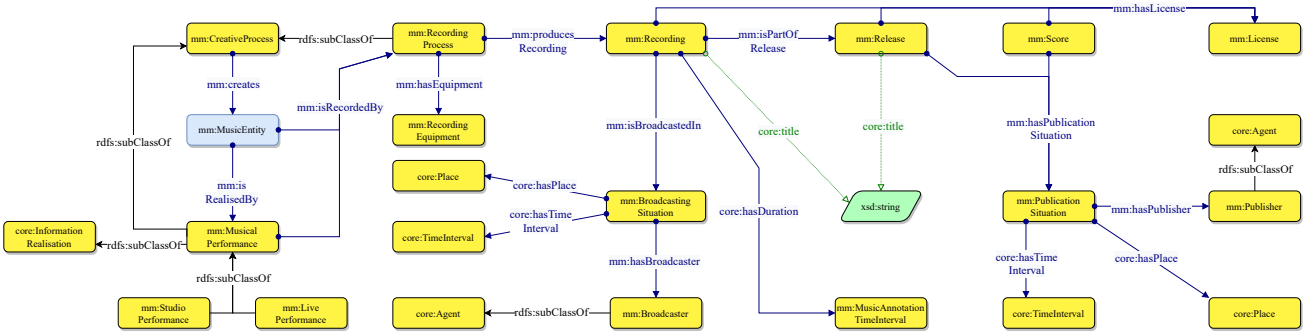


Figure 4. Describing performance, recording, broadcasting, publication, and licensing.

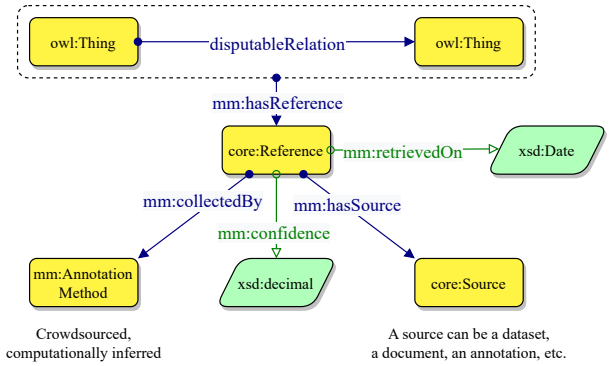


Figure 5. Our pattern to describe provenance with RDF*.

3.3 Conversion rules and code support

To facilitate the reuse of Music Meta and its data conversion into OWL/RDF Knowledge Graphs, we developed `PyMusicMeta` – a library to map arbitrary music metadata into RDF triples. This enables a practical and scalable workflows for data lifting to create Music KGs without expert knowledge of our ontological model. The library is developed in Python as an extension of RDF-Lib [50].

With each triple, `PyMusicMeta` adds alignments to the supported schema whenever possible. For example, the pseudo triple `<DavidBowieURI, rdf:type, mm:Musician>` in Music Meta will be complemented with `<DavidBowieURI, rdf:type, http://purl.org/ontology/mo/MusicArtist>` for Music Ontology, `<DavidBowieURI, rdf:type, http://erlangen-crm.org/E21_Person>` for DOREMUS (via the Erlangen Conceptual Reference Model [51]) and `<DavidBowieURI, rdf:type, https://www.wikidata.org/wiki/Q639669>` for Wikidata; to achieve interoperability of the Music KG.

4. VALIDATION AND ADOPTION

Following the XD methodology (c.f. Section 3.1), we validate Music Meta against the competency questions (CQs) driving its design. In this context, testing consists in formulating logical statements for each competency question – using the ontology as a formal model. Logical statements are encoded as SPARQL queries to evaluate the model. Examples of tested CQs include "*In which time interval did the creation process take place?*" and "*Which is the language of the name/alias of a music artist?*". The complete list of CQs, together with their correspondent SPARQL queries can be found in the project's repository. This also contributes a test framework where the ontology is automatically tested using the available SPARQL queries [52], whenever changes occur or new requirements are supported in future versions of Music Meta.

Music Meta has already been used in ChoCo [40], the largest Harmony KG to date, obtained from the integration of 18 MIR datasets [4]. The ontology has also been *specialised* for folk metadata (Tunes Ontology) and *extended* to describe music datasets (CoMeta Ontology). All ontologies are part of the Polifonia Ontology Network (PON) and can be found at `https://github.com/polifonia-project/ontology-network`. We also provide documentation, examples, and tutorials [5].

5. CONCLUSIONS

The interoperability of metadata is an essential requirement for the integration of music datasets, which is currently hampered by the specificity of existent ontologies.

Our work addresses interoperability requirements for the design of the Music Meta ontology – a rich and flexible semantic model for (Western) music metadata across different genres and periods, for various stakeholders and music datasets. The model is based on the Information-Realisation ontology design pattern, allowing to reduce complexity while maintaining alignment to other ontologies (Music Ontology, DOREMUS). We validate Music Meta following the XD methodology, to demonstrate the support of requirements collected from various stakeholders (music analysts, archivists, musicologists, and data engineers). The model has modular design – allowing users to describe music data depending on their specificity and type, while providing provenance support through RDF*.

We are extending the evaluation of Music Meta across cultural heritage and music industry datasets, while working with our stakeholders to specialise the model for the integration and release of Music Knowledge Graphs.

[4] `https://github.com/smashub/choco`
[5] `https://polifonia-project.github.io/ontology-network/`

Acknowledgements This project has received funding from the European Union's H2020 research and innovation programme under grant agreement No 101004746. The authors also acknowledge Philippe Rigaux, Peter van Kranenburg, Marco Gurrieri, and Mari Wigham for their support and feedback throughout the design of Music Meta.

6. REFERENCES

[1] T. Bottini, V. A. Carriero, J. Carvalho, P. Cathé, F. Ciroku, E. Daga, M. Daquino, A. Davy-Rigaux, M. Guillotel-Nothmann, Gurrieri, P. van Kemenade, E. Marzi, A. Meroño Peñuelala, P. Mulholland, E. Musumeci, V. Presutti, and A. Scharnhorst, "D1.1 roadmap and pilot requirements 1st version," EU Commission, The Polifonia consortium, Tech. Rep., 2021.

[2] F. Pachet, "Knowledge management and musical metadata," *Idea Group*, vol. 12, 2005.

[3] C. Sitonio and A. Nucciarelli, *The impact of blockchain on the music industry*. Calgary: International Telecommunications Society (ITS), 2018.

[4] S. Giannoulakis, N. Tsapatsoulis, and N. Grammalidis, "Metadata for intangible cultural heritage," in *Proceedings of the 13th international joint conference on computer vision, imaging and computer graphics theory and applications (VISAPP 2018)*, 2018, pp. 634–645.

[5] B. de Miguel-Molina and R. Boix-Doménech, "Introduction: Music, from intangible cultural heritage to the music industry," *Music as Intangible Cultural Heritage: Economic, Cultural and Social Identity*, pp. 3–8, 2021.

[6] T. Berners-Lee, J. Hendler, and O. Lassila, "The semantic web," *Scientific american*, vol. 284, no. 5, 2001.

[7] O. Lassila, R. R. Swick *et al.*, "Resource description framework (RDF) model and syntax specification," 1998.

[8] N. Corthaut, S. Govaerts, K. Verbert, and E. Duval, "Connecting the Dots: Music Metadata Generation, Schemas and Applications." in *ISMIR*, 2008, pp. 249–254.

[9] Y. Raimond, S. A. Abdallah, M. B. Sandler, and F. Giasson, "The Music Ontology." in *ISMIR*, vol. 2007. Vienna, Austria, 2007, p. 8th.

[10] M. Achichi, R. Bailly, C. Cecconi, M. Destandau, K. Todorov, and R. Troncy, "Doremus: Doing reusable musical data," in *ISWC: International Semantic Web Conference*, 2015.

[11] Y. Raimond and M. Sandler, "Evaluation of the music ontology framework," in *The Semantic Web: Research and Applications: 9th Extended Semantic Web Conference, ESWC 2012, Heraklion, Crete, Greece, May 27-31, 2012. Proceedings 9*. Springer, 2012.

[12] P. Choffé and F. Leresche, "DOREMUS: connecting sources, enriching catalogues and user experience," in *24th IFLA World Library and Information Congress*, 2016, pp. 1–20.

[13] M. Buffa, E. Cabrio, M. Fell, F. Gandon, A. Giboin, R. Hennequin, F. Michel, J. Pauwels, G. Pellerin, M. Tikat *et al.*, "The WASABI dataset: cultural, lyrics and audio analysis metadata about 2 million popular commercially released songs," in *The Semantic Web: 18th International Conference, ESWC 2021, Virtual Event, June 6–10, 2021, Proceedings 18*. Springer, 2021, pp. 515–531.

[14] M. Á. Rodríguez-García, L. O. Colombo-Mendoza, R. Valencia-García, A. A. Lopez-Lorca, and G. Beydoun, "Ontology-based music recommender system," in *Distributed Computing and Artificial Intelligence, 12th International Conference*. Springer, 2015, pp. 39–46.

[15] A. Adamou, S. Brown, H. Barlow, C. Allocca, and M. d'Aquin, "Crowdsourcing Linked Data on listening experiences through reuse and enhancement of library data," *International Journal on Digital Libraries*, vol. 20, no. 1, pp. 61–79, 2019.

[16] P. van Kranenburg, B. Janssen, A. Volk *et al.*, "The Meertens tune collections: The annotated corpus (mtc-ann) versions 1.1 and 2.0. 1," *Meertens Online Reports*, vol. 2016, no. 1, 2016.

[17] M. Daquino, V. Pasqual, and F. Tomasi, "Knowledge Representation of digital Hermeneutics of archival and literary Sources," *Knowledge Representation of digital Hermeneutics of archival and literary Sources*, pp. 59–76, 2020.

[18] M. Daquino, V. Pasqual, F. Tomasi, and F. Vitali, "Expressing Without Asserting in the Arts," in *CEUR WORKSHOP PROCEEDINGS*, vol. 3160, 2022.

[19] S. M. Rashid, D. De Roure, and D. L. McGuinness, "A Music Theory Ontology," in *Proceedings of the 1st International Workshop on Semantic Applications for Audio and Music*, ser. SAAM '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 6–14.

[20] J. Jones, D. de Siqueira Braga, K. Tertuliano, and T. Kauppinen, "MusicOWL: The Music Score Ontology," in *Proceedings of the International Conference on Web Intelligence*, ser. WI '17. New York, NY, USA: Association for Computing Machinery, 2017.

[21] S. S.-s. Cherfi, C. Guillotel, F. Hamdi, P. Rigaux, and N. Travers, "Ontology-Based Annotation of Music Scores," in *Proceedings of the Knowledge Capture Conference*, ser. K-CAP 2017. New York, NY, USA: Association for Computing Machinery, 2017.

[22] E. M. Sanfilippo and R. Freedman, "Ontology for analytic claims in music," in *New Trends in Database and Information Systems*, S. Chiusano, T. Cerquitelli, R. Wrembel, K. Nørvåg, B. Catania, G. Vargas-Solar, and E. Zumpano, Eds. Cham: Springer International Publishing, 2022, pp. 559–571.

[23] G. Fazekas, Y. Raimond, K. Jacobson, and M. Sandler, "An overview of Semantic Web activities in the OMRAS2 project," *Journal of New Music Research*, vol. 39, 12 2010.

[24] G. Fazekas and M. B. Sandler, "The Studio Ontology 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.

[25] A. Allik, G. Fazekas, and M. B. Sandler, "An Ontology for Audio Features," in *Proceedings of the 17th International Society for Music Information Retrieval Conference, ISMIR 2016, New York City, United States, August 7-11, 2016*, 2016.

[26] T. Wilmering, G. Fazekas, and M. B. Sandler, "The Audio Effects Ontology," in *Proceedings of the 14th International Society for Music Information Retrieval Conference, ISMIR 2013, Curitiba, Brazil, November 4-8, 2013*, 2013.

[27] Òscar Celma and X. Serra, "FOAFing the music: Bridging the semantic gap in music recommendation," *Journal of Web Semantics*, vol. 6, no. 4, pp. 250–256, 2008, semantic Web Challenge 2006/2007. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1570826808000711

[28] S. Song, M. Kim, S. Rho, and E. Hwang, "Music Ontology for Mood and Situation Reasoning to Support Music Retrieval and Recommendation," in *2009 Third International Conference on Digital Society*, 2009, pp. 304–309.

[29] B. Fields, K. R. Page, D. D. Roure, and T. Crawford, "The segment ontology: Bridging music-generic and domain-specific," in *Proceedings of the 2011 IEEE International Conference on Multimedia and Expo, ICME 2011, 11-15 July, 2011, Barcelona, Catalonia, Spain*. IEEE Computer Society, 2011.

[30] N. Harley and G. Wiggins, "An ontology for abstract, hierarchical music representation," in *Demo at the 16th International Society for Music Information Retrieval Conference (ISMIR 2015), Malaga, Spain*, 2015.

[31] K. Jacobson, Y. Raimond, and M. B. Sandler, "An Ecosystem for Transparent Music Similarity in an Open World," 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. 33–38. [Online]. Available: http://ismir2009.ismir.net/proceedings/OS1-2.pdf

[32] V. A. Carriero, F. Ciroku, J. de Berardinis, D. S. M. Pandiani, A. Meroño-Peñuela, A. Poltronieri, and V. Presutti, "Semantic Integration of MIR Datasets with the Polifonia Ontology Network," in *ISMIR Late Breaking Demo*, 11 2021.

[33] L. Turchet, F. Antoniazzi, F. Viola, F. Giunchiglia, and G. Fazekas, "The internet of musical things ontology," *Journal of Web Semantics*, vol. 60, p. 100548, 2020.

[34] J. de Berardinis, A. M. Penuela, A. Poltronieri, and V. Presutti, "The Music Annotation Pattern," in *The Semantic Web–ISWC 2022 21st International Semantic Web Conference: 13th Workshop on Ontology Design and Patterns (WOP2022)*, 2022.

[35] E. J. Humphrey, J. Salamon, O. Nieto, J. Forsyth, R. M. Bittner, and J. P. Bello, "JAMS: A JSON Annotated Music Specification for Reproducible MIR Research." in *ISMIR*, 2014, pp. 591–596.

[36] A. Poltronieri and A. Gangemi, "The HaMSE Ontology: Using Semantic Technologies to support Music Representation Interoperability and Musicological Analysis," *arXiv preprint arXiv:2202.05817*, 2022.

[37] D. Lewis, E. Shibata, M. Saccomano, L. Rosendahl, J. Kepper, A. Hankinson, C. Siegert, and K. Page, "A model for annotating musical versions and arrangements across multiple documents and media," 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. 10–18. [Online]. Available: https://doi.org/10.1145/3543882.3543891

[38] A. Meroño-Peñuela, R. Hoekstra, A. Gangemi, P. Bloem, R. de Valk, B. Stringer, B. Janssen, V. de Boer, A. Allik, S. Schlobach, and K. Page, "The MIDI Linked Data Cloud," in *The Semantic Web – ISWC 2017*. Cham: Springer International Publishing, 2017.

[39] P. Lisena, A. Meroño-Peñuela, and R. Troncy, "MIDI2vec: Learning MIDI embeddings for reliable prediction of symbolic music metadata," *Semantic Web*, vol. 13, no. 3, pp. 357–377, 2022.

[40] J. de Berardinis, A. Meroño-Peñuela, A. Poltronieri, and V. Presutti, "ChoCo: a Chord Corpus and a Data Transformation Workflow for Musical Harmony Knowledge Graphs," in *Manuscript under review*, 2023.

[41] ——, "The Harmonic Memory: a Knowledge Graph of harmonic patterns as a trustworthy framework for computational creativity," in *Proceedings of the ACM Web Conference 2023 (WWW '23), April 30-May 4, 2023, Austin, TX, USA*, 2023.

[42] J. de Berardinis, V. A. Carriero, N. Jain, N. Lazzari, A. Meroño-Peñuela, A. Poltronieri, and V. Presutti, "The polifonia ontology network: Building a semantic backbone for musical heritage," in *Manuscript under review*, 2023.

[43] E. Blomqvist, K. Hammar, and V. Presutti, "Engineering Ontologies with Patterns-The eXtreme Design Methodology." *Ontology Engineering with Ontology Design Patterns*, no. 25, 2016.

[44] V. A. Carriero, M. Daquino, A. Gangemi, A. G. Nuzzolese, S. Peroni, V. Presutti, and F. Tomasi, "The Landscape of Ontology Reuse Approaches," in *Applications and Practices in Ontology Design, Extraction, and Reasoning*, ser. Studies on the Semantic Web, G. Cota, M. Daquino, and G. L. Pozzato, Eds. Amsterdam: IOS Press, 2020, vol. 49, pp. 21–38.

[45] A. Gangemi and S. Peroni, "The Information Realization Pattern," in *Ontology Engineering with Ontology Design Patterns - Foundations and Applications*, ser. Studies on the Semantic Web, P. Hitzler, A. Gangemi, K. Janowicz, A. Krisnadhi, and V. Presutti, Eds. IOS Press, 2016, vol. 25, pp. 299–312. [Online]. Available: https://doi.org/10.3233/978-1-61499-676-7-299

[46] "Genius website," https://www.ifla.org, accessed: 2023-04-14.

[47] J. Riley, "Application of the Functional Requirements for Bibliographic Records (FRBR) to Music." in *ISMIR*, 2008, pp. 439–444.

[48] A. Gangemi, N. Guarino, C. Masolo, A. Oltramari, and L. Schneider, "Sweetening ontologies with dolce," in *Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web*, A. Gómez-Pérez and V. R. Benjamins, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 166–181.

[49] O. Hartig, "Foundations of RDF* and SPARQL* (An Alternative Approach to Statement-Level Metadata in RDF)," in *Alberto Mendelzon Workshop on Foundations of Data Management*, 2017.

[50] C. Boettiger, *rdflib: A high level wrapper around the redland package for common rdf applications*, 2018. [Online]. Available: https://doi.org/10.5281/zenodo.1098478

[51] J. Merges, M. Scholz, and G. Goerz, "Erlangen implementation of frbroo," in *CIDOC 2012*, 2012.

[52] A. Fernández-Izquierdo, "Ontology testing based on requirements formalization in collaborative development environments." in *DC@ ISWC*, 2017.

POLAR MANHATTAN DISPLACEMENT: MEASURING TONAL DISTANCES BETWEEN CHORDS BASED ON INTERVALLIC CONTENT

Jeff Miller **Johan Pauwels** **Mark Sandler**
Centre for Digital Music
Queen Mary University of London
j.k.miller@qmul.ac.uk j.pauwels@qmul.ac.uk mark.sandler@qmul.ac.uk

ABSTRACT

Large-scale studies of musical harmony are often hampered by lack of suitably labelled data. It would be highly advantageous if an algorithm were able to autonomously describe chords, scales, etc. in a consistent and musically informative way. In this paper, we revisit tonal interval vectors (TIVs), which reveal certain insights as to the interval and tonal nature of pitch class sets. We then describe the qualities and criteria required to comprehensively and consistently measure displacements between TIVs. Next, we present the Polar Manhattan Displacement (PMD), a compound magnitude and phase measure for describing the displacements between pitch class sets in a tonally-informed manner. We end by providing examples of how PMD can be used in automated harmonic sequence analysis over a complex chord vocabulary.

1. INTRODUCTION

Attempts to autonomously label and analyse harmonic sequences in music constitute some of the longest-standing challenges in music information research (MIR) [1]. Various strategies have been applied to chord sequence identification, including information theory [2], graph theory [3–5], and predictive methods such as Hidden Markov Models [6]. Much attention has been given to developing geometric models of musical distance [7–11].

First proposed by Lewin in 1959 [12] and later, in 2007, [13], the discrete Fourier transform can be applied to collections of pitch classes to produce Tonal Interval Vectors (TIVs), which can be used to describe the tonal qualities of chords and pitch class profiles (PCPs) by revealing their constituent intervals and tonal structures [14]. Yust et al. [15] have employed the Fourier transform as a form of cluster analysis on large groups of weighted PCP sets, while Tymoczko and Yust [16] have explored the relationship between voice-leading and Fourier analysis. Other previous work has focused primarily on the magnitude components of TIVs [17, 18], which offer a useful but incomplete picture of intervallic content and tonal quality, as transposition is ignored and certain chord types such as major and minor cannot be disambiguated [19]. Furthermore, many existing methods of measuring distance between TIVs are problematic as they are restricted to pairs of chords and are inconsistent when groups of three or more chords are considered. Additionally, many musical distances falter because they do not capture the directional nature of musical harmonic tension or are adversely affected by enharmonic spellings and conflicting chord vocabularies.

We present the Polar Manhattan Displacement (PMD), a method of describing component-wise directional distance (i.e., displacement) between TIVs which utilises both magnitude and phase information. We demonstrate PMD within the context of the 12-tone equally tempered symbolic domain.

PMD addresses all 4,095 possible pitch class combinations and thus can be applied to any chord vocabulary. PMD offers a consistent displacement measure between chord types (e.g., major7, diminished7, etc.) regardless of transposition or the complexity of the chord vocabulary employed. PMD also measures the intervallic displacement between chords, regardless of chord type. In both cases, displacement measurements are unaffected by transposition of an entire sequence, allowing PMD to identify relative harmonic movements regardless of local key structure.

To demonstrate the utility of PMD, we employ a robust chord vocabulary of 13 chord types including triads, 7th chord types, and suspended chords, as well as chromatic and whole tone scales. We measure displacements amongst these chord types and transpositions, and close by presenting example applications of PMD to automated harmonic analysis and discuss potential applications to other areas of music informatics.

2. BACKGROUND

2.1 Pitch Class Profiles & common musical terms

A pitch class profile (PCP) is a vector of 12 binary values, each representing the categorical presence of its corresponding pitch class in the relevant musical context. This context is set within the time domain; unless otherwise specified, we shall be considering notes which occur si-

© J. Miller, J. Pauwels, and M. Sandler. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).
Attribution: J. Miller, J. Pauwels, and M. Sandler, "Polar Manhattan Displacement: measuring tonal distances between chords based on intervallic content", in *Proc. of the 24th Int. Society for Music Information Retrieval Conf.,* Milan, Italy, 2023.

multaneously. Commonly occurring collections of simultaneous pitch events may be referred to as 'chords'. A succession of notes occurring sequentially in ascending or descending pitch order is commonly referred to as a 'scale'. When the time window is increased further and some statistical weighting or filtering is considered, the dominant members of the pitch class set may imply a 'key'.

Throughout this paper, when referring to the collection of possible PCPs, we exclude the empty PCP [0,0,0,0,0,0,0,0,0,0,0,0] which represents an absence of all pitch classes, resulting in $2^{12} - 1 = 4,095$ possible PCPs. To improve clarity, the terms 'chord' and 'scale' shall be considered interchangeable with 'PCP' unless otherwise noted. The term 'chord type' refers to the quality of a chord (e.g., major, minor, etc.) regardless of the chord transposition. A 'chord', however, is a specific combination of root and chord type, such as A_{min} or F_{maj7}.

2.2 Chord vocabulary

For transcription and harmonic analysis purposes, it is useful to focus on the subset of PCPs which correspond to certain chord types. Within the domain of all 4,095 PCPs, the choice of chord vocabulary can be a fairly arbitrary decision. Often, smaller vocabularies of simple chords are chosen to simplify experiments and boost performance scores. There is a risk that an over-simplified chord vocabulary can reduce the usefulness of an analytic system, so it is advantageous that a suitably complex chord vocabulary is employed.

For our examples, we restrict ourselves to a vocabulary of 13 chord types (including, by extension, two scales), but the PMD can be applied to any pitch class profile. Our vocabulary included triads: *major, minor, diminished, augmented, suspended4*; tetrads: *major7, minor7, dominant7, diminished7, half-diminished7, minor/major7*, and scales: *chromatic, wholetone*. Note that some PCPs can be described using different chord types depending on context, thus some chord types, such as maj6, min6 and sus2, are synonymous with other chords already listed in our vocabulary. Regardless of the labels assigned to such synonymous chords, the source PCPs remain the same. For example, PCP [1,0,0,0,1,0,0,1,0,1,0,0] could be described as either C_{maj6} or A_{min7}. Labelling of chords is highly dependent on context and annotator subjectivity. As PMD operates on the basis of underlying PCPs, it is unaffected by such discrepancies in annotation.

2.3 DFTs and Tonal Interval Vectors

By applying a discrete Fourier transform (DFT) to a pitch class profile, we can decompose the PCP into a series of constituent intervallic components. Each of these components will describe the degree to which a particular interval is present within the PCP. Only components F_1 – F_6 are required to provide a complete representation, since components $F_7 - F_11$ are redundant. Additionally, F_0 reveals the cardinality of the pitch class set; its value is useful for normalisation and allows us to compare any pair of PCPs regardless of the number of pitches present in either.

The resulting 6-dimensional complex vector is referred to as a tonal interval vector (TIV). Scaling may be applied to the various components – often for normalisation purposes, but also in an attempt to more accurately depict perceived cognitive distances between the various interval types within a musical context. [18]

It is worth noting that when generating TIVs, the DFT is applied to the symbolic pitch class vectors and not to audio data. The purpose of the DFT and resulting TIV is to discover the manner in which the pitch classes divide an octave into various musical intervals, and to describe the strength and evenness of each intervallic division.

2.4 Mapping PCPs to TIV space

Each of the 4,095 possible non-empty PCPs produces a unique tonal interval vector. The TIV space therefore is an injection of PCP space: each TIV can be mapped unambiguously to a corresponding pitch class profile. Furthermore, each 6D complex TIV can be represented as a 12-dimensional real-valued vector by converting the complex values of the TIV into magnitude and phase values.

The magnitude and phase values of the recast TIV vector can be represented as a set of 6 tuples (m, p), where m is an unbounded positive real value, and p is a real value $p \in R$ such that $-\pi \leq p \leq \pi$. We will refer to a magnitude and phase tuple (m, p) as a MagPhase tuple. The set of 6 tuples describing Fourier components F_1 - F_6 of a TIV will be referred to as a MagPhase vector.

It has been shown [14,17,18] that converting the Fourier coefficients from complex values to real magnitude and phase values reveals direct correlations to chord type, transposition, and interval ordinality.

2.5 Descriptive properties of Magnitude and Phase

Each TIV Fourier component F_n can be associated with a tonal interval and its complementary inverse interval, e.g. F_1 is associated with the presence of both minor 2nd and major 7th intervals, etc. [14] The magnitude value of a TIV component reflects how strongly the associated interval occurs in the source PCP. For example, F_3 is associated with the presence of major 3rds. Augmented triads (which are composed of nothing but major 3rds) have a maximal F_3 magnitude, whereas diminished 7th chords, which contain no major 3rds, have an F_3 magnitude of 0. Most chords are composed of several interval types and thus have non-zero magnitudes for all 6 components.

Chords containing the same collection of intervals will share a common magnitude profile. This allows some degree of chord classification based on magnitude values. However, magnitude profiles do not convey the ordinal placement of the intervals within the source PCP, meaning that chord types with identical sets of intervals cannot be disambiguated. For example, major and minor triads both contain one of each of the following atomic intervals: $\{m3, M3, P4\}$ and thus will have identical magnitude profiles. For the same reason, magnitude profiles alone do not encode information about the transposition (i.e., the root) of a source PCP, making it impossible to differentiate

C_{dom7} from F_{dom7} or G_{dom7}, for example. By incorporating phase information, both of these shortcomings can be addressed.

3. MOTIVATION

Each component F_1 - F_6 of a TIV describes the strength and position of a particular intervallic quality. By measuring the differences between TIV dimensions separately, the intervallic content of TIVs could be exposed and compared. It is reasonable to consider how distances between them might be useful in describing the relationship between their respective PCPs, as well as enabling computational modelling of musical chords and chord sequences, and the automated study of large corpora.

Distance measures are by definition non-directional. However, harmonic transitions are often asymmetrical in practice, e.g., there is a difference harmonically between a $V7 \rightarrow I$ transition and a $I \rightarrow V7$ transition. By measuring displacement between TIVs, both the distance and direction between them are exposed, which increases the descriptive power of the measure.

To extend the difference measure beyond comparisons of isolated pairs of chords, it would be highly advantageous for the measurements to exhibit collinearity under the addition operation. For example, if two chords of the same type a semitone apart have a particular displacement value, it stands to reason that two chords a whole tone apart should have a displacement double that value. Crucially, this would allow multiple displacements to be summed, making it possible to build sequences and represent various progressions between two chords consistently.

Finally, if differences in chord types and intervallic distances between chords were decoupled, displacement between chords could be visualised on a grid whose orthogonal axes represented each quality. The additive collinearity discussed above would ensure that displacement values could be summed consistently, regardless of which displacements were measured or in what order.

This would allow chord sequences to be represented as traversals across a tonal grid. An example of such a grid is displayed in Figure 1. Chord transitions such as $V7 \rightarrow I$ would have the same displacement regardless of the transposition of the pair. For example, $G_7 \rightarrow C_{maj}$ and $A_7 \rightarrow D_{maj}$ would have an identical displacement, despite being in different keys. Such transpositional invariance would enable an algorithm to catalogue and identify commonly occurring functional sequences.

4. POLAR MANHATTAN DISPLACEMENT

To satisfy the criteria outlined in section 3, we propose the Polar Manhattan Displacement (PMD). PMD measures the component-wise displacement between two TIVs by calculating the differences in the magnitude and phase of each component. As each component F_k describes some aspect of the tonal nature of the source chord, this allows us to measure the displacements between chords in an intervallically informed way.

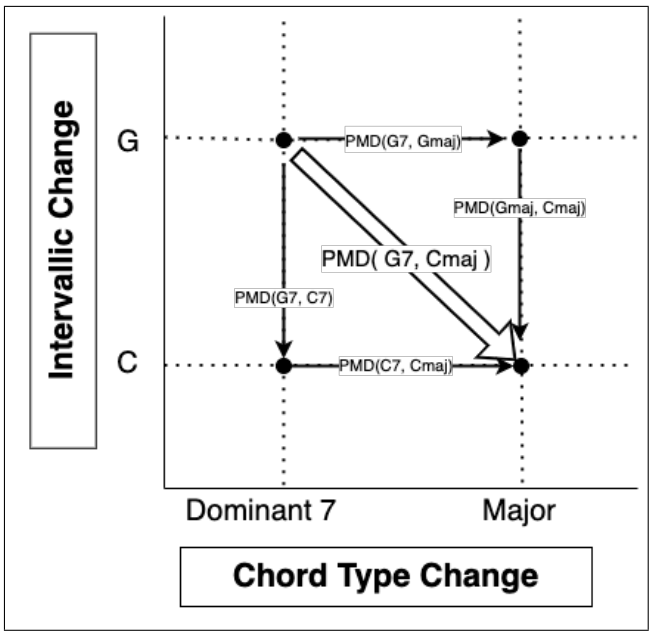


Figure 1. Decoupled chord type and intervallic displacements for the chord transition $G_7 \rightarrow C_{maj}$. There are three routes from $G_7 \rightarrow C_{maj}$; The PMDs of each arrow are summed. All routes result in the same final displacement value. Note that the direction of the progression is significant.

PMD thus borrows from L-1 Manhattan distance but extends it to measure the directional displacements between corresponding magnitudes and phases in each dimension of the polar representation of the TIV plane. The differences are signed, meaning that PMD reflects displacement rather than formal distance. This is advantageous within a musical context because tonal harmony is directional; for example the transition $(C_{maj} \rightarrow G_7)$ has a markedly different tonal impact than $(G_7 \rightarrow C_{maj})$.

4.1 Magnitude processing

The processing of magnitude values is straightforward as it involves only scaling and subtraction. After a TIV is extracted from a PCP, the magnitudes of F_1 to F_6 are divided by the value of F_0. This normalises all magnitudes to the same scale and allows comparison of TIVs having different numbers of pitches in their source PCPs. Further scaling of each component magnitude may be applied to improve the perceptual basis of the space [18] [20]. However, perceptual scaling has not been applied in our study. Following the application of normalisation and scaling, magnitude values are simply subtracted such that the difference between the magnitudes of TIVs U and V is

$$\text{Disp}_{mag}(U \rightarrow V) = (V_{mag} - U_{mag}) \tag{1}$$

4.2 Phase processing

Phase values are simply subtracted in a similar fashion to that presented in section 4.1 such that the angular displace-

Chord type	M1	M2	M3	M4	M5	M6
Maj7						X
Min7		X				X
Dim7	X	X	X		X	X
Aug	X	X		X	X	
Sus4				X		
Chrom	X	X	X	X	X	X
WholeTone	X	X	X	X	X	

Table 1. Some common chord types with zero-magnitude components. 'X' indicates a zero-magnitude vector.

ment between TIVs U and V is

$$\mathrm{Disp}_{phase}(U \to V) = (V_{phase} - U_{phase}) \tag{2}$$

The cyclic nature of phase information necessitates additional processing. Angular phase values θ need to be cyclically wrapped into the interval $-\pi < \theta \leq \pi$ after all additive and subtractive operations on phase values.

4.3 Definition of PMD

Having defined Disp_{mag} in Eq. (1) and Disp_{phase} in Eq. (2), we can now present the definition of the Polar Manhattan Displacement. PMD is created by concatenating the 6D magnitude and phase displacement vectors Disp_{mag} and Disp_{phase} into one 12D vector.

Given chords (P_j,Q_k) and their corresponding TIVs (U,V),

$$\mathrm{PMD}(P_j, Q_k) = \begin{bmatrix} \mathrm{Disp}_{mag}(U \to V) \\ \mathrm{Disp}_{phase}(U \to V) \end{bmatrix} \tag{3}$$

4.4 Zero-magnitude handling

Some PCPs have various TIV components with zero-valued magnitudes. This indicates that the corresponding interval type is absent from the source PCP. While this is of no consequence when calculating magnitude difference (for the magnitude is simply 0), the associated phase of these components is effectively undefined, complicating the calculation of differences between phase values. In our vocabulary, the following chord types contain one or more zero-magnitude component vectors: $\{maj7, min7, dim7, aug, sus4, chrom, wltn\}$. Table 1 details these zero-magnitude chords and their affected components.

We employ the convention that these phase values are considered to be 0. This preserves the additive properties of PMD and allows multiple displacement values to be added together consistently.

4.5 Investigating displacements of type and interval

As indicated in section 3, it would be highly advantageous (and musically interesting) to find a way to distinguish the degree of displacement due to changes in chord types versus shifts of interval. In this section, we examine to what extent such a decoupling is possible.

For convenience, we present the terms Disp_{type} (representing the displacement between chord types) and Disp_{intv} (representing the intervallic displacement between any two chords of the same type). We define each with a functional representation, then provide an example of the relevant function in use. We employ the following convention to represent a movement from one chord to another:

$$(P_j \to Q_k) \tag{4}$$

where j and k denote chord types from our vocabulary and P and Q denote two arbitrary chord roots. For example, the following represents a movement from G_7 to C_{maj}:

$$(G_7 \to C_{maj}) \tag{5}$$

where $j = dom7$, $k = maj$, $P = G$, and $Q = C$. Note that the direction of the chord transition is significant.

4.5.1 *Type-based displacement*

A type-based displacement Disp_{type} can be derived by calculating the PMD of two chords (P_j, Q_k) having the same root (i.e., $P = Q$) but different types ($j \neq k$). Formally,

$$\mathrm{Disp}_{type}(j, k) = \mathrm{PMD}(P_j, P_k) \tag{6}$$

where $j \neq k$.

Significantly, these displacement values are consistent for all pairs of chord types (j, k) regardless of the values of root (P). As an example, the PMD values corresponding to the displacement from a chromatic scale (j) to each of the chords in our vocabulary (k) are detailed in table 2.

4.5.2 *Intervallic displacement*

Likewise, the intervallic displacement Disp_{intv} between any two chords (P_j, Q_k) should ideally be consistent regardless of the chord types involved. In a similar fashion to the calculation of Disp_{type}, we calculate Disp_{intv} by calculating the PMD of two chords (P_j, Q_k) having different roots (i.e., $P \neq Q$) but identical types ($j = k$).

$$\mathrm{Disp}_{intv}(P, Q, j) = \mathrm{PMD}(P_j, Q_j) \tag{7}$$

where $P \neq Q$. Note that Disp_{intv} is a vector of real number values and is not expressed in semitones.

Crucially, the Disp_{intv} is largely – but not entirely – independent of chord type j. For all chord types j with uniquely non-zero TIV magnitude components, the Disp_{intv} in function of the root interval shift ($P \to Q$) expressed in semitones is shown in table 3. Chord types j that contain magnitudes of zero (as shown in table 1) have the same PMD as in table 3 for their non-zero components, but both magnitude and phase components of the PMD corresponding to the TIV components with value zero are also zero. By combining tables 1 and 3, the Disp_{intv} for all chord types in our vocabulary can be determined.

Comp	Chrom	Dim7	Wltn	Aug	Min7	Dom7	Maj7	Dim	HDim7	Major	MinMaj7	Minor	Susp4
M_1	0.00	0.00	0.00	0.00	0.18	0.13	0.13	0.33	0.13	0.17	0.25	0.17	0.24
M_2	0.00	0.00	0.00	0.00	0.00	0.25	0.43	0.33	0.25	0.33	0.25	0.33	0.67
M_3	0.00	0.00	0.00	1.00	0.50	0.35	0.71	0.33	0.35	0.75	0.79	0.75	0.33
M_4	0.00	1.00	0.00	0.00	0.50	0.66	0.25	1.00	0.66	0.58	0.25	0.58	0.00
M_5	0.00	0.00	0.00	0.00	0.68	0.48	0.48	0.33	0.48	0.64	0.25	0.64	0.91
M_6	0.00	0.00	1.00	1.00	0.00	0.50	0.00	0.33	0.50	0.33	0.50	0.33	0.33
P_1	0.00	0.00	0.00	0.00	0.52	1.31	0.26	-1.57	-0.26	-2.36	0.00	-1.31	3.14
P_2	0.00	0.00	0.00	0.00	0.00	1.05	0.53	0.00	1.05	0.00	0.00	-1.05	0.00
P_3	0.00	0.00	0.00	0.00	1.57	0.79	0.79	1.57	2.36	0.46	1.25	1.11	0.00
P_4	0.00	0.00	0.00	0.00	-1.05	-1.76	-2.09	0.00	-0.33	-1.57	0.00	-0.52	0.00
P_5	0.00	0.00	0.00	0.00	-0.52	0.26	1.31	-1.57	-1.31	0.79	0.00	-0.26	0.00
P_6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	3.14	3.14	3.14

Table 2. PM Displacements for each chord type from our vocabulary as end chord, starting from the chromatic scale and with the same root. Each column is a PMD vector representing the displacement from the chromatic scale chord type. To reverse the direction, invert the signs of the values.

Component	+1	+2	+3	+4	+5	+6	+7	+8	+9	+10	+11	+12
M_1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
M_2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
M_3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
M_4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
M_5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
M_6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
P_1	-0.52	-1.05	-1.57	-2.09	-2.62	3.14	2.62	2.09	1.57	1.05	0.52	0.00
P_2	-1.05	-2.09	3.14	2.09	1.05	0.00	-1.05	-2.09	3.14	2.09	1.05	0.00
P_3	-1.57	3.14	1.57	0.00	-1.57	3.14	1.57	0.00	-1.57	3.14	1.57	0.00
P_4	-2.09	2.09	0.00	-2.09	2.09	0.00	-2.09	2.09	0.00	-2.09	2.09	0.00
P_5	-2.62	1.05	-1.57	2.09	-0.52	3.14	0.52	-2.09	1.57	-1.05	2.62	0.00
P_6	3.14	0.00	3.14	0.00	3.14	0.00	3.14	0.00	3.14	0.00	3.14	0.00

Table 3. PM Displacements of each ascending transposition interval in semitones for chord types that have no zero-valued TIV magnitudes. Each column is a PMD vector. Note the symmetry of the column values; intervallic distances are constant, while the sign indicates the direction of transposition. To reverse the direction, (i.e., to transpose down) invert the signs of the values. Also, note that only the phase elements are affected by transposition.

Component	Tritone Substitution		$V_7 - I$	
	(G_7, Db_7)	(A_7, Eb_7)	(G_7, C_{maj})	(B_7, E_{maj})
M_1	0	0	0.043	0.043
M_2	0	0	0.083	0.083
M_3	0	0	0.392	0.392
M_4	0	0	-0.084	-0.084
M_5	0	0	0.161	0.161
M_6	0	0	-0.167	-0.167
P_1	3.142	3.142	0	0
P_2	0	0	0	0
P_3	3.142	3.142	-1.893	-1.893
P_4	0	0	2.285	2.285
P_5	3.142	3.142	0	0
P_6	0	0	3.142	3.142

Table 4. PMD comparisons of a) two tritone substitutions and b) two $V_7 - I$ sequences. Notice that within each pair the PMD values are identical. This indicates that any pair of chords separated by this displacement will constitute a tritone substitution pair or $V_7 - I$ sequence, respectively, regardless of the transposition.

5. PMD EXAMPLES

5.1 Example 1:Tritone substitution

It is well known within musical harmonic practice that certain chords may be substituted for one another to provide alternative or extended versions of an existing or expected harmony. A common example of this is the tritone substitution, wherein a dominant 7th chord can be replaced with a different dominant 7th whose root is a tritone (i.e., an augmented 4th or diminished 5th) away from the original root. The pitches acting as the 3rd and 7th of the original chord are retained, but their functions are swapped. The remaining 2 pitches of the first chord are replaced with other pitches. The overall function is of a new chord that retains the essential character of the original chord, but provides addition harmonic tension.

While issues of perceptual similarity are beyond the scope of the current study, PMD can provide an objective means of numerically describing such relationships. For example, consider the Polar Manhattan Displacements between each of these two tritone substitution pairs: (G_7, Db_7) and $(A_7, E\flat_7)$ as detailed in Table 4. Notice that the PMD values are identical. Any pair of chords separated by this displacement will constitute a tritone substitution pair.

5.2 Example 2: V7 - I detection

There are a number of MIR tasks involving chord estimation, transcription, and automated harmonic analysis that could benefit from the ability to autonomously identify certain chord progressions, particularly those which are harmonically significant. Traditionally, these tasks are hampered by lack of labelled data, inconsistent chord vocabularies, inter-annotator disagreement, etc. As PMD operates on unlabelled symbolic data, it could contribute to addressing such shortcomings and improving performance in automated transcription, labelling, and harmonic analysis. Table 4 describes the displacement between two different $(V_7 \rightarrow I)$ progressions and confirms that PMD can identify and encode the $(V_7 \rightarrow I)$ progression consistently, regardless of key or transpositional context.

6. CONCLUSIONS & FURTHER WORK

We began with a background review of pitch class profiles and some basic terms of musical harmonic structures. We then discussed tonal interval vectors: how discrete Fourier transforms can be applied to PCPs to create TIVs, how TIVs can be represented as vectors containing magnitude and phase values, and how those values describe some aspects of the intervallic construction of a chord. We proposed that it could be useful to measure displacements between these objects, and then described the properties necessary for a robust and self-consistent measure of displacement.

We then presented the Polar Manhattan Displacement, its fundamental components, and the processing required to calculate the measurement of magnitude and phase difference values. There was a brief description of our chord vocabulary and the need for suitable chord vocabularies, and a brief discussion of how to maintain transpositional invariance when dealing with non-existent magnitude vectors.

Having discussed the criteria for a suitable displacement measure, and detailed the functional components of our proposed measurement algorithm, we demonstrated how these components could be aggregated to create the Polar Manhattan Displacement measure. We then provided two examples of potential use cases of PMD, one involving the autonomous identification of tritone substitutions, and the other, $V_7 - I$ progressions.

Future technical work will involve evaluating the robustness of PMD when employed on audio data and at various scales of temporal granularity. It would be interesting to investigate extension of PMD to process non-binary PCPs, such as weighted PCPs and harmonic pitch class profiles. As the additive properties of PMD allow displacements to be summed, we would also like to extend the application of PMD to chord sequence modelling and analysis. Finally, we would like to deploy PMD as part of a large corpus study to investigate chord similarity and harmonic practice.

7. ACKNOWLEDGEMENTS

This research was partially supported by EPSRC grant EP/R512072/1 and the British Broadcasting Corporation through an Industrial CASE studentship in collaboration with the BBC Audio Research Partnership.

8. REFERENCES

[1] T. Fujishima, "Realtime Chord Recognition of Musical Sound: A System Using Common Lisp Music," pp. 464–467, 1999.

[2] M. T. Pearce, "The Construction and Evaluation of Statistical Models of Melodic Structure in Music Perception and Composition," Ph.D. dissertation, City University, London, 2006.

[3] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon, "Network Motifs: Simple Building Blocks of Complex Networks," *Science*, vol. 298, pp. 824 – 827, 2002.

[4] R. Milo, S. Itzkovitz, N. Kashtan, R. Levitt, S. Shen-Orr, I. Ayzenshtat, M. Sheffer, and U. Alon, "Superfamilies of Evolved and Designed Networks," *Science*, vol. 303, no. March, pp. 1538 – 1542, 2004.

[5] S. Itzkovitz, R. Milo, N. Kashtan, R. Levitt, A. Lahav, and U. R. I. Alon, "Recurring Harmonic Walks and Network Motifs in Western Music," *Advances in Complex Systems*, vol. 9, no. 1 & 2, pp. 121–132, 2006.

[6] C. Harte, M. Sandler, and M. Gasser, "Detecting harmonic change in musical audio," *Proceedings of the*

ACM International Multimedia Conference and Exhibition, pp. 21–26, 2006.

[7] E. Chew, "Towards a mathematical model of tonality," Ph.D. dissertation, 1999. [Online]. Available: http://www-rcf.usc.edu/{~}echew/papers/Dissertation2000/ec-dissertation.pdf

[8] ——, "Out of the Grid and Into the Spiral: Geometric Interpretations of and Comparisons with the Spiral-Array Model," *Computing in Musicology*, vol. 15, pp. 51–72, 2007.

[9] C. Harte, "Towards Automatic Extraction of Harmony Information from Music Signals," Ph.D. dissertation, Queen Mary University London, 2010.

[10] D. Tymoczko, "The geometry of musical chords," *Science*, vol. 313, no. 5783, pp. 72–74, 2006.

[11] ——, *A Geometry of Music: Harmony and Counterpoint in the Extended Common Practice*. New York, New York, USA: Oxford University Press, 2011.

[12] D. Lewin, "Re : Intervallic Relations between Two Collections of Notes," *Journal of Music Theory*, vol. 3, no. 2, pp. 298–301, 1959.

[13] ——, *Generalized Musical Intervals and Transformations*. Oxford University Press, 2007.

[14] J. D. Harding, "Applications of the Discrete Fourier Transform to Music Analysis," Ph.D. dissertation, Florida State University, 2021.

[15] J. Yust, J. Lee, and E. Pinsky, "A Clustering-Based Approach to Automatic Harmonic Analysis: An Exploratory Study of Harmony and Form in Mozart's Piano Sonatas," *Transactions of the International Society for Music Information Retrieval*, vol. 5, no. 1, pp. 113–128, 2022.

[16] D. Tymoczko and J. Yust, "Fourier Phase and Pitch-Class Sum," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11502 LNAI, pp. 46–58, 2019.

[17] G. Bernardes, D. Cocharro, C. Guedes, and M. E. P. Davies, "Conchord: An Application for Generating Musical Harmony by Navigating in a Perceptually Motivated Tonal Interval Space," *International Symposium on Computer Music Multidisciplinary Research*, pp. 1–16, 2015.

[18] G. Bernardes, D. Cocharro, M. Caetano, C. Guedes, and M. E. Davies, "A multi-level tonal interval space for modelling pitch relatedness and musical consonance," *Journal of New Music Research*, vol. 45, no. 4, pp. 281–294, 2016.

[19] A. Ramires, G. Bernardes, M. E. P. Davies, and X. Serra, "TIV.lib: An open-source library for the tonal description of musical audio," in *Proc. of the 23rd International Conference on Digital Audio Effects (DAFx-20)*, Vienna, 2020, pp. 8–13.

[20] I. Sha'ath, "Estimation of Key in Digital Music Recordings," Master's thesis, Birkbeck College, University of London, 2011.

Author Index

Acquilino, Alberto 684
Akama, Taketo 215
Alonso-Jiménez, Pablo 824
Amatov, Amantur 649
Antonacci, Fabio 138, 257
Aravind, R 56
Araújo, Felipe 619
Arteaga, Daniel 304
Arthur, Claire 239

Baelemans, Mariëlle 725
Baker, David John 817
Ballen, Gustavo A. 571
Barkan, Oren 642
Barnes, Connelly 190
Barthet, Mathieu 121
Bauer, Christine 482
Beguerisse-Díaz, Mariano 605
Benetos, Emmanouil 311, 343, 457
Berg-Kirkpatrick, Taylor 657
Bigo, Louis 508, 741
Bjare, Mathias Rose 810
Bogdanov, Dmitry 824
Borovik, Ilya 588
Borsan, Vanessa Nina 474, 545
Bozzon, Alessandro 612
Breebaart, Jeroen 304
Briot, Jean-Pierre 89
Buisson, Morgan 417
Bunks, Carey 757
Burgoyne, John Ashley 725, 817
Bükey, Irmak 733

Calvo-Zaragoza, Jorge 182, 319
Cancino-Chacón, Carlos Eduardo 848
Carriero, Valentina Anita 859
Carvalho, Luís 700
Castellanos, Francisco J. 106
Cengarle, Giulio 304
Chae, Yunkee 716
Chang, Wen-Whei 174
Chen, Ke 657
Chen, Siqi 765
Chen, Tsung-Ping 281
Chen, Wenhu 343
Chen, Xingran 457
Cheng, Tian 466
Cheung, Vincent K. M. 197
Chhel, Fabien 89
Choi, Ahyeon 492
Choi, Keunwoo 409
Clayton, Martin 21
Comanducci, Luca 257
Condit-Schultz, Nathaniel 239
Couturier, Louis 508
Crayencour, Helene C. 417
Cwitkowitz, Frank 676

Dai, Shuqi 765
Dannenberg, Roger B. 343, 457, 765
De Backer, Jos 247
de Berardinis, Jacopo 859
Deniffel, Simon 433
Devaney, Johanna 71
D'Hooge, Alexandre 741
Di Giorgi, Bruno 757
Ding, Yiwei 579
Dinnissen, Karlijn 482
Dixon, Simon 500, 757, 848
Doh, SeungHeon 409
Driedger, Jonathan 725
Duan, Zhiyao 676
Dubnov, Shlomo 657
Duguay, Michèle 71
Déguernel, Ken 741

Ehmann, Andreas 375
Eisenbrand, Friedrich 383
Elharar, Almog 642
Essid, Slim 417

Fazekas, György 166, 667
Ferraro, Andres 375
Finkelstein, Adam 190
Finkensiep, Christoph 383
Flexer, Arthur 47
Foscarin, Francesco 425
Foubert, Katrien 247
Fradet, Nathan 89
Fu, Jie 343, 457
Fujinaga, Ichiro 106, 684
Fujishima, Takuya 352
Furuya, Shinichi 197

Gallego, Antonio Javier 106
García, Hugo Flores 359
García-Portugués, Eduardo 802
Garrido-Munoz, Carlos 182
Gerstner, Wulfram 627
Giraud, Mathieu 474, 545
Gotham, Mark R. H. 98, 272
Goto, Masataka 129, 197, 398, 466, 561, 774, 782, 840
Gouyon, Fabien 375
Groult, Richard 474, 545
Guo, Yike 343, 457
Gutowski, Nicolas 89
Gyenge, Norbert 457

Hadjeres, Gaëtan 535
Haeberle, Matthieu 383
Hajič jr., Jan 571
Haki, Behzad 114
Hamasaki, Masahiro 561, 782
Han, Danbinaerin 440
Hankinson, Andrew 795
Harasim, Daniel 425
Hentschel, Johannes 516
Heyen, Frank 692
Hristova, Desislava 605
Hsiao, Yo-Wei 281
Hu, Patricia 297
Huang, Jen-Wei 174
Huang, Roy 765
Hung, Tzu-Yun 281

Ikemiya, Yukara 215

Jeon, Chang-Bin 716
Jeong, Dasaem 440, 708
Jiang, Junyan 231
Jiang, Yucong 367
Jin, Zeyu 190
Johnson, Max 98
Jordà, Sergi 114
Joung, Haesun 492

Kaneshiro, Blair 264
Karystinaios, Emmanouil 597, 848
Kearns, Eoin J. 391
Kepper, Johannes 795
Kim, Haven 774
Kim, Jaehun 375
Kim, Sehun 524
Kirchhoff, Holger 29
Koenigstein, Noam 642
Koo, Junghyun 716
Korzeniowski, Filip 619
Kotowski, Błażej 114
Krause, Michael 289, 433, 832
Kudinov, Mikhail 649
Kumar, Rithesh 359

Lamanov, Dmitry 649
Lattner, Stefan 448, 535, 810
Laufer, Moshe 642
Lecroq, Thierry 474
Lee, Cheuk Lun Isaac 114
Lee, Jin Ha 80
Lee, Jongpil 409
Lee, Joongseek 492
Lee, Kyogu 492, 716
Lerch, Alexander 579

Levé, Florence 508
Lewis, David 795
Li, Yizhi 343, 457
Liao, Wei-Hsiang 215
Lin, Chenghua 343, 457
Lin, Tzu-Ling 174
Liu, Ruibo 457
Liu, Si 343
Llorens, Ana 802
Lofi, Christoph 612
Luo, Jing 207
Lustig, Ethan 335

Ma, Yinghao 343, 457
Maezawa, Akira 352
Makarov, Ilya 649
Malandro, Martin E. 327
Malvermi, Raffaele 138
Mancey, Kate 71
Marinelli, Luca 166
Martelloni, Andrea 121
Martinez-Sevilla, Juan C. 319
McFee, Brian 64, 417
McLeod, Andrew 516
McPherson, Andrew P. 121
Meroño-Peñuela, Albert 859
Miller, Jeff 868
Min, Lejun 231
Miron, Marius 553
Mitsufuji, Yuki 215
Morreale, Fabio 37
Morris, Lidia 80
Morsi, Alia 352
Murthy, Hema A. 56
Mühlová, Klára Hedvika 571
Müller, Meinard 223, 289, 433, 832

Nadkarni, Shreyas 21
Nakano, Tomoyasu 561, 840
Nam, Juhan 409, 774
Neuwirth, Markus 383
Newman, Michele 80
Ngo, Quynh Quang 692

Okuma, Lana 197
Oramas, Sergio 375
Özer, Yigitcan 223

Page, Kevin R. 795
Pan, Jiahao 343
Papaioannou, Charilaos 311
Pardo, Bryan 359
Pascual, Santiago 304
Pauwels, Johan 868
Peeters, Geoffroy 535, 749
Pereira, Igor 619
Perez, Miguel 29
Peter, Silvan David 634
Pezzoli, Mirco 138
Peñarrubia, Carlos 182
Plaja-Roglans, Genís 553
Poltronieri, Andrea 859
Pons, Jordi 304
Potamianos, Alexandros 311
Presutti, Valentina 859
Puranik, Ninad 684

R, Gowriprasad 56
Ragni, Anton 457
Rajagopalan, Neha 264
Rammos, Yannis 516
Ramoneda, Pedro 708
Rao, Preeti 21
Regan, Brian 605
Ren, Zeng 627
Repetto, Rafael Caro 440
Richard, Gaël 64, 448
Riley, Xavier 500
Riou, Alain 535
Rizo, David 319
Rohrmeier, Martin 383, 516, 627

Roselló, Adrián 319
Rosendahl, Lisa 795
Roychowdhury, Sujoy 21

Saccomano, Mark 795
Saitis, Charalampos 166
Samiotis, Ioannis Petros 612
Sandler, Mark 868
Sarti, Augusto 138, 257
Scaini, Davide 304
Scavone, Gary 684
Sedlmair, Michael 692
Seetharaman, Prem 359
Serra, Xavier 29, 223, 352, 553, 708, 824
Serrano, Martín 802
Serrà, Joan 304
Shankar, Adithi 553
Shao, Keren 657
Sharma, Megha 37
Shibata, Elisabete 795
Shibata, Kazuhisa 197
Shin, Eunsik 492
Shuai, Hong-Han 174
Shvartzman, Shlomi 642
Siegert, Christine 795
Simonetta, Federico 802
Sioros, George 146
Spijkervet, Janne 817
Sridharan, Srikrishnan 56
Strahl, Sebastian 289
Sturm, Bob L. T. 47
Su, Li 281
Sun, Maosong 157

Takeda, Kazuya 524
Takida, Yuhta 215
Tamer, Nazif Can 223
Tan, Xu 157
Tatsumi, Kana 352
Temperley, David 335
Titov, Maksim 649
Toda, Tomoki 524
Torrente, Álvaro 802
Torres, Bernardo 448
Toyama, Keisuke 215
Tsai, TJ 733
Tseng, Li-Yang 174
Tsukuda, Kosetsu 197, 561, 782

Uzrad, Noy 642

Valero-Mas, Jose J. 182, 708
van Kranenburg, Peter 391
Veldhuis, Tinka 247
Viro, Vladimir 588
Vlhová-Wörner, Hana 571
Vogl, Richard 619
Volk, Anja 247
Vovk, Ivan 649
Vásquez, Marcel A. Vélez 725

Wang, Changhong 64
Watanabe, Kento 398, 774
Wei, I-Chieh 37
Weiß, Christof 832
Weyde, Tillman 757
Widmer, Gerhard 297, 425, 597, 700, 810, 848
Wu, Shangda 157
Wu, Yuxuan 765

Xia, Gus 231, 457
Xue, Wei 343

Yakura, Hiromu 129
Yang, Qiaoyu 676
Yang, Xinyu 207
Yang, Yuting 190
Yin, Hanzhi 457
Yu, Chin-Yun 667
Yu, Dingyao 157

Yuan, Ruibin 343, 457

Zeitler, Johannes 433

Zhang, Ge 343, 457

Zhang, Huan 848

Zhang, Jason 733

Zhang, Liyue 207

Zhang, Yichi 207

Zhao, Jingwei 231

Zhuo, Le 343

Zuidema, Willem 725